INTERNATIONAL TELECOMMUNICATION UNION

ITU-T Q.932

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(03/93)

DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1 NETWORK LAYER

DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1 (DSS 1) – GENERIC PROCEDURES FOR THE CONTROL OF ISDN SUPPLEMENTARY SERVICES

ITU-T Recommendation Q.932

(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.932 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 Similarly, in this reform process, the CCIR and the IFRB have been replaced by the Radiocommunication Sector.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.

© ITU 1994

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	General	1
2	Overview of the generic protocols and of their scope	1
2.1	Three generic protocols	1
3	Co-existence of protocols supported by a network	3
4	Keypad protocol	3
4.1	General	4
4.2	Messages used in the Keypad protocol	4
4.3	Coding of the Keypad facility information element	4
4.4	Elements of procedure	4
4.5	Procedures at the invocation interface	5
4.6	Procedures at the remote interface	7
5	Feature key management protocol	8
5.1	Messages	8
5.2	Procedures	8
6	Functional protocol	11
6.1	General	11
6.2	Separate messages category	12
6.3	Common information element category	16
6.4	Network side channel reservation function	21
7	Message functional definition and content	26
7.1	Messages for supplementary service control	26
8	General message format and information element coding	31
8.1	Message type	31
8.2	Other information elements	31
9	Generic Notification Procedures	46
9.1	General	46
9.2	Call Related Notifications	47
9.3	Call Independent Notifications	47
9.4	Extension of the Notification indicator information element	48
Annex A –	User service profiles and terminal identification	48
A.1	Introduction	48
A.2	User service profiles	50
A.3	Terminal identification	50
A.4	Initialization	50
A.5	Identification procedures	51
Annex B –	SDL diagrams for Hold/Retrieve functions	51
B.1	Introduction	51
Annex C –	Definition of address types	61
Appendix I –	Illustration of the application of the three protocol types	63
I.1	Introduction	63
I.2	Example use of the Keypad protocol	63
I.3	Example of use of the Feature key management protocol	66
I.4	Examples of use of the Functional protocol	70

Page

Appendix II – Functional reference model for the operation of supplementary services 73

Appendix III – General description of component encoding rules 75

III.1 General component structure 75

III.2 Tag 76

III.3 Length of the contents 78

III.4 Contents 80

Appendix IV – Definition of operations, errors and data types 80

IV.2 Components 80

IV.3 Operation value tag 83

IV.4 Error value tag 83

IV.5 Problem tag 84

IV.6 Parameters 86

Appendix V – List of object identifiers defined in Recommendation Q.932 87

Acronyms used in this Recommendation 87

References 88

DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1 (DSS 1) – GENERIC PROCEDURES FOR THE CONTROL OF ISDN SUPPLEMENTARY SERVICES

(Melbourne, 1988; modified at Helsinki, 1993)

1 General

This Recommendation defines the generic procedures applicable for the control of supplementary services at the user-network interface. These procedures may be used for the invocation and operation of supplementary services in association with existing calls or outside any existing calls.

The detailed procedures applicable to individual supplementary services are provided in Q.95x-Series Recommendations. However, typical examples of the application of these generic procedures to some supplementary services are provided in Appendix I for explanatory and illustrative purposes only.

2 Overview of the generic protocols and of their scope

Three generic protocols are defined for the control of supplementary services at ISDN user-network interfaces. These protocols operate at layer 3 of the control plane at the S/T reference points, and assume that the use of layers 1 and 2 conforms to Recommendations I.430 [1], I.431 [2] and Q.921 [3]. In addition, the three generic protocols assume the existence of an established data link and use the acknowledged information transfer service available at the layer 2 to layer 3 interface.

2.1 Three generic protocols

Three generic protocols are defined for the control of supplementary services, two of which are stimulus, the third being functional; these protocols are:

- the Keypad protocol;
- the Feature key management protocol;
- the Functional protocol.

2.1.1 Stimulus protocols

2.1.1.1 Keypad protocol

The Keypad protocol is based on the use of the Keypad facility and Display information elements. The Keypad facility information element may be included in the SETUP and INFORMATION messages. The Display information element may be included in any message sent by the network to the user according to Recommendation Q.931 [4].

This protocol applies to supplementary service invocation in the user-to-network direction, and the keypad facility codes used for the invocation of individual supplementary services are network dependent.

The protocol is stimulus in the sense that it does not require any knowledge about the invoked supplementary service by the user equipment. It may be used in any state of a call and in association with a call for supplementary service invocation and is applicable to both the basic and primary rate access structures. Clause 4 contains a detailed specification of this generic protocol.

2.1.1.2 Feature key management protocol

The Feature key management protocol is based on the use of two information elements that are specified in clause 8: the Feature activation and Feature indication information elements. The Feature activation information element may be included in the SETUP and in the INFORMATION messages in the user-to-network direction. The Feature indication information element may be included in basic call control messages in the network-to-user direction.

This protocol typically applies to supplementary service operation during calls but also allows for non-call related supplementary service control. Non-call related supplementary service control is accomplished by sending an INFORMATION message with the dummy call reference value and which contains a Feature activation information element. The user may send a Feature activation request at any time, and the network may send a Feature indication information element at any time. The supplementary service associated with the Feature identifier is service provider dependent and must be coordinated between the user and the service provider at subscription time. As a service provider option, more than one service profile may be allocated to an interface, but in this case the terminal identification procedures as defined in Annex A must be used in order to relate an appropriate service profile to a particular user.

NOTE – The term "service profile" refers to the information that the network maintains for a given user to characterize the service offered by the network to that user. A portion of this may contain the association of feature identifiers to specific supplementary services. A service profile is normally allocated to an interface but may optionally be allocated to a particular user's terminal equipment or to a group of user's terminal equipment using the procedures as defined in Annex A.

This protocol is stimulus in the sense that it does not require knowledge of the invoked supplementary service by the user's terminal equipment. Knowledge of the service profile contained in the network and of the association of Feature keys to specific supplementary service invocations is required to unambiguously define the requested supplementary service. This protocol is typically applicable to the basic rate access structure. A detailed description of this protocol is contained in clause 5.

2.1.1.3 Information Request procedures

For networks which support access to services using the Keypad protocol and/or the Feature key management protocol, the Information Request (IRQ) procedures may be utilized to prompt for additional information when the network determines that additional information is required.

The support of the procedures in this subclause and the recognition of the Information Request information element is a network and a user option, and is supported on the basis of a bilateral agreement between the user and the network.

The information request message sequence is initiated when the network sends the user and INFORMATION message (in any call state using an active Call Reference Value or the dummy call reference value) or a SETUP ACKNOWLEDGE message (as first response to a SETUP message in case of overlap sending) that contains the Information Request information element. The information request may be included in the SETUP ACKNOWLEDGE message when the network is responding to a feature request contained in a SETUP message that contains no called party address information. The Information Request information element shall be coded with the information request indicator set to "prompt for additional information" and type of information set to the appropriate value. After sending the information request prompt, the network will start timer T302 on receipt of every INFORMATION message if the request information is not complete.

No Recommendation Q.931 call state changes should occur when the INFORMATION message is sent or received.

The user may always send the requested information in Keypad facility information elements contained in one or more INFORMATION messages. In addition, if the information requested was a called party number, then the user may also send the requested information in the Called party number information element in the INFORMATION messages.

In both the call associated and non-call associated cases, when the network has determined that sufficient information has been received to proceed, it may send an INFORMATION message to the user, containing the "information request completed" to signal the end of information sending.

If the additional information was requested during overlap sending, and if the network has determined that sufficient information has been received for the call to proceed, then the network shall send a CALL PROCEEDING message to the user with the Information Request information element coded to indicate that the request for information has been completed unless this complete indication has been returned in an INFORMATION message earlier. If no call is to be established based on the information received by the network in the overlap sending state but a non-dummy call reference has been used for the information exchange, the network should initiate clearing of the call reference by sending a DISCONNECT message. The DISCONNECT message in this case may contain the Information Request information element, coded to indicate that the request for information has been completed and the Cause information element coded to cause value #16 "normal clearing".

If the user initiates call clearing with a clearing message that allows a response from the network (DISCONNECT or RELEASE), the network should follow normal call clearing procedures and may include the Information Request information element in the appropriate call clearing message (RELEASE or RELEASE COMPLETE), coded to indicate that the request for information is complete.

2.1.2 Functional protocol

The Functional protocol is based on the use of the Facility information element and the FACILITY message, as well as of other specific functional messages specified in clause 7. This protocol is symmetrical and is applicable to both the basic and primary rate access structures.

This protocol is functional in the sense that it requires the knowledge of the related supplementary service by the user equipment supporting it. This facilitates user equipment operation without human intervention by defining semantics for the protocol elements which user equipment can process on its own.

Functional procedures may follow a Keypad or a Feature key management supplementary service invocation.

3 Co-existence of protocols supported by a network

Networks may support more than one of these generic protocols for the control of supplementary services. The support of multiple generic protocols is a network option. Users shall be informed by the service provider at subscription time of the supplementary services available, and of the generic protocols supported on their access.

As a general rule, the Functional protocol shall be used unless the network specifies the use of a stimulus protocol for the invocation of certain supplementary services, or the users have subscribed to a feature key management facility and service profile.

In general, the Keypad protocol and Feature key management protocol have only local significance while the Functional protocol may have other than local significance.

For a given call instance, the protocol applied at a local interface may be different from the one applied at a remote user's interface.

Some networks may support only one of the generic protocols per user access for the invocation of supplementary services. Other networks may choose to support a single generic protocol for the control of supplementary services, depending on the user access interface type (e.g. Feature key or Keypad on the basic access, functional on the primary access). This has to be arranged at subscription time.

Network supporting multiple generic protocols per access in the user to network direction (i.e. for the supplementary service invocation) will implicitly recognize the protocol option chosen by the user on the basis of the received message type or information element type.

Networks supporting more than one generic protocol per access in the network to user direction (i.e. at the remote user interface) may choose to apply a particular protocol depending on the supplementary service characteristics involved. In a case where, for a given supplementary service, more than one protocol can be supported, then the use of the terminal identification procedure as described in Annex A may have to be used in order to determine the protocol supported by that user's terminal equipment, as registered at subscription time.

User service profile procedures as described in Annex A provide a means of characterising the services offered to different groups of one or more terminals on the same user access interface. A network may, therefore, use a parameter within a user service profile to determine the appropriate procedures for network initiated supplementary services towards the associated group of one or more terminals.

4 Keypad protocol

The Keypad protocol is based on the use of the Keypad facility and Display information elements. While the generic procedures associated with Keypad invocation are specified in this clause, the allocation of the access codes used to request/indicate a supplementary service are not to be standardized within the CCITT.

An example of the use of the Keypad protocol is given in Appendix I.

4.1 General

This generic procedure is based on the use of:

– the Keypad facility information element by the user to invoke a supplementary service from the network by providing access codes using either en-bloc or overlap sending; and

– the Display information element by the local network to give an indication to the local user (or by the remote network to the remote user) regarding a supplementary service being invoked. This procedure may be complemented in the case of calls where the Bearer capability information element in the SETUP message is coded indicating "speech" or "3.1 kHz audio", or "UDI with tones/announcements", by the provision of in-band tones/announcements to the user.

NOTE– As a network option, the Keypad facility information element may be used by the network to give an indication to the user when the network expects an automatic reaction to the received information to acknowledge an invoked supplementary service. As the semantics of the Keypad facility information element are not standardized, the use of the Keypad facility information element in the network-to-user direction may inhibit terminal portability since for a terminal to operate successfully on more than one network it must be capable of interpreting various different semantics as assigned by the network to the Keypad facility information. In any case, user equipment not supporting this option shall follow the error recovery procedures defined in 5.8/Q.931 of receipt of the Keypad facility information element.

The Keypad protocol may be used in conjunction with the Feature key management (see clause 5) or Functional protocol (see clause 6) during the invocation of a supplementary service.

The Keypad protocol is based on the use of the Keypad facility information element within the INFORMATION or SETUP messages during the establishment, active and clearing phases of a call.

4.2 Messages used in the Keypad protocol

As specified in Recommendation Q.931, the Keypad facility information element may be included in both the SETUP and INFORMATION messages and may be sent in the user-to-network direction.

4.3 Coding of the Keypad facility information element

The contents of the Keypad facility information element are a string of IA5 characters. The syntax of the IA5 character string and the allocation of values for given supplementary services are not subject to CCITT standardization.

4.4 Elements of procedure

4.4.1 General

The Keypad protocol includes the following aspects:

1) the Keypad protocol may be used during the call establishment, active, and clearing phases of a call to invoke supplementary services. Supplementary service information is conveyed in Keypad facility information elements sent in either SETUP or INFORMATION messages;

2) supplementary service information can be sent from the user to the network either en-bloc or using overlap sending;

3) the network may prompt the user to send the required information using the Display information element and/or in-band tones or announcements. Whether this action shall occur or not is supplementary service and network specific. In any case, in-band tones or announcements shall only be used when the Bearer capability information element indicates "speech" or "3.1 kHz audio"or "UDI with tones/announcements";

4) there may be different combinations of user provided information followed by network prompts. Examples of such possible combinations are shown in Table 4-1, where the term "stage" is used to refer to information sent by the user between network prompts (if any).

TABLE 4-1/Q.932

Example of stages for sending of information

Number of stages	Sending information		
1	All information sent en-bloc		
1	All information sent overlap		
2	Overlap	Prompt	Overlap
2	En bloc	Prompt	En-bloc
2	Overlap	Prompt	En-bloc
2	En bloc	Prompt	Overlap
3	Overlap	Prompt	Overlap
	... Prompt	Overlap, etc.	

NOTE – The number of possible stages is network dependent and may also be dependent on the specific supplementary service being invoked.

4.5 Procedures at the invocation interface

4.5.1 User procedures

The procedures below define how information (using either en-bloc or overlap sending) may be sent in a single stage from the user to the network. The procedures are applicable for each stage of user-to-network information sending.

4.5.1.1 En-bloc sending of access codes

En-bloc sending of supplementary service information is accomplished by sending the "complete" supplementary service information in:

– the SETUP message, if the supplementary service is being invoked during the call establishment; or

– the INFORMATION message, if the supplementary service is being invoked from the active phase of the call or during the clearing phase of a call.

The term "complete" supplementary service information means that sufficient supplementary service information is sent to the network to specify a service without any additional network prompting being required. The network determines that the supplementary service information is "complete" by either:

– analysis of the information contents of the Keypad facility information element; or

– the presence of a "sending complete" indication (see 5.1.3/Q.931).

If the network determines that the information contents of the Keypad facility information element are invalid, the network shall use the error procedures specified in 4.5.2.3.

If the network determines that the information contents are valid and that the user is allowed to invoke the requested service, the network shall respond using the procedures as specified in 4.5.2.1.

4.5.1.2 Overlap sending of access codes

Overlap sending of supplementary service information is the sending of the "complete" supplementary service information (see 4.5.1.1 for the definition of complete) segmented such that a number of Recommendation Q.931 messages are used to convey the "complete" supplementary service information. The possible combination of messages:

a) for supplementary services invoked during call establishment, consists of using the SETUP message plus one or more INFORMATION messages which will be sent in the overlap sending state; or

b) for supplementary services invoked in the active or clearing phases of the call, consists of using two or more INFORMATION messages.

For case a), normal overlap sending procedures, as specified in 5.1.3/Q.931, shall be used.

For case b), the transmission or receipt of INFORMATION messages shall not cause any change to the Recommendation Q.931 call state.

The network shall respond to valid supplementary service information with one of the network responses as described in 4.5.2.1. If the supplementary service information is invalid, then the error procedures as described in 4.5.2.3 shall apply.

4.5.2 Network procedures

4.5.2.1 Network responses to user requests

After receiving information from the user, the network may take one of the following actions. Items 1) to 4) are applicable in the cases of both en-bloc and overlap sending; item 5) is applicable only in the case of information sent using overlap sending.

1) Clear the call reference via the normal call clearing procedures (see 5.3/Q.931) including the appropriate Cause and optional Display information element(s).

2) Send a CALL PROCEEDING message to the user.

   NOTE – This network response is only applicable in a case where the supplementary service is being invoked during call establishment and not in the cases of the supplementary service being invoked from the active or clearing phases of the call.

3) Send an INFORMATION or clearing message to the user that includes a Display information element containing an appropriate response to the request for a supplementary service. The receipt of an INFORMATION message by the user shall not cause any change to the Recommendation Q.931 call state.

4) Prompt the user for more information using the procedures as specified in 4.5.2.2. This further information could be additional, or new information input by the user or another attempt by the user to re-input the original information correctly. Such procedures are network dependent and may be supplementary service specific.

5) Wait for more overlap information. The allowed waiting period is governed by timer T302 in the case of information sent in the overlap sending state and call control timers for overlap information sent during other phases of the call.

The precise action to be taken is dependent on the specific supplementary service being invoked.

4.5.2.2 Network prompting and in-band tone/announcement control

The network may prompt the user for more information or may provide in-band tones or announcements regardless of whether or not the Keypad facility information element was included in the initial SETUP message. The network shall determine whether prompting and/or in-band tone or announcement control should occur. Possible factors governing the provision of prompting and in-band information are:

- the nature of the supplementary service;
- the value of the inter-digit timer;
- the type of interface; and
- the current status or progress of the supplementary service request.

Simultaneously with the application of in-band tones or announcements, the network may send a PROGRESS message containing a Progress indicator information element with the progress descriptor No. 8, *In-band information or appropriate pattern now available*.

The network may, in addition to an audible prompt (i.e. tone or announcement), request information from the user by sending an INFORMATION message which contains the Display and/or Signal information elements (but shall not contain the Called party number information element).

The sending of the INFORMATION message by the network does not result in a change to the Recommendation Q.931 call state. However, when this message is sent in the network overlap sending state, timer T302 shall be re-initialized.

The network may prompt the user more than once (i.e. multiple stages may occur), but the network should not prompt the user again prior to the user's response or, when in the overlap sending state, prior to the expiry of timer T302. This is to avoid situations where a user's response could be related to two unacknowledged network prompts.

NOTE – As a network option, the Information Request procedures described in 2.1.1.3 may be used to prompt the user for additional information related to a given service request.

4.5.2.3 Error conditions and treatment

An error condition exists in the following circumstances:

a) timer T302 expires and complete information has not been received;

b) information containing a "sending complete" indication indicating en-bloc sending, but the user information sent is not complete;

c) information received by the network (complete or incomplete) is invalid. Invalid information is information sent with incorrect format or containing invalid facility identifier or parameter codes;

d) the user attempts to invoke a supplementary service to which the user has not subscribed or to which the user is not allowed access.

The action to be taken by the network in these situations is as follows:

NOTE – The text below identifies possible actions that may be taken in an error situation. The specific action to be taken is network and supplementary service dependent.

4.5.2.3.1 Supplementary service being invoked during call establishment

The network shall take one of the following actions:

i) In-band tones or announcements are applied. If a SETUP ACKNOWLEDGE message has not already been sent, the network shall send a CALL PROCEEDING message to the user, indicating the B-channel to be used and including the Progress indicator information element with progress descriptor No. 8, *In-band information or appropriate pattern is now available*.

If a SETUP ACKNOWLEDGE message has already been sent, the network shall send a PROGRESS message to the user, including the Progress indicator information element with the progress descriptor No. 8, *In-band information or appropriate pattern is now available*.

The network may prompt the user using the procedures as specified in 4.5.2.2 to re-input the required information. Otherwise, after the in-band tone or announcement has been applied, the call reference shall be cleared by either the user initiating call clearing or the network initiating call clearing at the expiry of a tone or announcement timer. Both the network and the user shall use the clearing procedures as specified in 5.3/Q.931.

ii) No in-band tones or announcements are to be applied. The call reference shall be cleared by the network initiating call clearing procedures as specified in 5.3/Q.931.

4.5.2.3.2 Supplementary service being invoked from the active state or during the call clearing phase

The network shall take one of the following actions:

i) In-band tones or announcements are applied. The network may prompt the user using the procedures as specified in 4.5.2.2 to re-input the request. Otherwise, depending on the specific supplementary service being invoked, the call shall either be cleared or remain in the same call state. In the case where the call is cleared, clearing shall occur after the in-band tone or announcement has been applied. Clearing shall occur either by the user initiating call clearing or by the network initiating call clearing at the expiry of a tone or announcement timer. Both the network and the user shall use the clearing procedures as specified in 5.3/Q.931.

ii) No in-band tones or announcements are to be applied. Depending on the specific supplementary service being invoked, the call shall either be cleared or remain in the same call state. In the case where the call is to be cleared, the call reference shall be cleared by the network initiating call clearing using the procedures as specified in 5.3/Q.931. If the call remains in the same call state, the user may be informed that the supplementary service request was unsuccessful by the network sending an INFORMATION message in accordance with 4.5.2.1, item 3).

4.6 Procedures at the remote interface

The Display and/or Signal information elements can be used for the purpose of providing notification to the remote user from the network. In this case, however, this information is used simply for the purpose of informing the human user, and no automatic reaction to the received information is to be performed by the user's equipment itself.

5 Feature key management protocol

The Feature key management protocol is a mechanism allowing users to invoke network supplementary services. As these are stimulus procedures, the protocol elements do not, by themselves, identify the service invoked. To determine the service invoked requires knowledge of the user's service profile maintained in the network. No call state changes directly occur by these procedures.

The Feature key management protocol is based on two information elements: Feature activation and Feature indication. The Feature activation information element is the means by which a user requests a supplementary service. The Feature activation information element contains a feature identifier number which the network then maps to the corresponding service as indicated by that user's service profile. The user's equipment need not have any knowledge of what service is being indicated by the feature identifier number and the user may send a feature request at any time.

Feature indication is the means by which a response to a Feature activation is indicated by the network. The feature identifier number correlates the network's response with a user's request and/or an indicator associated with a user's equipment. The Feature indication information element also contains a status indicator. The status indicator indicates the status of the requested service and may be used by the user's equipment as appropriate with its man-machine interface.

5.1 Messages

The Feature activation and Feature indication information elements may be present in several of the messages defined in Recommendation Q.931. The Feature activation information element may appear in the following messages in the user-to-network direction:

1) SETUP

2) INFORMATION

The Feature indication information element may be sent in the network-to-user direction in the following messages:

a) SETUP

b) SETUP ACKNOWLEDGE

c) CONNECT

d) CALL PROCEEDING

e) ALERTING

f) INFORMATION

g) DISCONNECT

h) RELEASE

i) RELEASE COMPLETE

5.2 Procedures

5.2.1 Assumptions and restrictions

a) These procedures assume that only one Feature activation request will appear in a message.

b) The phrase "call associated services" used herein is defined as services which act upon or relate to an existing call (as defined by the existence of a call reference).

c) These procedures are used for the invocation of supplementary services which relate to predefined specific bearer capabilities and/or are context dependent. Hence the capability to include protocol elements to indicate the bearer capability that the supplementary service is to act upon is not provided.

5.2.2 Invocation of supplementary services

The user may request a feature by including a Feature activation information element in the messages defined in 5.1. If the INFORMATION message is used, it may be sent at any time. The user will indicate the desired feature by specifying the appropriate value in a feature identifier number.

5.2.2.1 Determination of call reference in the INFORMATION message

When the Feature activation information element is sent in the INFORMATION message, then the following rules apply:

a) if no call references exist, then the dummy call reference must be used (for this non-call associated service type);

b) if a call reference(s) has been established, then that value may be used regardless of whether the service type is call associated or non-call associated;

c) if a call reference(s) has been established, the dummy call reference may be used only if the service type is non-call associated. If the service type is call associated, then the appropriate call reference must be used. An exception to this rule is when only one call is established. In this instance it is permissible for the user to use the dummy call reference for either service type.

This is summarized in Figure 5-1.

It is always correct for the user's equipment to use the dummy call reference when no calls exist, or to use an established call reference if one exists, independent of the service type.

Service type	No calls exist	Call(s) exist
Non-call associated	Use dummy call reference	Use dummy or active call reference
Call associated	Error; not allowed	Use an active call reference (Note)

NOTE – The dummy call reference value may be used if only one call is established.

FIGURE 5-1/Q.932

Use of the call reference in an INFORMATION message

5.2.3 Network responses

The network may respond to a Feature activation request in several ways. This action will be supplementary service and network specific.

5.2.3.1 Normal responses

5.2.3.1.1 Return of a Feature indication

The network may return a Feature indication information element in an INFORMATION message or any other appropriate call control message as defined in 5.1. The feature indication may or may not have the same feature identifier number as was present in the original feature activation request. The status indicator will be provided as appropriate to the specific supplementary service requested.

5.2.3.1.2 Prompting for further information

The network may prompt the user for more information. When in the overlap sending state, it may do so using the Information Request procedures (described in 2.1.1.3).

The user's response shall follow normal overlap sending procedures as defined in Recommendation Q.931. As a network option, the Information Request procedures described in 2.1.1.3 may be used to prompt the user for additional information related to a given service request.

5.2.3.1.3 Implicit response

The network, under certain situations, may not return any explicit indication to the user after a feature activation request. In this case the response is implicit, such as the acknowledgement inherent in providing the service.

5.2.3.1.4 Return of Signal, Cause, or Display information elements

The network may return any combination of Signal, Cause, or Display information elements in conjunction with the responses as described in 5.2.3.1. The use of these information elements is supplementary service and network specific. Coding and the appropriate messages that may contain these information elements are as defined in Recommendation Q.931.

5.2.3.2 Responses during error conditions

When an error condition exists (as defined in 5.2.5), the network may:

a) Respond with one or more of the following options:

1) return a Feature indication information element;

2) prompt for further information (see 2.1.1.3);

3) provide an implicit response; or

4) return Signal, Cause, or Display information elements.

b) Ignore the Feature activation request and not respond at all.

c) Clear appropriate existing calls in conjunction with the above actions.

5.2.4 General aspects

5.2.4.1 Use of Feature indication information elements independent of a feature request

The network may choose to send Feature indication information at any time independent of the status of any call(s). Multiple Feature indication information elements may be returned in an INFORMATION message or in an appropriate call control message if more than one indicator is to be updated.

5.2.4.2 Deactivation procedures

When explicitly deactivating a supplementary service, two methods may be used:

a) sending of a feature activation request with the same feature identifier may deactivate the supplementary service. Some supplementary services may be "toggled" on and off;

b) sending of a feature activation request with a different feature identifier which is explicitly defined (between the user and network) as the deactivator for that particular supplementary service.

5.2.4.3 Clearing of a call

If a Feature activation information element is sent using the call reference of an active call, and that call is cleared for some reason, then there does not exist a call reference with which to correlate the feature indication. If a Feature indication information element is to be returned, then one of the following options may be used:

a) the network may send a Feature indication information element in one of the call clearing messages (i.e. DISCONNECT, RELEASE, or RELEASE COMPLETE);

b) the network may send a Feature indication information element in an INFORMATION message after clearing has occurred using the dummy call reference.

5.2.5 Error conditions

5.2.5.1 Invalid feature activation request

If a user requests a feature using an invalid feature identifier number, the network may take actions specified in 5.2.3.2 as appropriate. An invalid feature identifier number is one in which the user has not subscribed to a corresponding service, or the value is not understood by the service provider (e.g. out of range).

5.2.5.2 Invalid call reference

If a user violates the use of the call reference as stated in 5.2.2.1, the network should not provide the service and should respond as indicated in 5.2.3.2.

5.2.5.3 Sending of multiple feature activation requests

If a sequence of feature activation requests is received in separate messages so rapidly that the network cannot respond to the first feature activation request prior to receiving a subsequent feature activation request, the network may take one of the following actions:

a) act upon all feature activation requests by returning multiple Feature indication information elements (or other responses as detailed in 5.2.3.1). These may be sent in a single message or in multiple messages;

b) act upon the first feature activation request by returning a single response. This response should correspond to the first feature activation request. Feature activation requests after the first request are discarded and ignored by the network.

The determination of which action to take is network and supplementary service specific.

6 Functional protocol

6.1 General

6.1.1 Introduction

This subclause specifies the functional signalling procedures for the control of supplementary services at the user-network interface. This generic protocol utilizes functions and services provided by Recommendations Q.930 [5] and Q.931 [4] basic call control procedures and the functions of the data link layer as defined in Recommendations Q.920 [6] and Q.921 [3].

Support of some supplementary services in the Q.95x-Series requires the support of procedures provided in this section and the associated protocol. The support of these procedures otherwise is a network and user option based on a bilateral agreement.

6.1.2 Scope of the procedures

The procedures defined in clause 6 specify the basic methodology for the control (e.g. invocation, notification, cancellation, etc.) of supplementary services. The procedures are independent of whether the user-network interface is a basic or primary rate interface.

6.1.3 Categories of procedures

Two categories of procedures are defined for the functional signalling for supplementary services. The first category, called the separate message approach, utilizes separate message types to indicate a desired function. The HOLD and RETRIEVE set of messages are identified for this category.

The second category, called the common information element procedure, utilizes the Facility information element.

Both categories are specified in a symmetrical manner and can be signalled both in the network-to-user and the user-to-network directions.

6.1.4 Supplementary service functions

The control of supplementary services by either the network or the user includes the following cases:

a) the invocation of supplementary services during the establishment of a call;

b) the invocation of supplementary services during the clearing of a call;

c) the invocation of call related supplementary services during the active state of a call;

d) the activation, deactivation, interrogation or registration of supplementary services independent from an active call;

e) the invocation of multiple, different supplementary services within a single message;

f) the invocation of supplementary services related to different calls;

g) cancellation of invoked supplementary services and notification to the initiator of the supplementary service.

The correlation of a call related supplementary service and the call which it modifies is provided by use of the call reference [cases a), b), c), e), f) and g) listed above].

The correlation of call independent supplementary service invocations and their responses is provided by the combination of the call reference of the message containing the Facility information element and the invoke identifier present within the Facility information element itself [refer to cases d), e) and g)].

The identification of different supplementary service invocations within one single message is provided by the invoke identifier of the corresponding Facility information element [refer to cases e) and g)]. The identification of supplementary service invocations related to different calls (e.g. HOLD) is provided by different messages with the corresponding call reference of the appropriate call [refer to case f)], i.e. different call reference values are used to identify each call individually.

6.2 Separate messages category

The messages defined in this section are specified as separate functional messages for invoking specific functions which require changes of the resources and the auxiliary state and also require synchronization of the peer-to-peer state machines. Therefore, these functions cannot be performed in conjunction with the call establishment and clearing procedures but may be used in conjunction with various supplementary services. The functions of these messages are not to be duplicated or overlapped by those of the Facility information element.

The following individual messages are defined:

HOLD

HOLD ACKNOWLEDGE

HOLD REJECT

RETRIEVE

RETRIEVE ACKNOWLEDGE

RETRIEVE REJECT

6.2.1 Hold and Retrieve functions

The Hold function is used to put an existing call which is in the establishment or in the active phase in the Call Held auxiliary state. If optionally subscribed by the user, it reserves the B-channel in use (if any) or any other B-channel (if none was already reserved) for that user which is identified by a Connection Endpoint Suffix (CES), as defined in Recommendation Q.921. In addition, the call reference of the held call shall be retained for possible subsequent call retrieval and channel reconnection.

As an option, based on a subscription arrangement between the user and the service provider, the B-channel may be released for subsequent re-use by the network for another call.

On receipt of a HOLD message the user or the network shall return a HOLD ACKNOWLEDGE message, provided that the requested function can be performed. The network disconnects any B-channel allocated to the call in progress or active when putting that call in the Call Held auxiliary state.

Procedures for reserving B-channels in conjunction with the provision of supplementary services are provided in 6.4.

The HOLD ACKNOWLEDGE message puts the call in the Call Held auxiliary state and indicates that the Hold function has been performed. The HOLD REJECT message indicates that the hold request was denied and returns the call to the condition it was in prior to the hold request. The HOLD REJECT message contains the Cause information element with an appropriate cause value.

The Retrieve function reconnects the user to the requested B-channel. The RETRIEVE message requests that a call be retrieved. The RETRIEVE ACKNOWLEDGE message indicates that the Retrieve function has been performed. The RETRIEVE REJECT message indicates that the retrieve request was denied. The RETRIEVE REJECT message contains the Cause information element with e.g. cause No. 44, *requested channel not available*, or No. 34, *no channel available*.

The HOLD and RETRIEVE family of messages may be used in a symmetrical manner.

6.2.1.1 Auxiliary states for Hold and Retrieve

The Hold function may be performed in the Q.931 call states specified in 6.2.2.1. The concept of two dimensional state space is being introduced in this subclause to ensure state synchronisation between the user and the network. In this way, there will be two states associated with each call. The first will be a Recommendation Q.931 call state and the second will be an auxiliary state associated with the Hold function. This state space can be represented by two coordinates: one is a Recommendation Q.931 call state coordinate and the other is a Hold function related auxiliary state coordinate. If a Recommendation Q.931 call state transition occurs, the first coordinate is updated. If a call is put on hold, the second coordinate is updated. When the held call is reconnected, the second coordinate is again updated.

There are six auxiliary states associated with the Hold and Retrieve functions:

i) Idle.

ii) Hold Request – A request has been made for the Hold function.

iii) Call Held – The call is held.

iv) Retrieve Request – A request has been made for the Retrieve function.

v) Hold Indication – A request has been received for the Hold function.

vi) Retrieve Indication – A request has been received for the Retrieve function.

6.2.1.2 An example of dimensioned state space

When a call is in the Outgoing Call Proceeding state, the dimensioned state space would be:

(Outgoing Call Proceeding, Idle)

Then the user requests the Hold function. The dimensioned state would become:

(Outgoing Call Proceeding, Hold Request)

The call is then put on Hold. The user becomes aware of this upon receiving the HOLD ACKNOWLEDGE message from the network. The dimensioned state would now be:

(Outgoing Call Proceeding, Call Held)

The user may receive subsequent call progress messages changing the dimensioned state to:

(Active, Call Held)

Then the user requests the Retrieve function. The dimensioned state would become:

(Active, Retrieve Request)

When the call is reconnected, the dimensioned state space would be:

(Active, Idle)

6.2.2 Hold procedures

The Hold function should be invoked in association with an existing call (i.e. during the establishment or active phase of a call).

The invocation of the Hold function does not affect the existing Recommendation Q.931 call states but does affect the auxiliary state. The request for placing a call on hold places the initiating entity in the Hold Request state. The responding entity will acknowledge this request with a HOLD ACKNOWLEDGE message if this operation was successful. This will result in the auxiliary state being put in the Call Held state. If the requested Hold function cannot be performed, then a HOLD REJECT message will be returned with the appropriate cause value. This will result in both the initiating and responding entities returning to their respective auxiliary states prior to the sending of the hold request.

6.2.2.1 Normal operation, initiating side

The Hold function is initiated by sending a HOLD message containing an established call reference across the user-network interface. Following the transmission of the HOLD message, the initiating entity shall start timer T-hld (the value of timer T-hld is specified in 6.2.5); enter the Hold Request auxiliary state; and wait for a HOLD ACKNOWLEDGE message. (The auxiliary states are defined in 6.2.l.l.) The call to be put on hold must be allocated to the Connection Endpoint Identifier (CEI) sending the request, and a B-channel must be selected for the call and/or for the CEI.

The Hold Function may be initiated in the following Q.931 call states for call originations:

- Outgoing call proceeding (U3), (N3);
- Call delivered (U4), (N4);
- Active (U10), (N1O).

The Hold function may be initiated in the following Q.931 call states for call terminations only if a point to point configuration exists:

- Call Received (U7), (N7);
- Connect Request (U8), (N8);
- Incoming Call Proceeding (U9), (N9);
- Active (U10), (N10).

For call terminations in a multipoint configuration, the Hold function may only be initiated in the Active (U10), (N10) state.

Upon receipt of a HOLD ACKNOWLEDGE message, the initiator of the hold request shall stop timer T-hld; release the B-channel, if connected; and enter the Call Held auxiliary state.

Upon receipt of a HOLD REJECT message, the initiator of the hold request shall stop timer T-hld and return to the auxiliary state it was in prior to the sending the HOLD message .

If timer T-hld expires before receiving a HOLD ACKNOWLEDGE, the initiating entity shall enter the Idle auxiliary state.

6.2.2.2 Normal operation, receiving side

Upon receipt of a HOLD message, the receiving entity shall enter the Hold Indication auxiliary state.

If the hold request is allowed within the current Q.931 call state (see 6.2.2.1 for allowed call states), the receiving entity shall release the B-channel, if connected; return a HOLD ACKNOWLEDGE message to the initiating entity; and enter the Call Held auxiliary state.

If the hold request is not allowed within the current Q.931 call state, the receiving entity shall follow the procedures specified in 6.2.2.4.

6.2.2.3 Call Held auxiliary state

Upon successful completion of the Hold function (i.e. the initiating entity has received a HOLD ACKNOWLEDGE message), the call shall be in the Call Held auxiliary state at both sides of the interface. While in the Call Held auxiliary state, a B-channel shall not be connected for the held call, even if an event occurs that would otherwise cause such a connection according to the procedures of Recommendation Q.931 (e.g. receipt of a CONNECT message for the call reference of the held call). A call may only be in the Call Held auxiliary state while the Q.931 call state is one of the states allowed by 6.2.2.1. or in U12/Nl2 Disconnect indication state.

On transition of the call state into another state that is not defined in 6.2.2.1 and is not U12/N12 Disconnect indication state, the entity shall transition into the Idle auxiliary state.

In general, the Hold function does not prohibit the retrieval of a held call by either side of the interface (i.e. the initiating entity of the Hold function may be the responding entity of the Retrieve function and the responding entity of the Hold function may be the initiating entity of the Retrieve function). However, the use of such symmetric procedures will be specified within the procedures of individual supplementary services when they apply.

6.2.2.4 Exceptional procedures

If the HOLD message is not recognized by the receiving entity, the error procedures of 5.8/Q.931 shall apply.

If a HOLD message is received in states (U/N12) or (U/Nl9), the receiving entity shall ignore the hold request and continue with normal call clearing procedures.

If a HOLD message is received in any other state that is not an allowed state for initiating the Hold function (see 6.2.2.1), the receiving entity shall return a HOLD REJECT message with cause value #101 "message not compatible with call state", and remain in the auxiliary state it was in prior to the reception of the HOLD message.

Upon receipt of a HOLD REJECT message, the initiator of the hold request shall stop timer T-hld and return to the auxiliary state it was in prior to the sending of the HOLD message.

6.2.3 Retrieve procedures

The Retrieve function is requested by sending a RETRIEVE message. This message may be sent while the auxiliary state is in the Call Held state.

The RETRIEVE message may indicate a preferred, any, or exclusive channel. Procedures for the use of the Channel identification information element are as defined for basic call control. Upon the sending of the RETRIEVE message, the auxiliary state of the initiator would be the Retrieve Request state.

If the Retrieve request is successful, the RETRIEVE ACKNOWLEDGE message will be returned with the selected B-channel indicated. The initiator should not assume that call retrieval has occurred until it receives this message. The initiating and responding entities would then return to the Idle state.

If the Retrieve request is not successful, the RETRIEVE REJECT message will be returned with an appropriate cause. The initiating and responding entities would then remain in the same auxiliary state as they were in prior to the sending and receipt of the RETREIVE message respectively.

6.2.3.1 Normal operation, initiating side

The Retrieve function is initiated by sending a RETRIEVE message containing the call reference of a held call across the user-network interface. The RETRIEVE message may only be sent in the Call Held auxiliary state and the Q.931 call states U/N 3, 4, 7, 8, 9, 10 and 12. Following the transmission of the RETRIEVE message, the initiating entity shall start timer T-ret (the value of timer T-ret is specified in 6.2.5); enter the Retrieve Request auxiliary state; and wait for a RETRIEVE ACKNOWLEDGE message.

Upon receipt of a RETRIEVE ACKNOWLEDGE message, the initiator of the retrieve request shall stop timer T-ret; connect to the B-channel; and enter the Idle auxiliary state. Upon receipt of a RETRIEVE REJECT message, the initiator of the retrieve request shall stop timer T-ret and enter the Call Held auxiliary state. If timer T-ret expires before receiving a RETRIEVE ACKNOWLEDGE message, the initiating entity shall enter the Call Held auxiliary state.

6.2.3.2 Normal operation, receiving side

Upon receipt of a RETRIEVE message, if the Retrieve request is allowed within the current auxiliary state and Q.931 call state (see 6.2.3.1 for allowed states), and an appropriate B-channel may be allocated for the call, the receiving entity shall return a RETRIEVE ACKNOWLEDGE message towards the initiating entity, enter the Idle auxiliary state, and connect to the appropriate B-channel.

If the RETRIEVE message contained a Channel identification information element indicating an exclusive B-channel and it is acceptable, the receiving entity shall not include the Channel identification information element in the RETRIEVE ACKNOWLEDGE message.

The channel negotiation procedures within the RETRIEVE/RETRIEVE ACKNOWLEDGE messages follow those specified in 5.1.2/Q.931.

6.2.3.3 Exceptional procedures

If a RETRIEVE message is received in a state other than the Call Held or Retrieve Request auxiliary state and the Q.931 call states allowed by 6.2.3.1, the receiving entity shall return a RETRIEVE REJECT message with cause value #101 "message not compatible with call state", and remain in the same auxiliary state it was in prior to the reception of the RETRIEVE message.

If a RETRIEVE message indicates an "exclusive" B-channel, and that channel is not available to retrieve the held call, then the receiving entity shall return a RETRIEVE REJECT message with cause value #44 "requested circuit/channel not available".

If a RETRIEVE message indicates a “preferred” or “any” B-channel, or if a Channel Identification is not included, and no channel is available to retrieve the held call, then the receiving entity shall return a RETRIEVE REJECT message with cause value #34 “no B-channel available”.

Upon receipt of a RETRIEVE REJECT message, the initiator of the retrieve request shall stop timer T-ret and enter the Call Held auxiliary state.

6.2.4 Collision of messages

These procedures assume both sides of an interface have implemented the Hold and Retrieve functions symmetrically. If a HOLD message is received immediately after sending a HOLD message for the same call reference (i.e. HOLD message is received in the Hold Request auxiliary state), the receiving entity shall continue processing the Hold request according to the procedures specified in 6.2.2.2.

If the user side receives a RETRIEVE message immediately after sending a RETRIEVE message for the same call reference (i.e. a RETRIEVE message is received in the Retrieve Request auxiliary state), the user side shall stop timer T-ret, enter the Retrieve Indication auxiliary state, and follow the procedures in 6.2.3.2.

If the network side receives a RETRIEVE message immediately after sending a RETRIEVE message for the same call reference (i.e. a RETRIEVE message is received in the Retrieve Request auxiliary state), the network side shall ignore the received RETRIEVE message, remain in the Retrieve Request auxiliary state, and continue according to the procedures specified in 6.2.3.1.

6.2.5 Parameter values (Timers)

The following timers shall be used:

Timer	Timeout value	Cause for start	Normal stop
T-hld	4s	HOLD sent	HOLD ACKNOWLEDGE HOLD REJECT received
T-ret	4s	RETRIEVE sent	RETRIEVE ACKNOWLEDGE RERIEVE REJECT received

6.2.6 Clearing of a held call

A call in the Call Held auxiliary state may be cleared in either direction by sending a DISCONNECT message across the user-network interface. Normal call clearing according to the procedures of 5.3/Q.931 shall apply, except that disconnecting the B-channel does not apply. Upon receiving or sending a RELEASE COMPLETE message, the held call shall be considered in the Idle auxiliary state and null (U0, N0) Q.931 call state.

6.3 Common information element category

A REGISTER, a FACILITY or an existing Recommendation Q.931 call control message is used to carry the Facility information element which requests the desired supplementary service.

This functional procedure provides a flexible and open ended approach to the provision of supplementary service protocols and:

- allows new services to be easily introduced;
- allows multiple supplementary service invocations within one message;
- supports supplementary services with a large number of variants without a proliferation of new messages;
- supports non-call associated supplementary services.

In addition, the use of the FACILITY message allows the actions and events related to supplementary services to be clearly separated from those associated with basic call control, hence providing improved stability to the basic call control procedures of Recommendation Q.931.

6.3.1 Call related supplementary service procedures

For call related supplementary service procedures initiated at call establishment or call clearing, the procedures for call control as specified in 5/Q.931 and 6/Q.931 are utilized. This enables, for example, the originating user to send a supplementary service invocation within a SETUP message and to receive from the remote user a return result, return error, or reject component type in the Facility information element within an ALERTING message, CONNECT message, or any other appropriate message from the service provider.

For call related supplementary service invocations during the Active state of a call, the FACILITY message is used for the exchange of the Facility information elements over the existing signalling connection. This signalling connection is identified by the call reference of the corresponding active call.

The call reference provides the means to correlate FACILITY messages belonging to the same signalling transaction. In the case of call related invocations, the call reference correlates the call with the appropriate supplementary service transaction. When a supplementary service affects more than one call, different call references are used to identify each call individually.

If a call related FACILITY message is sent using the call reference of a call in progress or of an active call, and this call is cleared due to call related causes, then the treatment of any outstanding supplementary service requests is dependent on the requirements of each individual service as specified in the Q.95x-Series Recommendations.

Additionally, the following guidelines apply:

1) A supplementary service functional protocol (using the Facility information element) may use an existing bearer associated call reference if it is to be coupled to the bearer, or it may use a call reference not associated with a bearer.

2) The implicit association provided by a Q.931 call reference shall always be cleared when a bearer connection is released.

3) If a bearer connection and a bearer unrelated call reference need to be associated at the receiving end, then the bearer unrelated protocol should include a request for the terminating end to associate the two call references.

6.3.2 Bearer connection independent supplementary service procedures

This subclause defines the transport functions employed for operations independent of a bearer connection. These transport functions are provided at the user-network interface by means of message exchange according to Recommendations Q.931 and Q.932 and utilize the data link services as described in Recommendation Q.921. The messages used for transport (i.e. REGISTER, FACILITY, RELEASE COMPLETE) carry the application oriented Facility information elements containing the operation components. The correlation among the various transport messages is provided by means of the call reference value of each message.

For general rules, format and coding of call reference values see 4.3/Q.931.

The bearer connection independent transport functions are divided into the following three categories:

– Point-to-point, connection-oriented;

– Point-to-point, connectionless;

– Broadcast, connectionless.

6.3.2.1 Point-to-point transport

Before these procedures are invoked, a reliable data link connection must be established between the user and the network. All messages shall be sent to the data link layer using a DL-DATA request primitive.

6.3.2.1.1 Connection oriented transport-Connection establishment

The initiator shall begin the establishment of the signalling connection by sending a REGISTER message to the responder and enter the Call Independent Service call state (U/N31). The responder upon receiving the REGISTER message, shall also enter the Call Independent Service call state (U/N31). Note that either the user or the network may assume the role of the initiator.

The signalling connection is identified by the call reference included in the REGISTER message. The call reference value shall be chosen in compliance to the procedures of 4.3/Q.931.

6.3.2.1.2 Data transfer phase

After its establishment the signalling connection can be used to exchange data between the user and the network involved in the connection. The user and the network are completely free to send data, i.e. there exists no predetermined sending scheme.

Data is transferred by sending a FACILITY message to the peer entity. Sending a FACILITY message shall not affect the call state.

The call reference identifying this connection shall be included in this FACILITY message.

The data, e.g. the component structures, shall be included in the Facility information element.

6.3.2.1.3 Connection release

The signalling connection may be released by the initiator or the responder of the REGISTER message establishing the connection. Releasing the connection shall be accomplished by sending a RELEASE COMPLETE message. The Cause information element shall indicate cause #16 “normal call clearing”.

The call reference identifying this connection shall be included in the RELEASE COMPLETE message.

After sending the RELEASE COMPLETE message, the sender shall release the used call reference and enter the Idle call state (U/N0).

On receipt of the RELEASE COMPLETE message, the receiver shall release the used call reference and enter the Idle call state (U/N0).

6.3.2.2 Connectionless transport

Where a point-to-point data link is known to exist, a connectionless protocol can be used by the network or the user. The connectionless protocol is based on FACILITY messages as well. However, the connectionless protocol shall only use the dummy call reference value as specified in 4.3/Q.931.

The FACILITY message and the Facility information element within it are used to carry the “user” information, e.g. the component structures in the Facility information element.

The contents of the FACILITY message may be extended by Called party number and Called party sub-address information elements. The specific requirements are the subject of individual supplementary service recommendations.

6.3.2.3 Broadcast transport mechanism connectionless

The broadcast connectionless protocol is based on FACILITY messages which are sent from the network to the user. The broadcast connectionless network protocol shall only use the dummy call reference value as specified in 4.3/Q.931.

The network shall send this FACILITY message, using the DL-UNIT DATA request service primitive and a TEI parameter equal to 127.

The contents of the FACILITY message may be extended by Called party number and Called party sub-address information elements. If one or more of these information elements are included in the FACILITY message, the receiving user shall check the identity according to B.3/Q.931 treating the FACILITY message similar to the SETUP message.

The application data shall be included in the Facility information element.

6.3.3 Responses to multiple supplementary service invocations

The correlation of responses to multiple supplementary service invocations is based on call references and invoke identifiers.

6.3.4 Coding of the call reference

For general rules, format and coding of call reference values, 4.3/Q.931 is applicable.

6.3.5 Formal definition of data types

Formal definition of data types to be used within the operations to be coded in the Facility information element are provided in Recommendation X.229 [10] (Remote Operations, Protocol Specification). An extract of the relevant clauses of X.229 is provided in Appendix IV.

6.3.6 Error procedures

In general, the error handling procedures specified in 5.8/Q.931 apply with the modification that in items a) and d) of 5.8.3.2/Q.931, "SETUP" shall be replaced with "SETUP, REGISTER".

Additional error handling required specifically for the common information element procedures are specified in the following subclauses.

6.3.6.1 Component related errors

If a facility information element is received with an invalid protocol profile in any message other than REGISTER, the procedures specified in 5.8.6/Q.931 and 5.8.7/Q.931 will apply as appropriate except that for connectionless transport, no STATUS message shall be returned.

If a network or user which implements procedures given in 6.3 receives a Facility information element containing an invoke component indicating an operation that is not recognized, i.e. a particular supplementary service or function has not been implemented, then a Facility information element containing a reject component reporting the general problem "unrecognized operation" shall be returned to the sending entity. This rejection will not affect the handling of the message in which the Facility information element was included or of other information elements included in that message.

If the operation value in the Facility information element is understood but it is not defined to be sent in the message in which it was received, then a return error component with the value "procedural error" (see Recommendation Q.950) will be returned.

Other errors specific to individual supplementary services are treated according to procedures provided in Q.95x-Series Recommendations.

6.3.6.2 Transport related errors

If a FACILITY message is received and it does not contain the Facility or Extended facility information element, the procedures specified in 5.8.6/Q.931 will apply except that for connectionless transport, no STATUS message shall be returned.

6.3.6.3 Call related errors

If the network or user recognizes a supplementary service in a SETUP message but is not able to process the requested operation, then the following operations apply:

1) the network or user may clear the call request and reject the supplementary service invocation by means of a RELEASE COMPLETE message which contains the Cause information element and the return error or reject component type with the appropriate parameters in the Facility information element;

2) the network or user may continue to process the call request according to normal Recommendation Q.931 call controll procedures, and reject the supplementary service invocation by including a return error or reject component type with an appropriate data element in the Facility information element by means of a FACILITY message or in any appropriate Recommendation Q.931 message;

3) the network or user may continue to process the call request according to the Recommendation Q.931 call control procedures, and ignore the supplementary service invocation.

The option to be used depends on the individual supplementary service procedures, which are the subject of the Q.95x-Series Recommendations.

The Cause information element in Q.931 call control messages will be used to report Q.931 errors outside the component portion of the Facility information element (octets 1-3). When no Q.931 protocol error is found, the Cause information element will convey cause #31 – normal, unspecified. Protocol errors in the component portion of the Facility information element (octets 4-?) will be reported in a Reject component carried in a Facility element.

If the call related FACILITY message is sent using the call reference of a call in progress or of an active call, and this call is to be cleared due to call related causes, then depending upon the supplementary service invoked, one of the following will occur:

– the network or user may retain both the connection and the call reference association and may send a response within a Facility information element in a FACILITY message prior to the initiation of the normal call clearing procedures; or

– the network or user may send a response within a Facility information element in the first clearing message (i.e. DISCONNECT, RELEASE, or RELEASE COMPLETE message); or

– the network or user may continue with the clearing procedures.

In the third option, if the signalling connection is cleared while a supplementary service related request is pending, handling of the outstanding request will be according to the Q.95x-Series Recommendations.

If a data link reset or data link failure occurs and a supplementary service request is outstanding, the procedures specified in 5.8.8/Q.931 and 5.8.9/Q.931 will apply, respectively. The procedures associated with the treatment of the outstanding supplementary service requests in this case are for further study.

6.3.6.4 Call independent errors

If a REGISTER message is received indicating a call reference value that is currently in use, it shall be ignored and a STATUS message with a Cause information element indicating Cause value #101 “message not compatible with call state” and a Call state information element indicating the appropriate call state shall be returned.

Only the FACILITY message, RELEASE COMPLETE message, STATUS message, and the STATUS ENQUIRY message shall be sent using the call reference that was assigned by a REGISTER message. If any other message is received, it should be ignored and a STATUS message with CCITT cause value #101 “message not compatible with call state”, and a Call state information element indicating call state 31 “call independent service” state shall be retumed.

If a Facility information element is received with an invalid protocol profile in a REGISTER message, the contents of the REGISTER message shall be discarded and a RELEASE COMPLETE message containing cause #100 “invalid information element contents” shall be returned.

If either protocol entity receives an indication that the data link has been released via the DL-RELEASE indication primitive, it shall release the call reference, enter the idle call state (U0, N0) and regard the signalling connection as released.

If either protocol entity receives an indication that the data link has spontaneously been reset via the DL-ESTABLISH indication primitive, it shall send a RELEASE COMPLETE message with the appropriate call reference, and with the Cause information element indicating cause value #41 “temporary failure”, release the call reference, enter the Idle call state (U0, N0) and regard the signalling connection as released.

If a protocol error occurs, either the network or the user may release the signalling connection by sending a RELEASE COMPLETE message. The call reference identifying this connection shall be included in the RELEASE COMPLETE message. The cause value as indicated by the Cause information element will be dependent on the error case. After sending the RELEASE COMPLETE message, the sender shall release the used call reference and enter the Null call state (U/N0). On receipt of the RELEASE COMPLETE message, the receiver shall release the used call reference and enter the Null call state (U/N0).

6.4 Network side channel reservation function

The network side channel reservation function allows the user to improve the success of a subsequent channel selection by making a channel unavailable for use by another user on the same access. The user is identified by the data link connection endpoint identifier (CEI). Reservations may only be used by a call associated with the same CEI (and thus the same user).

Two methods of reservation are defined: Implicit and Explicit reservation. Both methods may coexist on the same user access configuration.

6.4.1 Implicit reservation

Implicit reservation allows user control of the network side channel reservation function by means of invocation of other functions from the user, e.g. by use of the Hold and Retrieve functions.

6.4.1.1 Reservation creation

On creation of a reservation the network shall reserve a channel against a specified data link connection endpoint identifier (CEI). This reservation renders a channel busy, such that another call may be rejected or enter a call waiting condition, even though not all channels are allocated to active calls.

NOTE 1 – A call allocated to a CEI with a channel selected is a call:

1) In one of the following states: Overlap Sending (N2), Outgoing Call Proceeding (N3); Call Delivered (N4) with an Idle or Hold Request auxiliary state; Active (N10) with an Idle or Hold Request auxiliary state; Suspend Request (N15).

   Appropriate states for point-to-point case are for further study.

2) In one of the following states where that state was reached via one of the states listed in (1): Disconnect Request (N11); Disconnect Indication (N12); Release Request (N19).

The following actions shall create a reservation:

a) If no reservation already exists, and if no other existing call for that CEI has a channel selected, then the receipt or sending of a HOLD ACKNOWLEDGE message shall create a reservation against the CEI for which that message was received or sent.

b) If an existing call (1) has a channel selected, and if no reservation already exists, and if an existing call (2) is in the Call Held or Retrieve Request auxiliary state, and if no other existing call for that CEI has a channel selected, then the receipt or sending of a RELEASE COMPLETE message for the call (1) shall create a reservation against the CEI for which that message was received or sent.

c) If an existing call (1) has a channel selected, and if no reservation already exists, and if an existing call (2) is in the Call Held or Retrieve Request auxiliary state, and if no other existing call for that CEI has a channel selected, then the sending of a SUSPEND ACKNOWLEDGE message for the call (1) shall create a reservation against the CEI for which that message was received or sent.

   NOTE 2 – On call suspension, the call is allocated to a call identity and not to a particular terminal, as identified by its CEI.

d) If

   – no reservation already exists;

   – and if an existing call is in the Call Held or Retrieve Request auxiliary state;

   – and if no other existing call for the CEI has a channel selected;

   then the receipt or sending of a RESTART ACKNOWLEDGE message, when the Restart indicator information element contained in the original RESTART message specified “indicated channels” where that channel is allocated to an existing call, shall create a reservation against the CEI for which that message was received or sent.

NOTE 3 – The network or the user may have already begun clearing the call by means of a DISCONNECT, RELEASE and RELEASE COMPLETE sequence of message, in which case item b) applies.

6.4.1.2 Reservation use

On use of a reservation the network shall perform the associated channel selection procedures applicable to the events concerned and delete the reservation against that CEI. The following actions shall use a reservation, if such a reservation exists against the CEI for which action was performed:

a) The sending of a SETUP ACKNOWLEDGE, CALL PROCEEDING, ALERTING or CONNECT message to the user in response to a received SETUP message.

b) The sending of a CONNECT ACKNOWLEDGE message to the user in response to a received CONNECT message.

NOTE – If the user wishes to retain the reservation in this case for some future outgoing call, then either the CONNECT message should be delayed, or explicit reservation should be used.

c) The sending of a RETRIEVE ACKNOWLEDGE message in response to a received RETRIEVE message.

d) The reception of a RETRIEVE ACKNOWLEDGE message in response to the sending of a RETRIEVE message.

6.4.1.3 Reservation cancellation

On reservation cancellation the network shall delete the reservation against that CEI. The following actions shall cancel a reservation if such a reservation exists against the CEI for which action was performed.

a) If only one call related to a specified CEI is in the Call Held or Retrieve Request auxiliary state, the sending or receipt of a RELEASE COMPLETE message for a call which is in the Call Held or Retrieve Request auxiliary state.

b) The sending of a RESUME ACKNOWLEDGE message.

NOTE – The reservation is not used for call resumption, as a channel is permanently allocated to a suspended call.

c) The sending or receipt of a RESTART ACKNOWLEDGE message when the Restart indicator information element contained in the original RESTART message specified “single interface” or “all interfaces”.

d) The receipt of a DL-RELEASE indication primitive.

6.4.2 Explicit reservation

Explicit channel reservation provides user control of the network side channel reservation function by explicit operations which may use a reservation indicator generated, issued and managed by the network. This allows the user to reserve B-channel resources for use by several held calls.

NOTE – This subclause provides operations for the creation, management and cancellation of reservations which the user may use in parallel with the procedures specified in 6.4.1.

6.4.2.1 Explicit reservation control

To control channel reservation explicitly the user shall include an explicit reservation creation control invoke component carried by a Facilty information element in an appropriate call related transport message.

The invoke component may contain an argument specifying one of three options:

i) no reservation required;

ii) reservation required without reservation indicator;

iii) reservation required with reservation indicator.

If no parameter is included in the invoke component, then “reservation required without reservation indicator” is assumed by the network.

If the network is able to provide the requested function, the network shall include an explicit reservation creation control return result component carried by a Facilty information element in an appropriate transport message related to the same call. If appropriate and required by the user (by a subscription parameter, or as requested in the invoke component), the network shall include a reservation indicator parameter; if this parameter is provided, the network shall store this parameter against the reservation, and shall only grant use of the reservation when this reservation indicator value is included in the explicit reservation management invoke component.

If the network is able to provide the requested reservations, and the explicit reservation creation control invoke component was included in a message that released a channel resource, or is about to release a channel resource on acknowledgement (i.e. a RELEASE, a RELEASE COMPLETE, a HOLD or a HOLD ACKNOWLEDGE message), that channel resource shall not be allocated to another call if it is required to meet the reservation requirements thus provided.

If the network is unable to provide the requested function, the network shall include an explicit reservation creation control return error component carried by a Facility information element in an appropriate transport message related to the same call. Appropriate errors are:

- maximum number of reservations reached. The maximum number of reservations (default = one) already exists for this CEI;
- function not available;
- function not subscribed;
- unwanted reservation created.

On receipt of the explicit reservation creation control return result component, the user shall retain knowledge of the reservation indicator, if provided.

6.4.2.2 Explicit reservation management

To manage the use of channel reservation the user shall include an explicit reservation management invoke component carred by a Facility information element in an appropriate call related transport message performing channel selection (i.e. SETUP, SETUP ACKNOWLEDGE, CALL PROCEEDING, ALERTING, CONNECT, CONNECT ACKNOWLEDGE, RETRIEVE or RETRIEVE ACKNOWLEDGE). If required by the network (on all reservations as indicated by a subscription parameter, or as requested in the explicit reservation creation control invoke component which generated the reservation), the user shall include a reservation indicator parameter since the network shall only grant use of the reservation when the required reservation indicator value is included in the explicit reservation management invoke component.

If no explicit reservation management invoke component is included in a call control message selecting a channel, then any existing implicit reservations shall be used by the network. If no implicit reservations exist, then any existing explicit reservations shall remain available.

If the user requires that an existing implicit reservation is not to be used by the call control message selecting a channel, then the user shall include an indicator in an explicit reservation management invoke component.

If the network is able to provide the requested management function, the network shall include an explicit reservation management return result component carried by a Facility information element in an appropriate transport message related to the same call.

If the network is unable to provide the requested function, the network shall include an explicit reservation management return error component carried by a Facility information element in an appropriate transport message related to the same call. Appropriate errors are:

- no explicit reservation exists, or invalid reservation indicator;
- function not available;
- functon not subscribed – implicit reservation used.

NOTE – Failure of the explicit reservation management operation will not necessarily result in channel selection failure, solely in failure to manage the reservation.

On receipt of the explicit reservation management return result component, the user shall remove knowledge of the reservation indicator, if used.

6.4.2.3 Explicit reservation cancellation

To cancel an explicit channel reservation the user shall include an explicit reservation cancel invoke component carried by a Facility information element in an appropriate call related transport message.

If the network is able to cancel the reservation, the network shall include an explicit reservation cancel return result component carried by a Facility information element in an appropriate transport message related to the same call. If required by the network (by a subscription parameter, or as requested in the invoke component), the user shall include a reservation indicator parameter and the reservation shall only be cancelled if this reservation indicator value is included in the explicit reservation cancel invoke component.

Reservation cancellation shall only cancel a single reservation; if multiple reservations exist, multiple reservation cancellations must be invoked.

If the network is unable to cancel the reservation, the network shall include an explicit reservation cancel return error component carried by a Facility information element in an appropriate transport message related to the same call. Appropriate errors are:

- no explicit reservation exists, or invalid reservation indicator;
- function not available;
- function not subscribed.

On receipt of the explicit reservation cancel return result component, the user shall remove knowledge of the reservation indicator, if used.

The network shall cancel all reservations on:

- the sending or receipt of a RELEASE COMPLETE message for the last call on that CEI;
- the sending or receipt of a RESTART ACKNOWLEDGE message when the Restart indicator information element contained in the original RESTART message specified "single interface" or "all interfaces";
- the receipt of a DL-RELEASE indication primitive.

There is no signalling protocol specific to the reservation function associated with this action. The user shall likewise remove all knowledge of any reservation indicators.

6.4.2.4 Formal definition

The formal definition of the explicit network side channel reservation function shall be as shown in Table 6-1.

6.4.2.5 Effect of reservation on channel selection for a new call

For a new incoming call to the access configuration, if a channel is being selected, the "no B-channel available" condition is used if the number of channels available for use by a terminal, minus the number of, if any, any reserved channels, is zero. An implicitly reserved channel should not be used unless the call is subsequently selected for that CEI. An explicitly reserved channel shall not be used unless an explicit request to use that channel, containing the appropriate reservation indicator value if required, is received using that CEI.

6.4.2.6 Interaction between implicit and explicit network side channel reservation functions on the same CEI

When an implicit reservation exists at the same time as an explicit reservation, then all call control messages which affect channel selection and do not contain an explicit reservation invoke component shall follow the procedures for implicit reservation in 6.4.1.

6.4.3 Effect of reservation on channel selection for a new call

For a new incoming call to the access configuration, if a channel is being selected, the "no B-channel available" condition is used if the number of channels available minus the number of reserved channels, is zero. A reserved channel should not be used, unless the call is subsequently selected for that CEI.

TABLE 6-1/Q.932

Explicit network controlled channel reservation

```
Explicit-Network-Controlled-Channel-Reservation

{ ccitt recommendation q 932 explicit-network-controlled-channel-reservation(4) }

DEFINITIONS    ::=

BEGIN

        IMPORTS

        OPERATION, ERROR

        FROM Remote-Operation-Notation { joint-iso-ccitt remote-operations(4) notation(0) }
        userNotSubscribed,
        notAvailable,
        FROM General-Errors { ccitt recommendation q 950 general-errors-list(1) };

ExplicitReservationCreationControl      ::=     OPERATION

            ARGUMENT controlOption ENUMERATED {
               noReservationRequired (0),
               reservationRequiredWithReservationIndicator (1),
               reservationRequiredWithoutReservationIndicator (2) }

        RESULT   ReservationIndicator -- optional
        ERRORS {     maximumNumberOfReservationsReached,
                     userNotSubscribed, notAvailable,
                     unwantedReservationCreated }

ExplicitReservationManagement ::= OPERATION

        ARGUMENT ReservationIndicator -- optional
        RESULT
        ERRORS {     noExplicitReservationExistsOrInvalidReservationIndicator,
                     userNotSubscribed,
                     notAvailable,
                     implicitReservationUsed }

ExplicitReservationCancel     ::=     OPERATION
        ARGUMENT ReservationIndicator -- optional
        RESULT
        ERRORS {     noExplicitReservationExistsOrInvalidReservationIndicator,
                     userNotSubscribed,
                     notAvailable }

MaximumNumberOfReservationsReached ::= ERROR

NoExplicitReservationExistsOrInvalidReservationIndicator ::= ERROR

UnwantedReservationCreated ::= ERROR

ImplicitReservationUsed ::= ERROR

explicitReservationCreationControl                      ::= 20

explicitReservationManagement                           ::= 21

explicitReservationCancel                               ::= 22

maximumNumberOfReservationsReached                      ::= 33

noExplicitReservationExistsOrInvalidReservationIndicator    ::= 34

unwantedReservationCreated                              ::= 35

implicitReservationUsed                                 ::= 36

ReservationIndicator                                    ::= INTEGER (-128, 127)

END -- of Q.932 explicit network controlled channel reservation definitions
```

7 Message functional definition and content

Message definitions given in 3/Q.931 will apply with additions that:

- Facility or Extended Facility information element may optionally be included in any of the Call establishment or Call clearing messages as well as the REGISTER, FACILITY and the HOLD/RETRIEVE set of messages defined in this Recommendation in either direction.

- Feature activation information element may optionally be included in the SETUP, INFORMATION messages in the user to network direction.

- Feature indication information element may optionally be included in any Call establishment or Call clearing message as well as the INFORMATION message in the network to user direction.

- Information request information element may optionally be included in the SETUP ACKNOWLEDGE or INFORMATION messages in the network to user direction.

- Notification indicator information element may optionally be included in Call establishment or Call clearing messages as well as the FACILITY and NOTIFY messages in either direction.

- Service profile identification information element may optionally be included in the INFORMATION message.

- Endpoint identifier information element may be included in the SETUP message.

7.1 Messages for supplementary service control

Table 7-1 summarizes the messages specific to supplementary service control procedures.

TABLE 7-1/Q.932

Messages specific to supplementary service control

	Reference
FACILITY	7.1.1
HOLD	7.1.2
HOLD ACKNOWLEDGE	7.1.3
HOLD REJECT	7.1.4
REGISTER	7.1.5
RETRIEVE	7.1.6
RETRIEVE ACKNOWLEDGE	7.1.7
RETRIEVE REJECT	7.1.8

7.1.1 FACILITY

This message may be sent to request or acknowledge a supplementary service. The supplementary service to be invoked, and its associated parameters, are specified in the Facility information element (see Table 7-2).

For the use of this message, see clause 6.

TABLE 7-2/Q.932

FACILITY message content

Message type: FACILITY Significance: local (Note 1) Direction: both				
Information element	Reference	Direction	Type	Length
Protocol discriminator	4.2/Q.931	both	M	1
Call reference	4.3/Q.931	both	M	2-*
Message type	8.1/Q.932	both	M	1
Facility	8.2/Q.932	both	M (Note 4)	8-*
Display	4.5/Q.931	n → u	O (Note 2)	(Note 3)

M Mandatory
O Optional

NOTES
1 This message has local significance; however, it may carry information of global significance.
2 Included if the network provides information that can be presented to the user.
3 The minimum length is 2 octets. The maximum length is network dependent and is either 34 or 82 octets.
4 Extended facility information element may be used instead.

7.1.2 HOLD

This message is sent by the network or the user to request the Hold function for an existing call (see Table 7-3).

For the use of this message, see clause 6.

TABLE 7-3/Q.932

HOLD message content

Message type: HOLD Significance: local Direction: both				
Information element	Reference	Direction	Type	Length
Protocol discriminator	4.2/Q.931	both	M	1
Call reference	4.3/Q.931	both	M	2-*
Message type	8.1/Q.932	both	M	1
Display	4.5/Q.931	n → u	O (Note 1)	(Note 2)

NOTES
1 Included if the network provides information that can be presented to the user.
2 The minimum length is 2 octets. The maximum length is network dependent and is either 34 or 82 octets.

7.1.3 HOLD ACKNOWLEDGE

This message is sent by the network or the user to indicate that the Hold function has been successfully performed (see Table 7-4).

For the use of this message, see clause 6.

TABLE 7-4/Q.932

HOLD ACKNOWLEDGE message content

Message type: HOLD ACKNOWLEDGE Significance: local Direction: both				
Information element	Reference	Direction	Type	Length
Protocol discriminator	4.2/Q.931	both	M	1
Call reference	4.3/Q.931	both	M	2-*
Message type	8.1/Q.932	both	M	1
Display	4.5/Q.931	n → u	O (Note 1)	(Note 2)

NOTES

1 Included if the network provides information that can be presented to the user.

2 The minimum length is 2 octets. The maximum length is network dependent and is either 34 or 82 octets.

7.1.4 HOLD REJECT

This message is sent by the network or the user to indicate the denial of a request to hold a call (see Table 7-5).

For the use of this message, see clause 6.

TABLE 7-5/Q.932

HOLD REJECT message content

Message type: HOLD REJECT Significance: local Direction: both				
Information element	Reference	Direction	Type	Length
Protocol discriminator	4.2/Q.931	both	M	1
Call reference	4.3/Q.931	both	M	2-*
Message type	8.1/Q.932	both	M	1
Cause	4.5/Q.931	both	M	4-32
Display	4.5/Q.931	n → u	O (Note 1)	(Note 2)

NOTES

1 Included if the network provides information that can be presented to the user.

2 The minimum length is 2 octets. The maximum length is network dependent and is either 34 or 82 octets.

7.1.5 REGISTER

This message is sent by the user or the network to assign a new call reference for non-call associated transactions (see Table 7-6).

For the use of this message, see clause 6.

TABLE 7-6/Q.932

REGISTER message content

Message type: REGISTER Significance: local (Note 1) Direction: both				
Information element	Reference	Direction	Type	Length
Protocol discriminator	4.2/Q.931	both	M	1
Call reference	4.3/Q.931	both	M	2-*
Message type	8.1/Q.932	both	M	1
Facility	8.2/Q.932	both	O (Note 4)	2-*
Display	4.5/Q.931	n → u	O (Note 2)	(Note 3)

NOTES

1 This message has local significance; however, it may carry information of global significance.

2 Included if the network provides information that can be presented to the user.

3 The minimum length is 2 octets. The maximum length is network dependent and is either 34 or 82 octets.

4 Included if the network or the user provides supplementary service information.

7.1.6 RETRIEVE

This message is sent by the network or the user to request the retrieval of a held call (see Table 7-7).

For the use of this message, see clause 6.

TABLE 7-7/Q.932

RETRIEVE message content

Message type: RETRIEVE Significance: local Direction: both				
Information element	Reference	Direction	Type	Length
Protocol discriminator	4.2/Q.931	both	M	1
Call reference	4.3/Q.931	both	M	2-*
Message type	8.1/Q.932	both	M	1
Channel identification	4.5/Q.931	both	O (Note 1)	2-*
Display	4.5/Q.931	n → u	O (Note 2)	(Note 3)

NOTES

1 If not included, its absence is interpreted as any channel acceptable.

2 Included if the network provides information that can be presented to the user.

3 The minimum length is 2 octets. The maximum length is network dependent and is either 34 or 82 octets.

7.1.7 RETRIEVE ACKNOWLEDGE

This message is sent by the network or the user to indicate that the Retrieve function has been successfully performed (see Table 7-8).

For the use of this message, see clause 6.

TABLE 7-8/Q.932

RETRIEVE ACKNOWLEDGE message content

Message type: RETRIEVE ACKNOWLEDGE Significance: local Direction: both				
Information element	Reference	Direction	Type	Length
Protocol discriminator	4.2/Q.931	both	M	1
Call reference	4.3/Q.931	both	M	2-*
Message type	8.1/Q.932	both	M	1
Channel identification	4.5/Q.931	both	O (Note 1)	2-*
Display	4.5/Q.931	n → u	O (Note 2)	(Note 3)

NOTES

1 Mandatory in all cases except when the sender accepts the specific B-channel indicated in the RETRIEVE message. If included, a channel is indicated and specified as exclusive.

2 Included if the network provides information that can be presented to the user.

3 The minimum length is 2 octets. The maximum length is network dependent and is either 34 or 82 octets.

7.1.8 RETRIEVE REJECT

This message is sent by the network or the user to indicate the inability to perform the requested Retrieve function (see Table 7-9).

For the use of this message, see clause 6.

TABLE 7-9/Q.932

RETRIEVE REJECT message content

Message type: RETRIEVE REJECT Significance: local Direction: both				
Information element	Reference	Direction	Type	Length
Protocol discriminator	4.2/Q.931	both	M	1
Call reference	4.3/Q.931	both	M	2-*
Message type	8.1/Q.932	both	M	1
Cause	4.5/Q.931	both	M	4-32
Display	4.5/Q.931	n → u	O (Note 1)	(Note 2)

NOTES

1 Included if the network provides information that can be presented to the user.

2 The minimum length is 2 octets. The maximum length is network dependent and is either 34 or 82 octets.

8 General message format and information element coding

This clause should be read in conjunction with 4/Q.931 and contains the coding of the information elements specifically used by the procedures described in this Recommendation.

8.1 Message type

The additional codings are defined in Table 8-1 for message type.

TABLE 8-1/Q.932

Q.932 message types

Bits								
8	7	6	5	4	3	2	1	
0	0	1	-	-	-	-	-	(Q.931 call information phase message group)
			0	0	1	0	0	HOLD
			0	1	0	0	0	HOLD ACKNOWLEDGE
			1	0	0	0	0	HOLD REJECT
			1	0	0	0	1	RETRIEVE
			1	0	0	1	1	RETRIEVE ACKNOWLEDGE
			1	0	1	1	1	RETRIEVE REJECT
0	1	1	-	-	-	-	-	(Q.931 miscellaneous message group)
			0	0	0	1	0	FACILITY
			0	0	1	0	0	REGISTER

8.2 Other information elements

These information elements are coded according to the general coding rules as defined in 4.5.1/Q.931.

NOTE – The value used for Protocol discriminator shall be as defined for messages used in Recommendation Q.931.

Table 8-2 contains the codepoints allocated to the information elements defined in this Recommendation.

8.2.1 Call state

The Call state information element is coded as shown in Figure 4-13/Q.931 and Table 4-7/Q.931. Table 8-3 contains additional codepoint(s) required for use within supplementary service control.

8.2.2 Endpoint identifier

The purpose of the Endpoint identifier information element is:

– to indicate the user service identifier and terminal identifier for the purpose of terminal identification; and

– to indicate a specific terminal for the purpose of terminal selection.

(See Annex A for the associated procedures.)

TABLE 8-2/Q.932

Information elements specific to supplementary service control

Bits 8 7 6 5 4 3 2 1		Subclause reference	Maximum length (octets) (Note 1)
0 : : : : : : :	*Variable length information elements:*		
0 0 0 1 1 0 1	Extended facility	8.2.4	(Note 4)
0 0 1 1 1 0 0	Facility	8.2.3	(Note 3)
0 0 1 0 1 0 0	Call state	8.2.1	3
0 1 1 0 0 1 0	Information request	8.2.7	3
0 1 0 0 1 1 1	Notification indicator	8.2.8	(Note 4)
0 1 1 1 0 0 0	Feature activation	8.2.5	4
0 1 1 1 0 0 1	Feature indication	8.2.6	5
0 1 1 1 0 1 0	Service profile identification	8.2.9	32
0 1 1 1 0 1 1	Endpoint identifier	8.2.2	4
All other values are reserved (Note 2)			

NOTES

1 The length limits described for the variable length information elements below take into account only the present CCITT standardized coding values. Future enhancements and extensions to this Recommendation will not be restricted to these limits.

2 The reserved values with bits 5-8 coded "0000" are for future information elements for which comprehension by the receiver is required (see 5.8.7.1/Q.931).

3 The maximum length of the Facility information element is application dependent consistent with the maximum length of the message.

4 The maximum length of this information element is network dependent.

TABLE 8-3/Q.932

Call state information element

Call state value (octet 3)	
Bits	
6 5 4 3 2 1	
0 1 1 1 1 1	Call independent service

The Endpoint identifier information element is coded as shown in Figure 8-1 and Table 8-4.

The default maximum length of the Endpoint identifier information element is four octets.

Bits								
8	7	6	5	4	3	2	1	Octets
	Endpoint identifier							
0	0	1	1	1	0	1	1	1
	Information element identifier							
Length of endpoint identifier contents								2
1 ext.	User service identifier							3
1 ext.	Interpreter		Terminal identifier					4[a]

a) This octet is optional.

FIGURE 8-1/Q.932

Endpoint identifier information element

TABLE 8-4 Q.932

Endpoint identifier information element

User service identifier (USID) (octet 3)

The USID is a selection parameter which identifies a group of terminals on an interface which share a common service profile and which may be addressed together. Upon receipt of this element, a terminal will consider itself as being addressed if the value received matches its stored value or if the value received is coded as all "1"s (127). When USID is coded as 127, octet 4 is not used.

Interpreter (octet 4)

Bit 7 of octet 4 indicates how a terminal is to interpret the TID field received. When set to "0", the terminal is being addressed only if the TID matches (see TID definition following). When set to "1", the terminal is being addressed only if the TID received is not 63 and does not match. In the user-to-network direction, this bit is set to "0".

Terminal identifier (TID) (octet 4)

The TID is a selection parameter which identifies a single terminal within a group designated by a USID value. For USID = 127, the TID does not apply. Upon receipt to this field, a terminal will consider itself addressed if one of the following is true:

- the interpreter bit = "0" and the value received matches the terminal's stored value;
- the interpreter bit = "1" and the value received does not match the terminal's stored value;
- the value received is coded all "1"s (63).

8.2.3 Facility

This subclause defines only the structure and the coding of the Facility information element. Specific procedures describing individual supplementary services are provided in Q.95x-Series Recommendations.

The purpose of the Facility information element is to indicate the invocation and operation of supplementary services, identified by the corresponding operation value within the Facility information element. The Facility information element is defined in Figures 8-2 and 8-4, Figures IV.1 and IV.2 and Tables 8-5, 8-7 and 8-8 and IV.2 through IV.12.

The Facility information element may be repeated in a given message.

The maximum length of the Facility information element is application dependent consistent with the maximum length of the message.

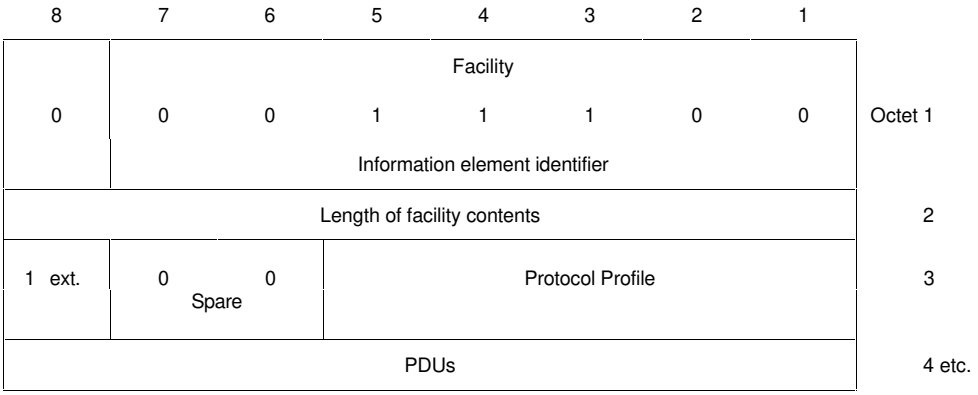


FIGURE 8-2/Q.932

Facility information element

TABLE 8-5/Q.932

Facility information element

Protocol Profile
Bits
5 4 3 2 1
1 0 0 0 1 Remote Operations Protocol
1 0 0 1 0 CMIP Protocol (see Recommendation Q.941 [11])
1 0 0 1 1 ACSE Protocol (see Recommendations X.217 and X.227 [12])
All other values are reserved and their usage is the subject of other Recommendations.

Subclause 8.2.3.1 contains the procedures for the Remote Operations Protocol. Procedures for the use of the "CMIP Protocol" codepoint within the Facility information element are contained in the Q.940-Series Recommendations. Procedures for the use of the "ACSE Protocol" codepoint are contained in Recommendations X.217 [12] and X.227 [13] as well as Recommendation Q.941 [11]. Table 8-6 provides an example mapping between ACSE services and Q.932 messages.

TABLE 8-6/Q.932

Mapping of ACSE services

Function	ACSE Service Primitive	APDU	Q.932 messages
Setup Association	A_ASSOCIATE req.ind	AARQ	REGISTER
	A_ASSOCIATE resp.conf	AARE	FACILITY
Release Association	A_RELEASE req.ind	RLRQ	FACILITY
	A_RELEASE resp.conf	RLRE	RELEASE COMPLETE
User Abort	A_ABORT req.ind	ABRT	RELEASE COMPLETE
Internal Abort	A_P-ABORT ind	None (Internal Abort)	None (Internal Abort)
NOTE – The above mapping which corresponds to the most efficient use of Q.932 messages is not the only mapping. Alternate mappings would be applicable if, for example, AARQ PDU needs to be segmented. Detailed coding of ACSE PDUs and associated procedures are specified in Recommendations X.217 and X.227. Procedures for including other protocol PDUs within the Facility information element are for further study.			

8.2.3.1 Remote Operations Protocol

This subclause defines the PDU contents for the Protocol Profile of Remote Operations Protocol.

8.2.3.1.1 Component (octets 4, etc.)

This specification makes use of and is a subset of Recommendations X.208 [7] [Specification of Abstract Syntax Notation One (ASN.1)], X.209 [8] [Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1)], X.219 [9] (Remote operations: model notation and service definition), and X.229 [10] (Remote operations: protocol specification). An extract of the relevant clauses of Recommendations X.208 and X.209 and X.219 and are provided in Appendices III and IV, respectively. Table 8-7 provides a formal definition of the abstract syntax of the different component types.

The component, comprised of octets 4, etc., may be repeated an indefinite number of times within the Facility information element. In the case of multiple service requests, the receiving entity (user or network) shall treat the repetition of components identical to the case where multiple Facility information elements are received in a single message.

Additional requirements for initiating multiple service requests (user or network) are for further study.

NOTES

1 – Recommendation X.229 which defines the Remote Operations Service Element (ROSE) uses the term Application Protocol Data Unit (APDU) in place of component. However, since this protocol element may be applied to the support of network layer services and of application layer services, the term "component" is more appropriate in the context of this Recommendation.

2 – See Appendices III and IV for a general description of the component coding and formatting principles.

TABLE 8-7/Q.932

Facility information element component coding

```
Facility-Information-Element-Component { ccitt recommendation q 932 facility-information-element-component
(3) }

DEFINITIONS        ::=
BEGIN
EXPORTS            Component, InvokeComponent, InvokeIdentifierType;
IMPORTS            OPERATION, ERROR     FROM-Remote-Operations-Notation

{ joint iso-ccitt x 229 remote-operation(4) notation(0) }

-- Component definitions:
-- Types and values of operations and errors are defined in the
-- Q.95x-Series
-- Recommendations or elsewhere using
-- Remote Operations notation.
-- Operation values and error values are either of integer type
-- or of object identifier
-- type. If integer types are used, they shall be
-- distinct within that abstract syntax adopted in the Q.95x-Series
-- Recommendations.

Component ::=              CHOICE {

                                                    invokeComp [1] IMPLICIT InvokeComponent,
                                                    retResultComp [2] IMPLICIT ReturnResultComponent,
                                                    retErrorComp [3] IMPLICIT ReturnErrorComponent,
                                                    rejectComp [4] IMPLICIT RejectComponent }

InvokeComponent ::=                SEQUENCE {

                                                    invokeIdentifier InvokeIdentifierType,
                                                    linkedIdentifier [0] IMPLICIT
                                                    invokeIdentifierType OPTIONAL,
                                                    operationValue OPERATION,
                                                    argument ANY DEFINED BY
                                                    operationValue OPTIONAL }

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

ReturnResultComponent ::=          SEQUENCE {

                                                    invokeid invokeIdentifierType,
                                                    SEQUENCE {
                                                    operationValue OPERATION,
                                                    result ANY DEFINED BY
                                                    operationValue OPTIONAL } }

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

TABLE 8-7/Q.932 (*end*)

Facility information element component coding

```
ReturnErrorComponent ::=        SEQUENCE {
                                        invokeId InvokeIdentifierType,
                                        errorValue ERROR,
                                        parameter ANY DEFINED BY errorValue
                                        OPTIONAL }

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

RejectComponent ::=             SEQUENCE {
                                        InvokeId CHOICE {
                                                InvokeIdentifierType,
                                                NULL },
                                          problemCHOICE {
                                            [0] IMPLICIT GeneralProblem,
                                            [1] IMPLICIT InvokeProblem,
                                            [2] IMPLICIT ReturnResultProblem,
                                            [3] IMPLICIT ReturnErrorProblem } }

InvokeIdentifierType ::=        INTEGER (–32768 .. 32767)

GeneralProblem ::=              INTEGER {
-- detected by the Q.932 protocol entity
                                        unrecognizedComponent (0),
                                        mistypedComponent (1),
                                        badlyStructuredComponent (2) }

InvokeProblem ::=               INTEGER {
-- detected by particular supplementary
-- service entity
                                        duplicateInvocation (0),
                                        unrecognizedOperation (1),
                                        mistypedArgument (2),
                                        resourceLimitation (3),
                                        initiatorReleasing (4),
                                        unrecognizedLinkedId (5),
                                        linkedResponseUnexpected (6),
                                        unexpectedChildOperation (7) }

ReturnResultProblem ::=         INTEGER {
-- detected by particular supplementary
-- service entity
                                        unrecognizedInvocation (0),
                                        resultResponseUnexpected (1),
                                        mistypedResult (2) }

ReturnErrorProblem ::=          INTEGER {
-- detected by particular supplementary
-- service entity
                                        unrecognizedInvocation (0),
                                        errorResponseUnexpected (1),
                                        unrecognizedError (2),
                                        unexpectedError (3),
                                        mistypedParameter (4) }

END             -- of Q.932 Facility Information Element Component
                -- definitions
```

8.2.3.1.2 Invoke identifier tags

An invoke identifier is used to identify an operation invocation and is reflected in the return result, return error or reject component sent in response to it. Invoke identifiers used within the Facility information element have significance only within the call reference, including the dummy call reference, in which they are sent. Procedures need to be provided to resolve possible conflicts in the case of use of the dummy Call reference value in the broadcast CEI and a specific CEI simultaneously. Procedures may need to be provided to ensure that invoke identifiers are not prematurely reused during the lifetime of an operation which may result in a return error (Class 3) or a reject (Classes 3 and 5).

Procedures may also need to be provided to avoid the case of two invoke components with the same Invoke Identifier being simultaneously sent in opposite directions. In the absence of such procedures, component exchanges resulting in ambiguous reject components can occur as shown in Figure 8-3.

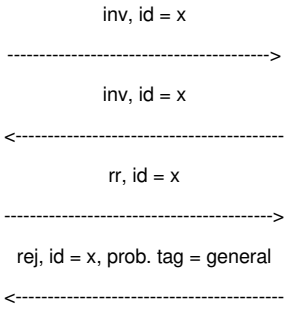


FIGURE 8-3/Q.932

8.2.3.1.3 Treatment of existing Q.931 information elements as parameters

Supplementary service protocol specifications are expected to require new parameters to be defined and to require the use of existing Q.931 information elements.

New parameters shall be defined using Recommendation X.209 coding if they do not appear elsewhere in Q.931 messages.

Supplementary service protocol specifiers may elect to encapsulate one or more existing Q.931 information elements within an X.209 data element, thereby retaining the Q.931 coding for these information elements. When this option is chosen, all the Q.931 information elements should be grouped together as the content following the Q.931 information elements tag. This is illustrated in Figure 8-4. The tag is defined in Table 8-8. This data element may appear by itself or as a member of a sequence or set as indicated in IV.6.

NOTE – Encapsulation of the Facility information element within Facility information elements shall not be used.

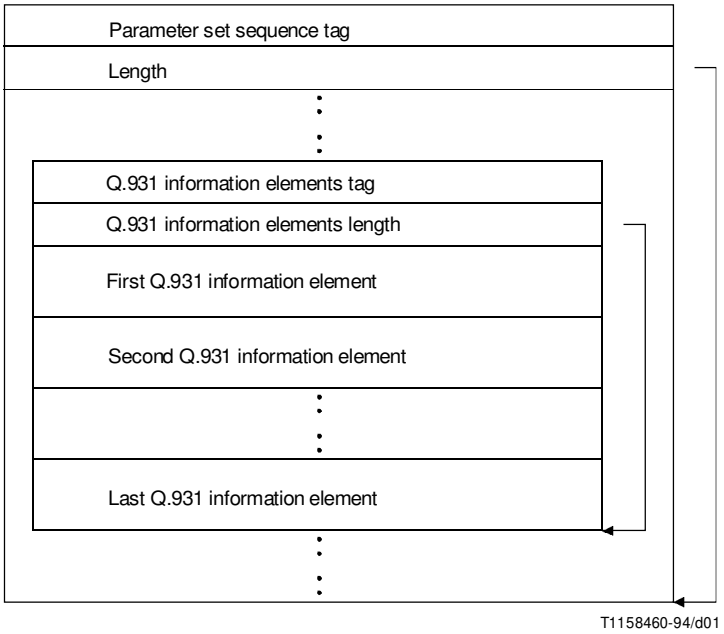


FIGURE 8-4/Q.932

Encapsulation of Q.931 information elements

TABLE 8-8/Q.932

Q.931 information elements tag

	Bits							
	8	7	6	5	4	3	2	1
Q.931 information elements	0	1	0	0	0	0	0	0
NOTE – All other values are reserved but this approach may also be applied in the future to coding structures from other Recommendations by defining other tags as required.								

Table 8-9 provides a formal definition of the Q.931 information element type.

TABLE 8-9/Q.932

```
Embedded-Q931-Types { ccitt recommendation q 932 embedded-q931-types (5) }
DEFINITIONS EXPLICIT TAGS ::=
BEGIN
EXPORTS                          Q931 InformationElement;
Q931InformationElement  ::=      [APPLICATION 0] IMPLICIT OCTET STRING
END -- of Embedded-Q931-Types
```

8.2.4 Extended facility information element

The Extended facility information element will be used when the PDUs to be included in the Facility information element have lengths such that the overall length of the Facility information element is greater than 255 octets.

Only the coding of the length of the Extended facility information is different than the Facility information element as illustrated in Table 8-10. The usage of the Extended facility information element is identical to the Facility element in all other respects and is described in 8.2.3.

TABLE 8-10/Q.932

Extended facility information element

8	7	6	5	4	3	2	1	
Extended facility								
0	0	0	0	1	1	0	1	1
0/1	Length of IE							2
1	Spare			Protocol Profile				3
PDU								4

The length of the Extended facility Information element shall be encoded as follows:

1) The length octets shall consist of one or more octets, and shall represent the number of octets of the information element.

2) For information element length less than or equal to 127 octets, the length shall consist of a single octet in which bit 8 is zero and bits 7 to 1 encode the number of octets in the information element with bit 7 as the most significant bit.

3) For information element length greater than 127, the length octets shall consist of an initial octet and one or more subsequent octets. The initial octet shall be coded as follows:

   a) bit 8 shall be one;

   b) bits 7 to 1 shall encode the number of subsequent octets in the length octets with bit 7 as the most significant bit;

   c) the value 11111111_2 shall not be used. This restriction is introduced for possible future extensions.

4) Subsequent octets of the length octets shall encode the length of the information element as follows:

   – Bits 8 to 1 of the first subsequent octet, followed by bits 8 to 1 of the second subsequent octets followed in turn by bits 8 to 1 of each further octet, up to and including the last subsequent octet shall represent an unsigned binary integer equal to the IE length with bit 8 of the first subsequent octet as the most significant bit.

   – Example: An extended facility Information element length of 201 octets can be encoded as:

1000 0001

1100 1001

8.2.5 Feature activation

The purpose of the Feature activation information element is to invoke a supplementary service as identified by the feature identifier number. The service associated with the feature identifier number is dependent on that particular user's service profile.

The maximum length of this information element is 4 octets.

The Feature activation information element is coded as shown in Figure 8-5 and Table 8-11.

Bits								
8	7	6	5	4	3	2	1	Octets
0	Feature activation 0	1	1	1	0	0	0 Information element identifier	1
Length of feature activation contents								2
0/1 ext.	Feature identifier number							3
1 ext.	Feature identifier number (continuation)							3a

FIGURE 8-5/Q.932

Feature activation information element

TABLE 8-11/Q.932

Feature activation information element

Feature identifier number (octets 3 and 3a) The feature identifier number is a unique number assigned to a feature in a customer account that is coded as part of both the Feature activation and Feature indication information elements. This number identifies the feature that is being requested or updated. The association of a particular number to a particular feature may be different for each user. Bit 8 in octet 3 is used to extend the feature identifier field. If bit 8 is 0, then another octet follows; if bit 8 is 1, then octet 3 is the last octet. The identifier numbers for a one octet field range from 1 to 127. For a multi-octet field, the order of bit values progressively decreases as the octet number increases.

8.2.6 Feature indication

The purpose of the Feature indication information element is to allow the network to convey feature indications to the user regarding the status of a supplementary service.

The maximum length of this information element is 5 octets.

The coding of the Feature indication information element is shown in Figure 8-6 and Table 8-12.

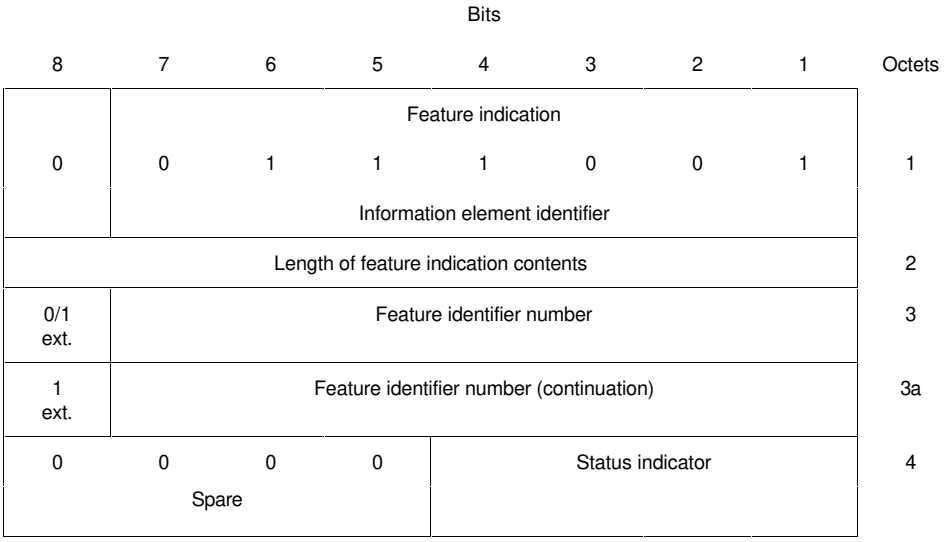


FIGURE 8-6/Q.932

Feature indication information element

TABLE 8-12/Q.932

Feature indication information element

Feature identifier number (octets 3 and 3a)

These fields are coded as described in Table 8-11.

Status indicator (octet 4)

The status indicator field identifies the current status of a supplementary service.

Bits 4 3 2 1	*Status*	*Meaning*	*Examples of possible user equipment implementation*
0 0 0 0	Deactivated	Feature is in the deactivated state	Lamp off
0 0 0 1	Activated	Feature is in the active state	Lamp steady on
0 0 1 0	Prompt	Feature prompt (waiting for user input)	Lamp steady flash
0 0 1 1	Pending	Feature is pending	Lamp steady wink

All other values are reserved.

8.2.7 Information request

The purpose of the Information request information element is to provide the capability for requesting additional information and signalling completion of the information request.

The Information request information element is coded as shown in Figure 8-7 and Table 8-13.

The default maximum length of the Information request information element is three octets.

Bits								
8	7	6	5	4	3	2	1	Octets
	Information request							
0	0	1	1	0	0	1	0	1
	Information element identifier							
Length of information request contents								2
1 ext.	Info. req. ind	Type of information						3

FIGURE 8-7/Q.932

Information request information element

TABLE 8-13/Q.932

Information request information element

Information request indicator (octet 3, bit 7)

Bit
7
–
0 Information request completed
1 Prompt for additional information

Type of information (octet 3, bits 1-6)

Bits 6 5 4 3 2 1	
0 0 0 0 0 0	undefined
0 0 0 0 0 1	authorization code
0 0 0 0 1 0	address digits
0 0 0 0 1 1	terminal identification

All other values are reserved.

8.2.8 Notification indicator

The following definition of the Notification indicator information element complements that given in Recommendation Q.931. (See also Table 8-15.)

The purpose of the Notification indicator information element is to indicate information pertaining to a call, for example, a supplementary service operating at some other user within that call. The Notification indicator information element is coded as shown in Figure 8-8 and Table 8-14. The maximum length of the information element is application dependent consistent with the maximum length of the message. The Notification indicator information element may be repeated in a message.

8	7	6	5	4	3	2	1	
	Notification indicator							
0	0	1	0	0	1	1	1	Octet 1
	Information element identifier							
Length of notification indicator contents								2
0/1 ext.	Notification description							3
1 ext.	Notification description							3a
ASN.1 Encoded Data Structure								4, etc.

FIGURE 8-8/Q.932

Notification indicator information element

TABLE 8-14/Q.932

Notification indicator

Bit eight in octet 3 is used to extend the notification description field. If bit eight is 0, then another octet follows; if bit eight is 1, then octet 3 is the last octet. The value for a one octet field ranges from 0 to 127. For a multi-octet field, the order of bit values progressively decreases as the octet number increases.

Notification description (octet 3)

Extension (bit 8 octet 3) is set to 1

Bits

7 6 5 4 3 2 1	
0 0 0 0 0 0 0	User suspended
0 0 0 0 0 0 1	User resume
0 0 0 0 0 1 0	Bearer service change
0 0 0 0 1 0 0	Call completion delay
0 0 0 0 0 1 1	Discriminator for extension to ASN.1 encoded component (Note)
1 0 0 0 0 1 0	Conference established
1 0 0 0 0 1 1	Conference disconnected
1 0 0 0 1 0 0	Other party added
1 0 0 0 1 0 1	Isolated
1 0 0 0 1 1 0	Reattached
1 0 0 0 1 1 1	Other party isolated
1 0 0 1 0 0 0	Other party reattached
1 0 0 1 0 0 1	Other party split
1 0 0 1 0 1 0	Other party disconnected
1 0 0 1 0 1 1	Conference floating
1 0 0 1 1 1 1	Conference floating, served user preempted
1 0 0 1 1 0 0	Conference disconnected, preemption
1 1 1 1 0 0 1	Remote hold
1 1 1 1 0 1 0	Remote retrieval
1 1 0 0 0 0 0	Call is a waiting call
1 1 1 1 0 1 1	Call is diverting
1 1 0 1 0 0 0	Diversion activated
1 1 0 1 0 0 1	Reserved for call transfer
1 1 0 1 0 1 0	Reserved for call transfer
1 1 0 1 1 1 0	Reverse charging

All other values are reserved, for other Q.95x-Series services.

NOTE – Octet 4 etc. shall only be included when this value is used.

TABLE 8-15/Q.932

Formal definition of notification indicator information element

```
Notification-Indicator-IE-Data-Structure
        { ccitt recommendation q 932 notification-data-structure (6)}

DEFINITION    ::=
BEGIN

EXPORTS       NOTIFICATION

NOTIFICATION MACRO ::=
BEGIN

TYPE NOTATION                   ::= Argument
VALUE NOTATION          ::= value (VALUE CHOICE
                                    { localValue INTEGER,
                                    globalValue OBJECT IDENTIFIER })

Argument            ::= "ARGUMENT" NamedType
NamedType           ::= identifier type / type
END       --    of NOTIFICATION MACRO

NotificationDataStructure ::=       SEQUENCE
                                                { notificationTypeID NOTIFICATION,
                                                notificationArgument ANY DEFINED BY
                                                notificationTypeID }

-- ANY is filled by the single ASN.1 type following
-- the keyword ARGUMENT in the type definition of a particular
-- notification
```

8.2.9 Service profile identification

The purpose of the Service profile identification information element is to allow the user to initiate automatic assignment of the user service identifier and terminal identifier (see Annex A).

The Service profile identification information element is defined in Figure 8-9 and Table 8-16.

The default maximum length of the Service profile identification information element is 32 octets.

	Bits							
8	7	6	5	4	3	2	1	Octets
	Service profile identification							
0	0	1	1	1	0	1	0	1
	Information element identifier							
Length of service profile identification contents								2
0	SPID (IA5 characters)							3, etc.

FIGURE 8-9/Q.932

Service profile identification information element

TABLE 8-16/Q.932

Service profile identification information element

SPID (octet 3, etc.) The service profile identifier parameter is coed in IA5 characters, according to the format specified by the network.

9 Generic Notification Procedures

9.1 General

9.1.1 Introduction

This subclause specifies the functional signalling procedures that support the delivery of notifications at the user-network interface. Notifications can be characterized by the following properties:

- they do not cause a change of state on either side of the user-network interface;
- they represent a one-way flow of information that requires no response; and
- they provide additional information that can be discarded without the need for significant error recovery they are unrecognized by a user.

As a consequence of these properties it is possible to provide a generic set of procedures optimized to support the delivery of notifications at the user-network interface.

This subclause builds in a compatible manner on the basic call control procedures and in particular on the following:

- Subclause 5.9/Q.931, User notification procedures.

9.1.2 Scope of the Procedures

The procedures in Clause 9 define the basic methodology for the delivery of notifications at the user-network interface. The procedures are independent of whether or not the user-network is a point to point or point to multi-point configuration. The application of the full range of these procedures in the direction user to network is for further study.

9.1.3 Categories of Procedures

The generic procedures for the delivery of notification can be primarily categorized by the context of the delivery and secondarily categorized by the type of information contained in the notification. The procedures specified in Clause 9 consider the delivery of notifications in the following two contexts:

- Subclause 9.2 defines the procedures for the delivery of call related notifications on an active call reference; and
- Subclause 9.3 defines the procedures for the delivery of call independent notifications when no active call reference appropriate to the notification exists.

Procedures for both contexts are defined for the delivery of three types of notification information as follows:

1) The delivery of simple notification “indicators” based on the Notification indicator information element, as well as additional codepoints in the Notification indicator information element defined for Q.95x – Series supplementary services;
2) The delivery of notification “parameters” that are specified as information elements using the Q.931 encoding scheme defined in 4.5/Q.931 (Note including Q.931 encoded information elements defined for Q.95x – Series supplementary services;
3) When no response is required (e.g. REJECT), the delivery of notification “components” using an extension codepoint in octet 3 of the Notification indicator information element and ASN.1 encoded information in subsequent octets.

When no “parameters” are present option 1 (delivery of notification indicator will be used. When parameters are present, individual supplementary services will determine which options are applicable.

NOTE – With regard to the delivery of notification “parameters”, the use of the Notification indicator information element and Q.931 information elements together in a message other than NOTIFY requires further study.

9.2 Call Related Notifications

9.2.1 Introduction

The generic procedures for call related notifications are a compatible extension of the user notification procedures specified in 5.9/Q.931. the procedures specified in 9.2.2 enable the network to notify a user of supplementary service related events on an appropriate active call reference. A call reference is considered active in this context from the initiation of call establishment (including the SETUP message) to the completion of call clearing (including the RELEASE COMPLETE message). These procedures encompass the delivery of notifications using the active call reference of the call the notification is associated with. The application of these procedures in the direction user to network, in addition to those already defined in Recommendation Q.931, remains for further study.

9.2.2 Procedures

9.2.2.1 Delivery of Call Related Notifications

The delivery of call related notifications make use of an active call reference and its underlying data link layer connection.

If the delivery of the notification coincides with call establishment or clearing procedures, the notification information can be carried in the associated call control messages. Otherwise, the notification information is delivered in a NOTIFY message. The three types of notification information defined in 9.1.3 are supported in these messages.

9.2.2.2 Error handling

If a terminal does not recognize an information element in a NOTIFY message, or a new codepoint or extension contents of the Notification indicator information element, it shall handle it according to the procedures in 5.8/Q.931.

9.3 Call Independent Notifications

9.3.1 Introduction

The generic procedures for call independent notifications are a compatible addition to the use notification procedures specified in 5.9/Q.931. The procedures specified in 9.3.2 enable the network to notify a user of supplementary service related events when no appropriate call reference is active.

The application of these procedures in the direction user to network remains for further study.

9.3.2 Procedures

9.3.2.1 Underlying Data Link Layer Services

The delivery of call independent notifications requires the underlying services of the data link layer.

The procedures described in 9.3.2.2 and 9.3.2.3 make use of the acknowledged data link layer service supported on point to point data link layer connections. The network can use the data link broadcast capability to transfer notifications.

Terminals requiring the delivery of call independent notifications should retain an active data link layer connection, unless sufficient subscription information is available in the network to cause a data link layer connection to the required terminal to be established. If no subscription information is available, mechanisms for the data link layer connection to be established are for further study.

9.3.2.2 Delivery of Call Independent Notifications

Call independent notifications are delivered using the NOTIFY message on the dummy call reference. The three types of notification information defined in 9.1.3 are supported in this message.

The dummy call reference is specified in 4.3/Q.931. The NOTIFY message is specified in 3.1.7/Q.931.

9.3.2.3 Error handling

If a terminal does not recognize an additional new information element in the NOTIFY message, or a new codepoint or extension contents of the Notification indicator information element, it shall handle it according to the procedures in 5.8/Q.931.

9.4 Extension of the Notification indicator information element

See 8.2.9.

Annex A

User service profiles and terminal identification

(This annex forms an integral part of this Recommendation)

A.1 Introduction

These optional procedures allow an ISDN to support identification and selection of specific terminals on a multi-point user-network interface to support multiple user service profiles in those cases in which Recommendation Q.931 information elements are not sufficient for such purposes.

A terminal or network which desires to support such multiple profiles for terminals which could not otherwise be distinguished, must support this additional identification procedure. Otherwise, it is completely optional.

TABLE A.1/Q.932

Terminology

Service profile	Service profile refers to the information that the network maintains for a given user to characterize the service offered by the network to that user. As an example, this may contain the association of feature identifiers to specific supplementary services. A service profile may be allocated to an access interface or to a particular user equipment or a group of user equipments.
SPID	The service profile identifier is a parameter carried in a service profile identification information element that is sent from the user to network to allow network assignment of a USID and TID. A user's SPID should uniquely identifiy a specific profile of service characteristics stored within the network. The SPID will allow the network to distinguish between different terminals that would otherwise be indistinguishable (e.g. same ISDN number). The SPID value is provided to the user at subscription time.
USID	User service identifier. A USID uniquely identifies a service profile on an access interface.
TID	Terminal identifier. A TID value is unique within a given USID. If two terminals on an interface subscribe to the same service profile, then the two terminals will be assigned the same service USID. However, two different TIDs are required to uniquely identify each of the two terminals.
EID	Endpoint identifier. The endpoint identifier information element is used for terminal identification. The endpoint identifier parameters contain a USID and TID and additional information used to interpret them.

Figure A.1 shows examples of the relationships of terminals, SPIDs, USIDs, and TIDs and their dynamic relationship to TEIs. In this example, terminals 1, 3, 4 and 5 support the automatic endpoint identifier parameter assignment procedure and terminal 2 does not, but has the endpoint identifier parameters locally entered. Terminal 6 does not support terminal identification, therefore it utilizes the specified default service profile.

NOTE – Items in parentheses indicate values or relationships which are dynamically established by initialization procedures (see A.4). Others are established via administrative actions and stored as a result of manual entry.

A user or network that does not recognize the information elements used by this annex shall, if these elements are received, apply the error procedures defined in 5.8/Q.931.

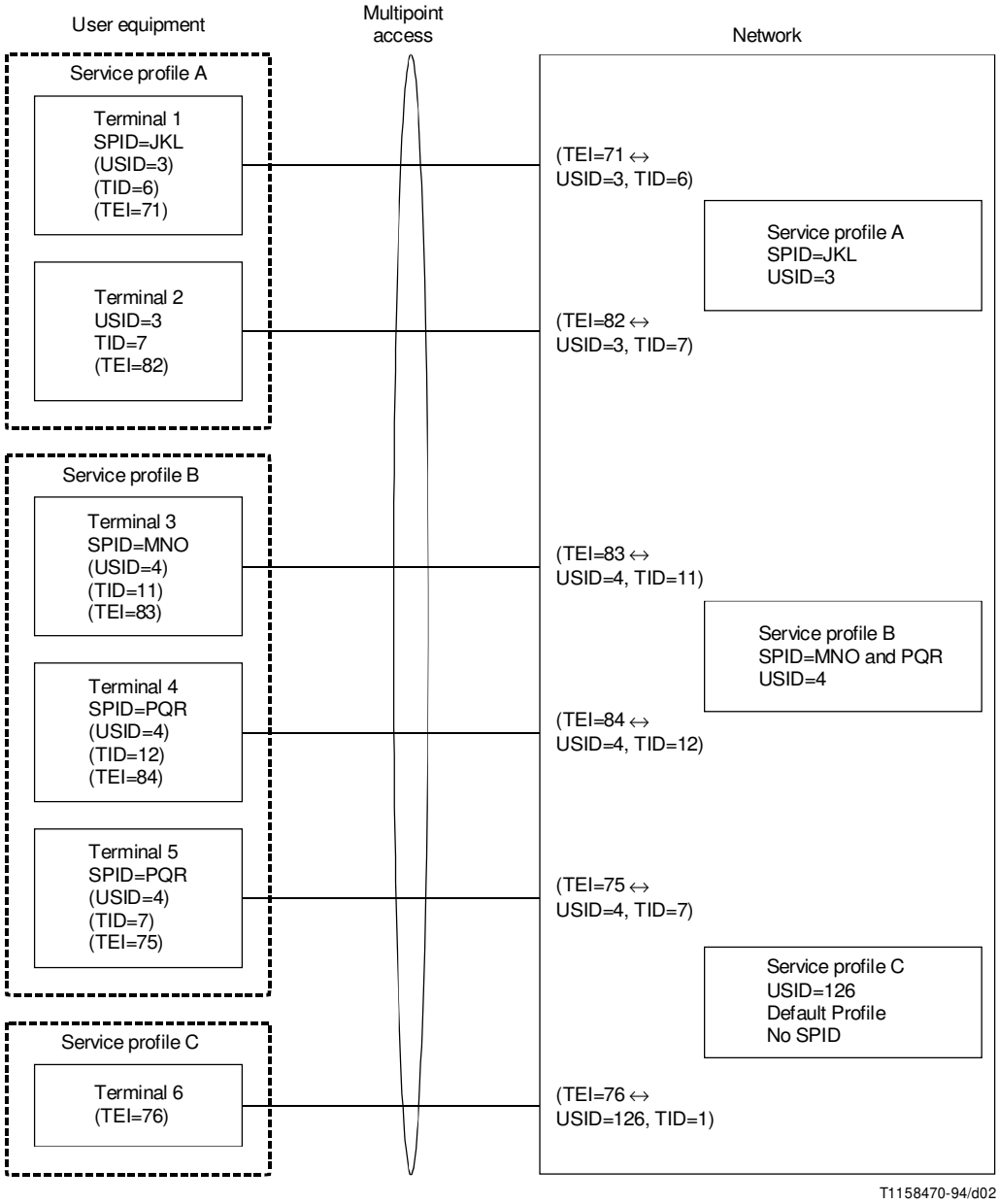


FIGURE A.1/Q.932

Relationship of service profile, SPID, USID, TID and TEI

A.2 User service profiles

The support of user service profiles requires that the service requests from a terminal are associated by the network with a specific profile. A USID is used to identify the profile on an access. The service profile is assigned to a data link connection so that the network can associate all of the service requests from the corresponding Connection Endpoint Suffix (CES) with the required profile (see Note). The assignment of a service profile to a data link connection minimizes the per-service request overhead of profile identification.

The procedures for assigning service profile to a data link connection are incorporated into the initialization procedures described in A.4.

NOTE – CES along with SAPI constitute the CEI (Connection Endpoint Identifier) that is used to identify message units passed between the data link layer (as represented by the TEI) and Layer 3.

A.3 Terminal identification

The support of terminal identification requires that a call sent by the network can be addressed to:

- all of the terminals of a user service profile;
- one terminal of a user service profile; or
- all but one terminal of a user service profile.

A USID is used to identify the user service profile with a (set of) terminals on an access interface and a TID is used to identify individual terminals within a user service profile on an access.

The USID and TID may be entered into the terminal by the user as arranged at subscription time, or dynamically downloaded to the terminal from the network with an automatic assignment procedure.

The USID and TID parameters are used by the terminal to check the compatibility of a call offered by the network. The inclusion of a USID and TID with only access uniqueness minimizes the per-call overhead of supporting terminal addressing.

The procedures for downloading the USID and TID to a terminal are incorporated into the automatic endpoint identifier allocation and initialization procedures described in A.4. The procedures for using a USID and TID for terminal identification in an offered call sent by the network are described in A.5.

A.4 Initialization

The initialization procedure provides for the association by the network of the service requests from a terminal on a particular data link connection (as represented by the TEI) with a user service profile. A user requested automatic assignment procedure is described to also support automatic assignment of USID and TID parameters and their downloading by the network to a terminal.

Since initialization provides the basis for subsequent association of a service profile with a data link connection, normally, user equipment that supports initialization is expected to request the initialization procedure (e.g. on the first Layer 3 message after dynamic assignment of a TEI). However, a request for initialization is allowed at any time. The data link connection is always associated with the most recently identified service profile. Under some circumstances, the network may solicit terminal initialization.

A.4.1 Terminal requested initialization

a) Terminals may initialize by sending an Endpoint identifier information element (containing a USID and TID) in an INFORMATION message at any time to the network. Subsequent to this, the network may associate the service profile with the data link over which the message was sent.

b) For terminals which support automatic assignment of USID and TID parameters, initialization (that is, association of a service profile with a data link connection) is provided as part of the automatic assignment procedure described here.

A user may initiate automatic assignment of the endpoint identifier by sending a Service profile identification information element in an INFORMATION message with the dummy call reference. The Service profile identification information element should contain the SPID parameter allocated at the time of subscription. The initialization is acknowledged with an INFORMATION message with the Endpoint identifier information element containing a USID and TID, the values of which are determined by the network. It results in an association of the data link over which it was received with the identified service profile.

When a terminal determines that the initialization procedure has failed, it assumes that the network cannot support the procedure and does not repeatedly attempt initialization.

A.4.2 Network solicited initialization

The network may solicit a request for initialization on a data link connection by sending an Information request information element with codepoint "terminal identification" in an INFORMATION message with the dummy call reference. Upon receiving the request, the terminal may respond as described in the previous A.4.1 a) or b).

When a network determines that the initialization procedure has failed, it assumes that the terminal cannot support the procedures and does not repeatedly request initialization.

A.4.3 Collision

When terminal initialization and network solicitation procedures collide, the terminal ignores the solicitation from the network and the network proceeds as normal upon receipt of the initialization request from the terminal.

A.5 Identification procedures

When the network offers a call using terminal addressing, the Endpoint identifier information element is included in the SETUP message.

When a terminal receives a SETUP message containing the Endpoint identifier information element, it shall:

- if it is not supported, handle the Endpoint identifier information element in accordance with 5.8.7/Q.931 and complete normal compatibility checking procedures; or,
- test for an address compatibility with the Endpoint identifier information element if it is supported in addition to completing the normal compatibility checking procedures.

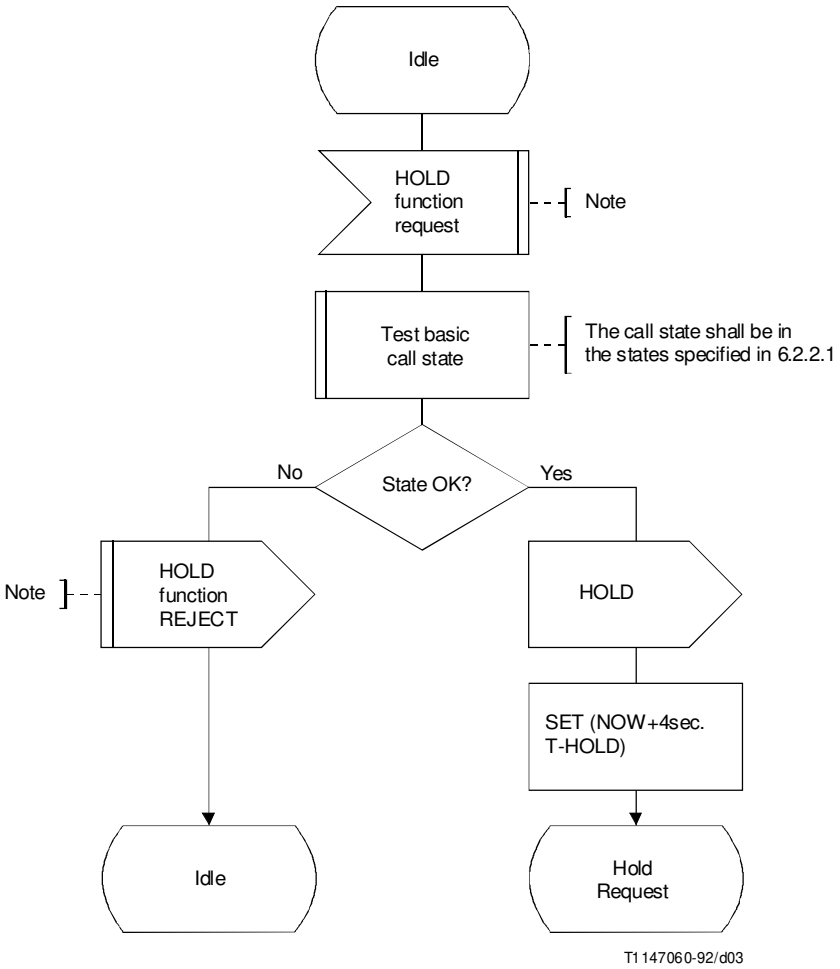
Annex B

SDL diagrams for Hold/Retrieve functions

(This annex forms an integral part of this Recommendation)

B.1 Introduction

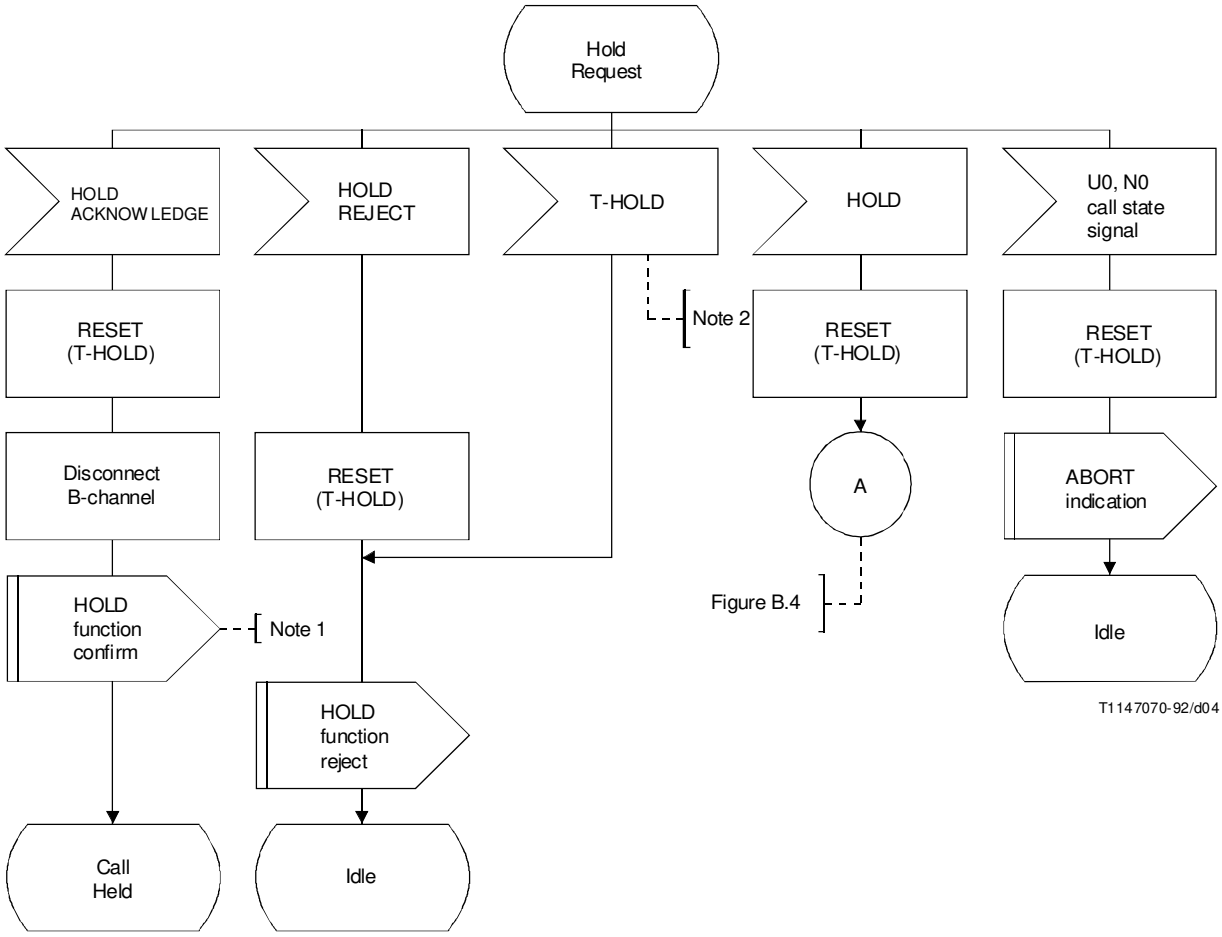
This annex provides SDL diagrams see Figures B.1 to B.5 for the description of the Hold and Retrieve functions according to the procedures defined in 6.2. Interaction of the Hold/Retrieve functions with the channel reservation function given in 6.4 is not covered in these SDLs.



NOTE – This signal is sent to/received from the appropriate individual service entiry using the HOLD/RETRIEVE functions.

FIGURE B.1/Q.932 (sheet 1 of 2)

Hold function initiating entity



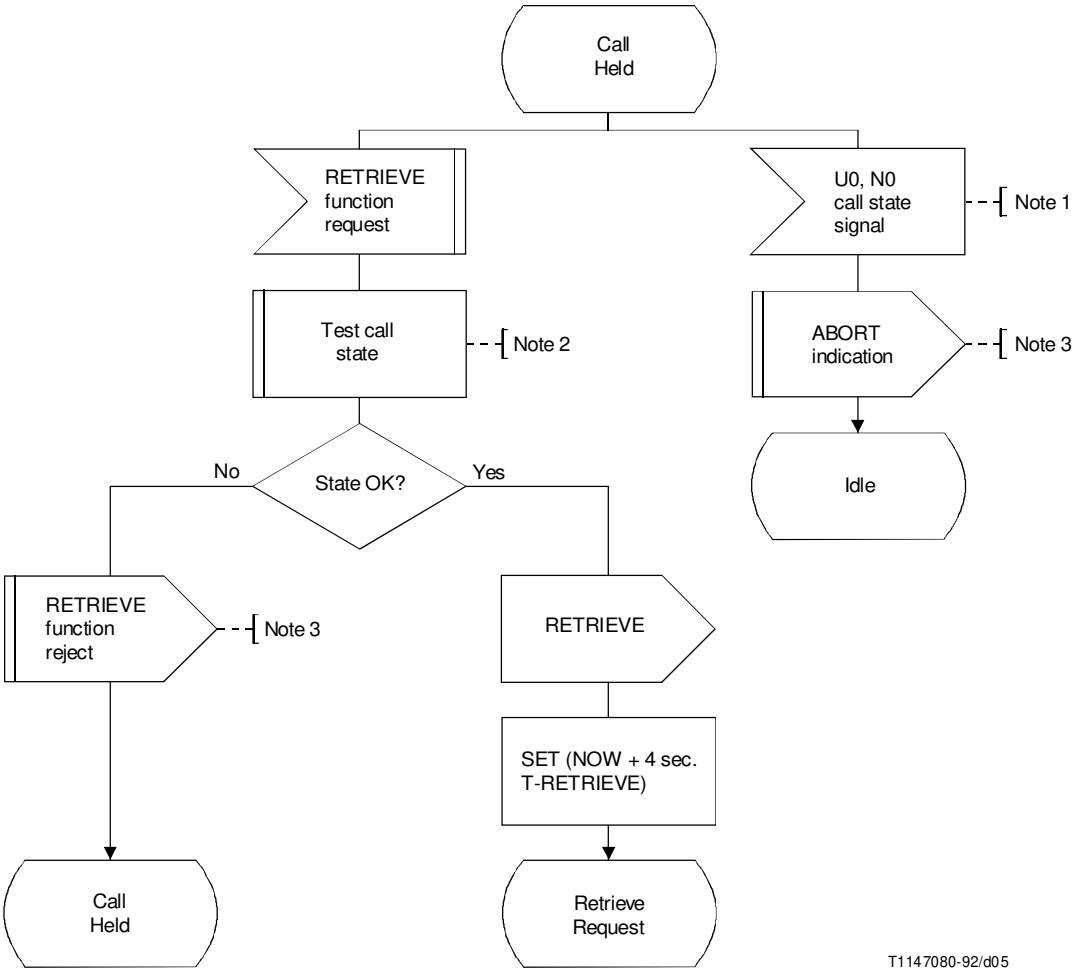
NOTES

1 This signal is sent/received to/from the appropriate supplementary service entity using the HOLD/RETRIEVE functions.

2 This signal is received as a result of T-hid expiry.

FIGURE B.1/Q.932 (sheet 2 of 2)

Hold function initiating entity



NOTES

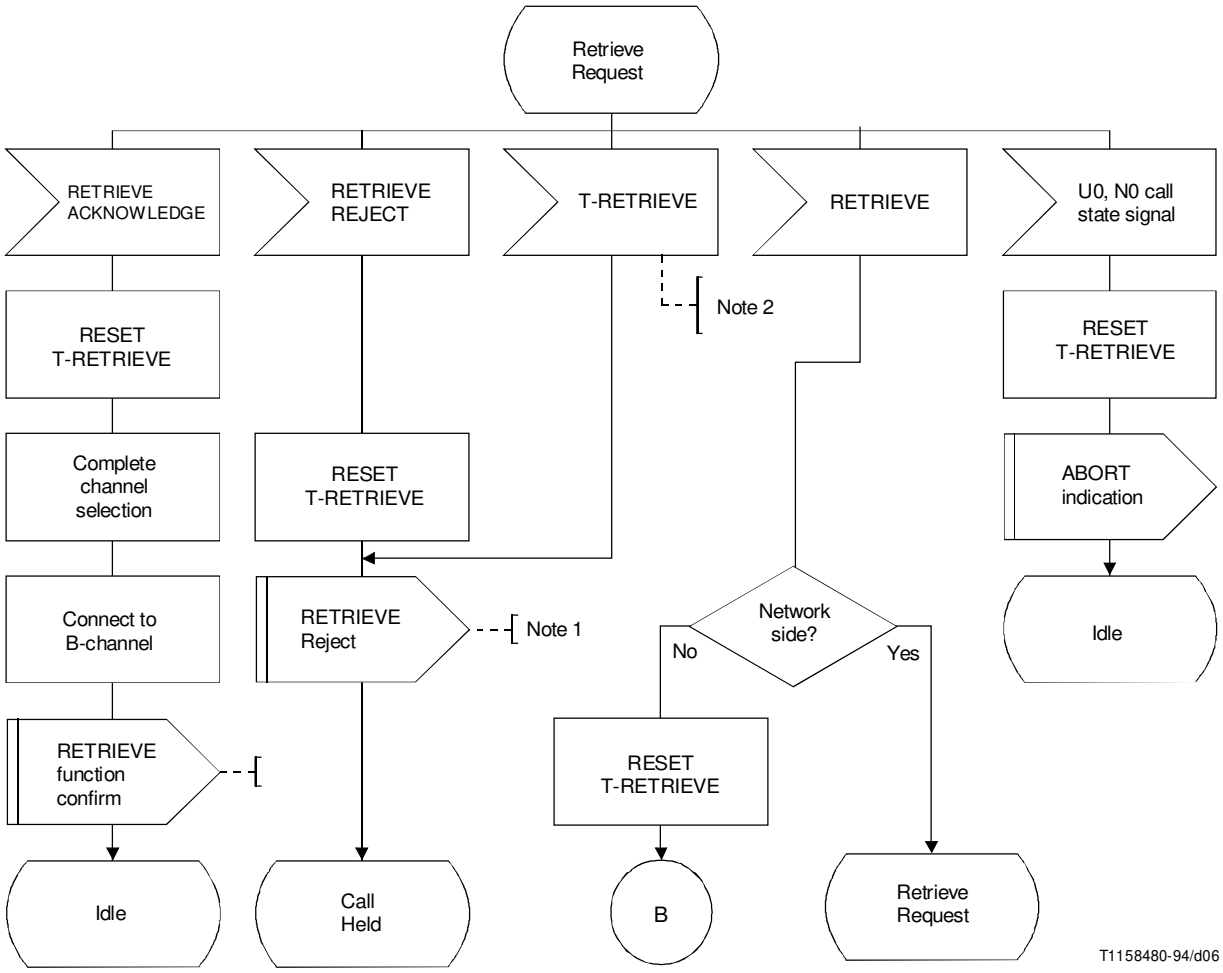
1 Any basic call signal that causes a transition to state 0U/N0 Idle shall generate this signal.

2 The call shall be in states specified in 6.2.3.1.

3 This signal is sent/received to/from the appropriate supplementary service entity using the HOLD/RETRIEVE functions.

FIGURE B.2/Q.932 (sheet 1 of 2)

Retrieve function initiating entity



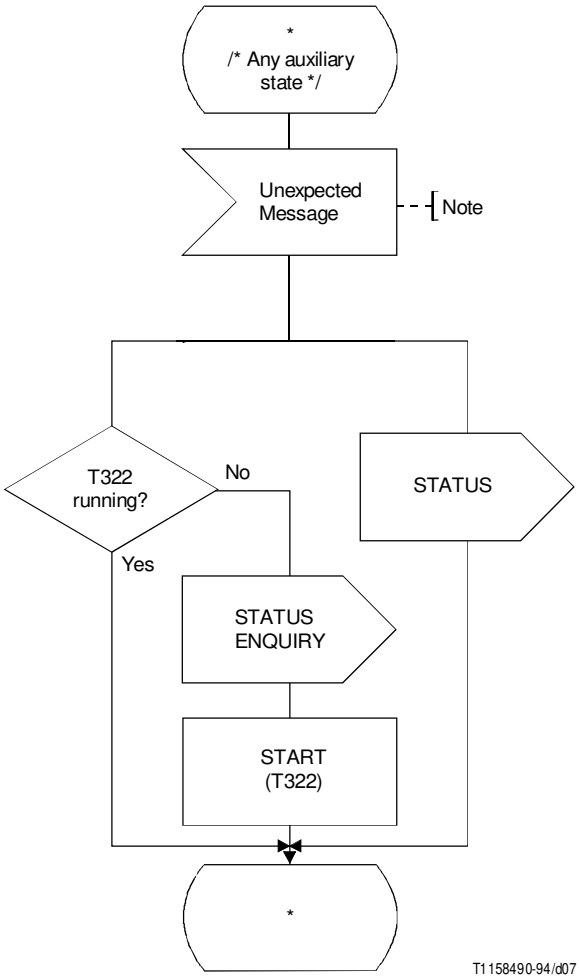
NOTES

1 This signal is sent to the appropriate individual service entity using the HOLD/RETRIEVE functions.

2 This signal is received as a result of T-ret expiry.

FIGURE B.2/Q.932 (sheet 2 of 2)

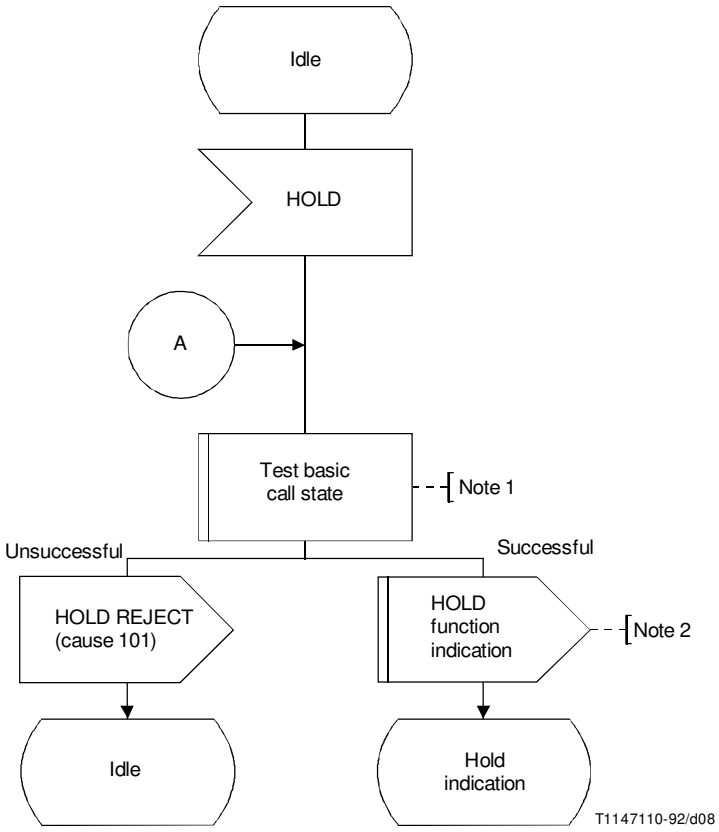
Retrieve function initiating entity



NOTE – Unexpected message means any message not indicated in this particular state and belonging to the HOLD/RETRIEVE family of messages. Further treatments are same as for basic call, e.g. T322 expiry.

FIGURE B.3/Q.932

Hold and Retrieve function for unexpected message treatment



NOTES

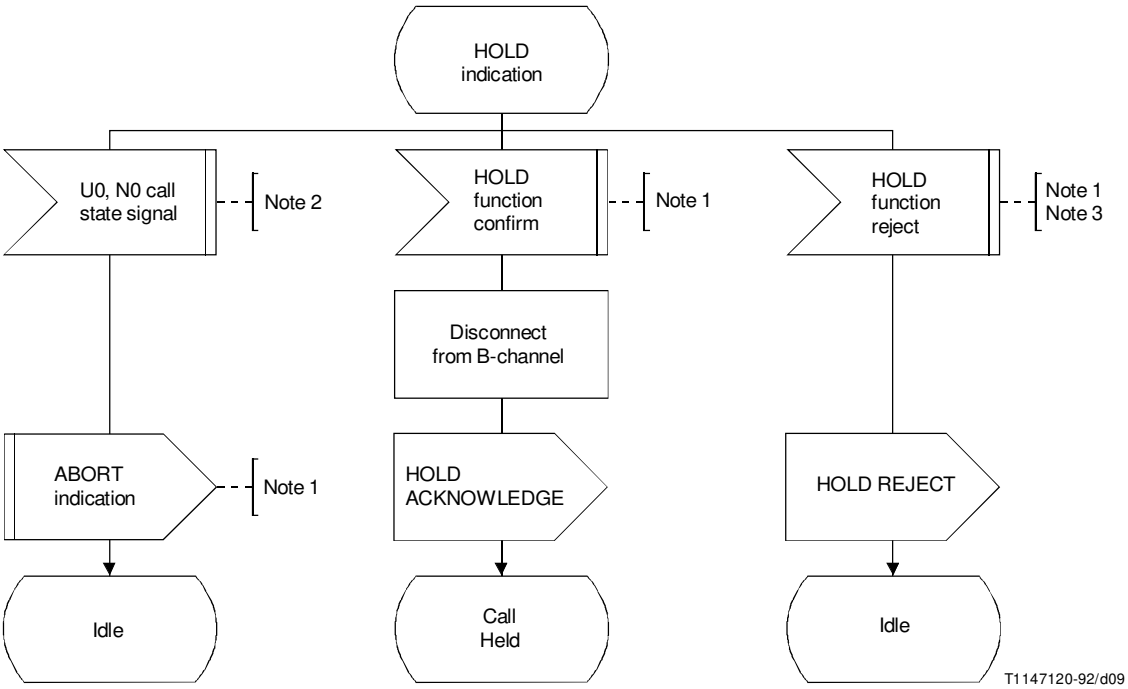
1 The call shall be in the states as specified in 6.2.2.1.

2 This signal is sent to the appropriate individual service entity using the HOLD/RETRIEVE functions.

3 In case of collision of two HOLD messages, this primitive will be coupled with the HOLD function confirm primitive of Figure B.4 (sheet 2 of 2).

FIGURE B.4/Q.932 (sheet 1 of 2)

Hold function responding entity



NOTES

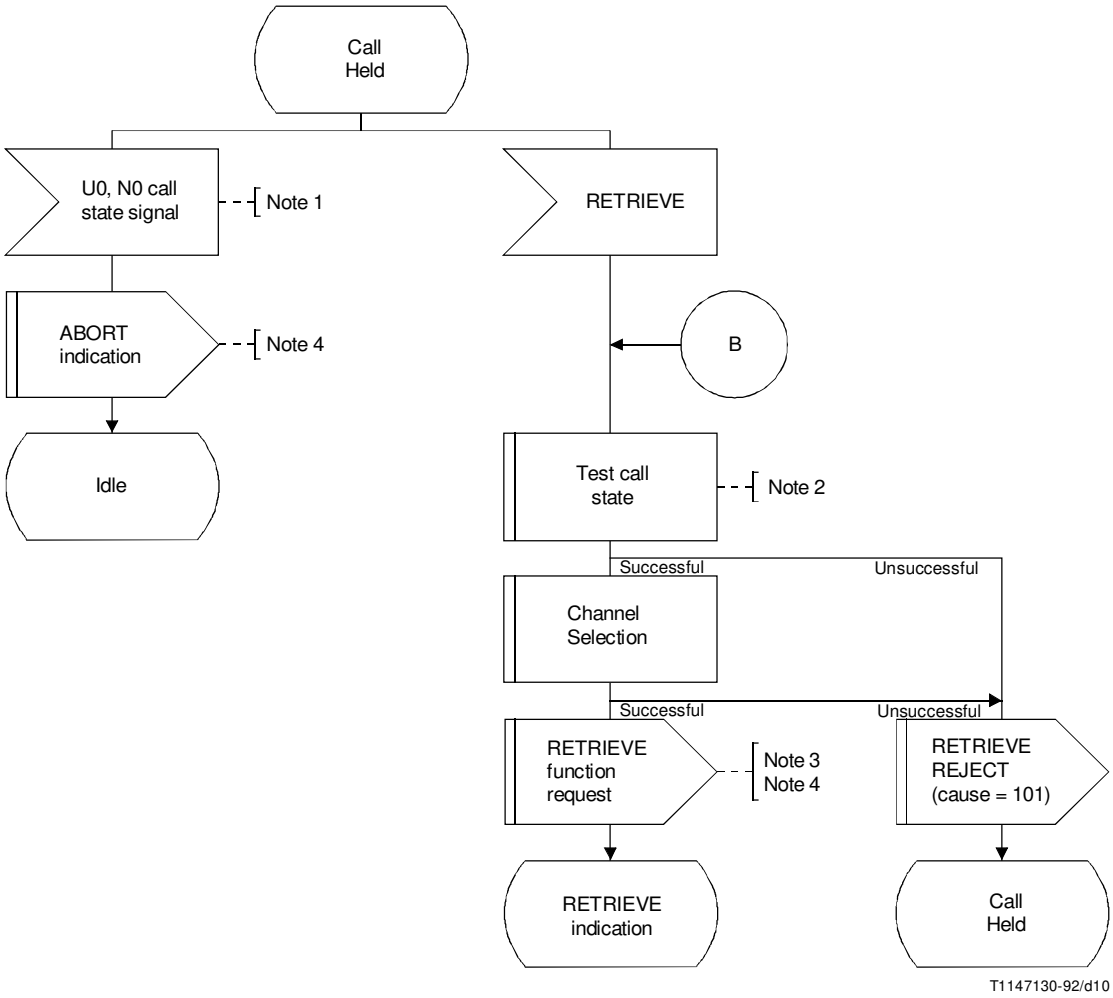
1 This signal is sent to/received from the appropriate individual service entity using the HOLD/RETRIEVE functions.

2 Any basic call signal which causes a transition to U0, N0 shall generate this signal.

3 The cause value is supplementary service specific.

FIGURE B.4/Q.932 (sheet 2 of 2)

Hold function responding entity



NOTES

1 Any basic call signal which causes a transition to U0, N0 Idle shall generate this signal.

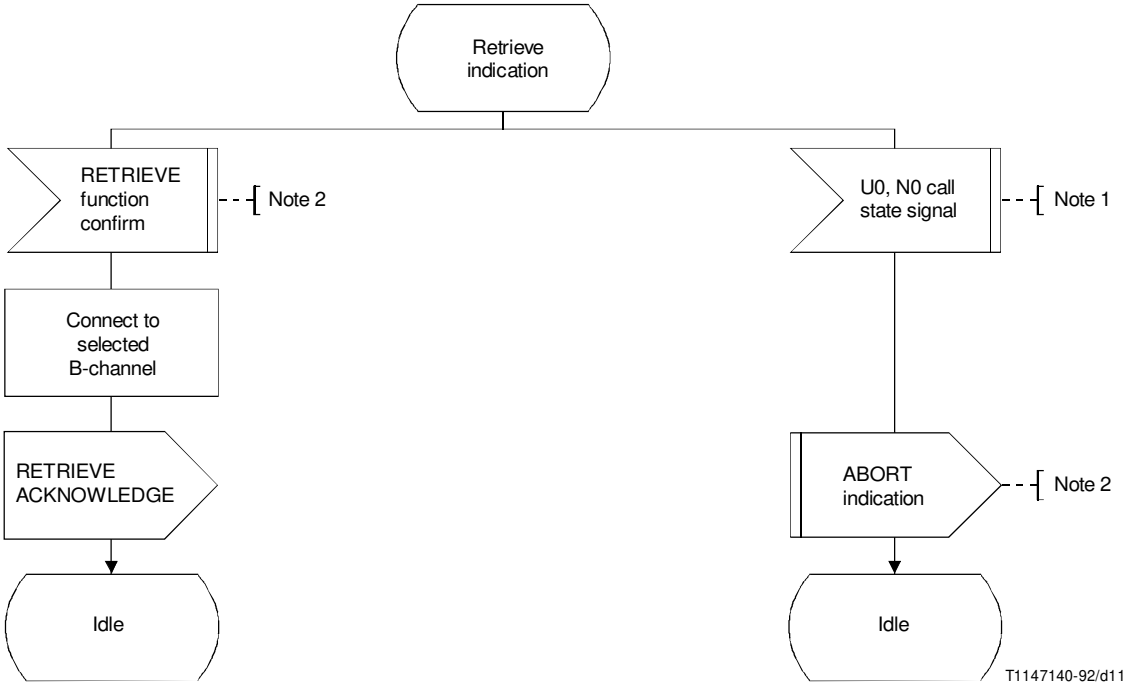
2 The call shall be in the states as specified in 6.2.3.1.

3 In case of a RETRIEVE message colliding with a RETRIEVE message, this primitive will be coupled with the RETRIEVE function confirm primitive of Figure B.5 (sheet 2 of 2).

4 This signal is sent to the appropriate individual service entity using the HOLD/RETRIEVE functions.

FIGURE B.5/Q.932 (sheet 1 of 2)

Retrieve function responding entity



NOTES

1 Any basic call signal which causes a transition to U0, N0 Idle shall generate this signal.

2 This signal is sent to/received from the appropriate individual service entity using the HOLD/RETRIEVE functions.

FIGURE B.5/Q.932 (sheet 2 of 2)

Retrieve function responding entity

Annex C

Definition of address types

(This annex forms an integral part of this Recommendation)

The following address definitions are imported to Q.95x-Series Recommendations.

```
Addressing-Data-Elements                { CCITT-recommendation q932
                                          addressing-data-elements (7) }

DEFINITIONS EXPLICIT TAGS       ::=

BEGIN

EXPORTS                                 PresentedAddressScreened,
                                        PresentedAddressUnscreened,
                                        PresentedNumberScreened,PresentedNumberUnscreened,
                                        Address, PartyNumber, PartySubaddress,
                                        ScreeningIndicator, PresentationAllowedIndicator;

PresentedAddressScreened        ::= CHOICE {
                                        presentationAllowedAddress [0] IMPLICIT AddressScreened,
                                        presentationRestricted [1] IMPLICIT  NULL,
                                        numberNotAvailableDueToInterworking [2] IMPLICIT NULL,
                                        presentationRestrictedAddress [3] IMPLICIT AddressScreened
                                        }

PresentedAddressUnscreened      ::= CHOICE {
                                        presentationAllowedAddress [0] IMPLICIT Address

                                        presentationRestricted [1] IMPLICIT  NULL,
                                        numberNotAvailableDueToInterworking [2] IMPLICIT NULL,
                                        presentationRestrictedAddress [3] IMPLICIT Address }

PresentedNumberScreened         ::= CHOICE {
                                        presentationAllowedNumber [0] IMPLICIT NumberScreened,
                                        presentationRestricted [1] IMPLICIT NULL,
                                        numberNotAvailableDueToInterworking [2] IMPLICIT NULL,
                                        presentationRestrictedNumber [3] IMPLICIT NumberScreened }

PresentedNumberUnscreened       ::= CHOICE {
                                        presentationAllowedNumber [0] PartyNumber,
                                        presentationRestricted [1] IMPLICIT NULL,
                                        numberNotAvailableDueToInterworking [2] IMPLICIT NULL,
                                        presentationRestrictedNumber [3] PartyNumber }

AddressScreened                 ::= SEQUENCE {
                                        PartyNumber,
                                        ScreeningIndicator,
                                        PartySubaddress OPTIONAL }

NumberScreened                  ::= SEQUENCE {
                                        PartyNumber,
                                        ScreeningIndicator }

Address                         ::= SEQUENCE {
                                        PartyNumber,
                                        PartySubaddress OPTIONAL }
```

```
PartyNumber                 ::= CHOICE {
                                    unkownPartyNumber [0] IMPLICIT NumberDigits,
                                    -- the numbering plan is the default numbering plan
                                    -- of the network. It is recommended that this value
                                    -- is used.
                                    publicPartyNumber [1] IMPLICIT PublicPartyNumber,
                                    -- the numbering plan is according to
                                    -- Recommendations E.163 and E.164.
                                    dataPartyNumber [3] IMPLICIT NumberDigits,
                                    -- not used, value reserved.
                                    telexPartyNumber [4] IMPLICIT NumberDigits,
                                    -- not used, value reserved.
                                    privateNumber [5] IMPLICIT PrivateNumber,
                                    nationalStandardPartyNumber [8] IMPLICIT NumberDigits }
                                    -- not used, value reserved.

PublicPartyNumber           ::= SEQUENCE {
                                    publicTypeOfNumber PublicTypeOfNumber,
                                    publicNumberDigits NumberDigits }

PrivatePartyNumber          ::= SEQUENCE {
                                    privateTypeOfNumber PrivateTypeOfNumber,
                                    privateNumberDigits NumberDigits }

NumberDigits                ::= NumericString (SIZE(1..20))

PublicTypeOfNumber          ::= ENUMERATED {
                                    unknown (0),
                                    -- if used number digits carry prefix indicating type
                                    -- of number according to national
                                    -- recommendations
                                    internationalNumber (1),
                                    nationalNumber (2),
                                    networkSpecificNumber (3),
                                    -- not used, value reserved
                                    subscriberNumber (4),
                                    abbreviatedNumber (6) }
                                    -- valid only for called party number at the
                                    -- outgoing access, network substitutes
                                    -- appropriate number

PrivateTypeOfNumber         ::= ENUMERATED {
                                    unknown (0),
                                    level2RegionalNumber (1),
                                    level1RegionalNumber (2),
                                    pTNSpecificNumber (3),
                                    localNumber (4),
                                    level3RegionalNumber (5),
                                    abbreviatedNumber (6) }

PartySubaddress             ::= CHOICE {
                                    UserSpecifiedSubaddress,
                                    -- not recommended
                                    NSAPSubaddress }
                                    -- according to Recommendation X.213

UserSpecifiedSubaddress     ::= SEQUENCE {
                                    SubaddressInformation,
                                    oddCountIndicator BOOLEAN OPTIONAL }
                                    -- used when the coding of subaddress is BCD

NSAPSubaddress              ::= OCTET STRING (SIZE(1..20))
                                    -- specified according to X.213. Some networks
                                    -- may limit the subaddress value to some other
                                    -- length, e.g. 4 octets
```

```
SubaddressInformation    ::= OCTET STRING (SIZE(1..20))
                                    -- coded according to user requirements. Some
                                    -- networks may limit the subaddress value to some
                                    -- other length, e.g. 4 octets

ScreeningIndicator       ::= ENUMERATED {
                                    userProvidedNotScreened (0),
                                    -- number was provided by a remote user terminal
                                    -- equipment, and has been screened by a
                                    -- network that is not the local public or local
                                    -- private network.
                                    userProvidedVerifiedAndPassed (1),
                                    -- number was provided by a remote user terminal
                                    -- equipment (or by a remote private network), and
                                    -- has been screened by the local public or local
                                    -- private network.
                                    userProvidedVerifiedAndFailed (2),
                                    -- not used, value reserved
                                    networkProvided (3) }
                                    -- number was provided by local
                                    -- public or local
                                    -- private network

PresentationAllowedIndicator     ::= BOOLEAN

END -- of addressing data elements
```

Appendix I

Illustration of the application of the three protocol types

(This appendix does not form an integral part of this Recommendation)

I.1 Introduction

This appendix is provided as an illustration of the application of the three protocol types defined in this Recommendation. The examples shown should not be taken as definitive examples, since the support of the Keypad and the Feature key management protocols are network dependent.

The signalling sequences shown are not exhaustive and are only intended to illustrate possible supplementary service control sequences.

I.2 Example use of the Keypad protocol

The example in Figure I.1 illustrates a user feature request using the Keypad protocol. The network associates the contents of the Keypad information element with the appropriate feature. The user is shown to subsequently enter supplementary service parameters using the Keypad protocol. Feature status information may be provided by the network in the Display information element. The network completes feature processing and the user is shown to clear call reference. Alternatively, depending on the specific feature request, a CALL PROCEEDING message might be returned by the network and normal call processing procedures would continue.

The specific example shown in Figure I.2 illustrates the support of a hold/retrieve function based on the use of INFORMATION messages for the conveyance of Keypad facility or Display information elements. An enquiry call is established through the conveyance of the called party address digits via a Keypad facility information element within INFORMATION messages. These address digits are sent after putting the existing call on hold through the transfer of a facility request via a Keypad facility information element within an INFORMATION message.

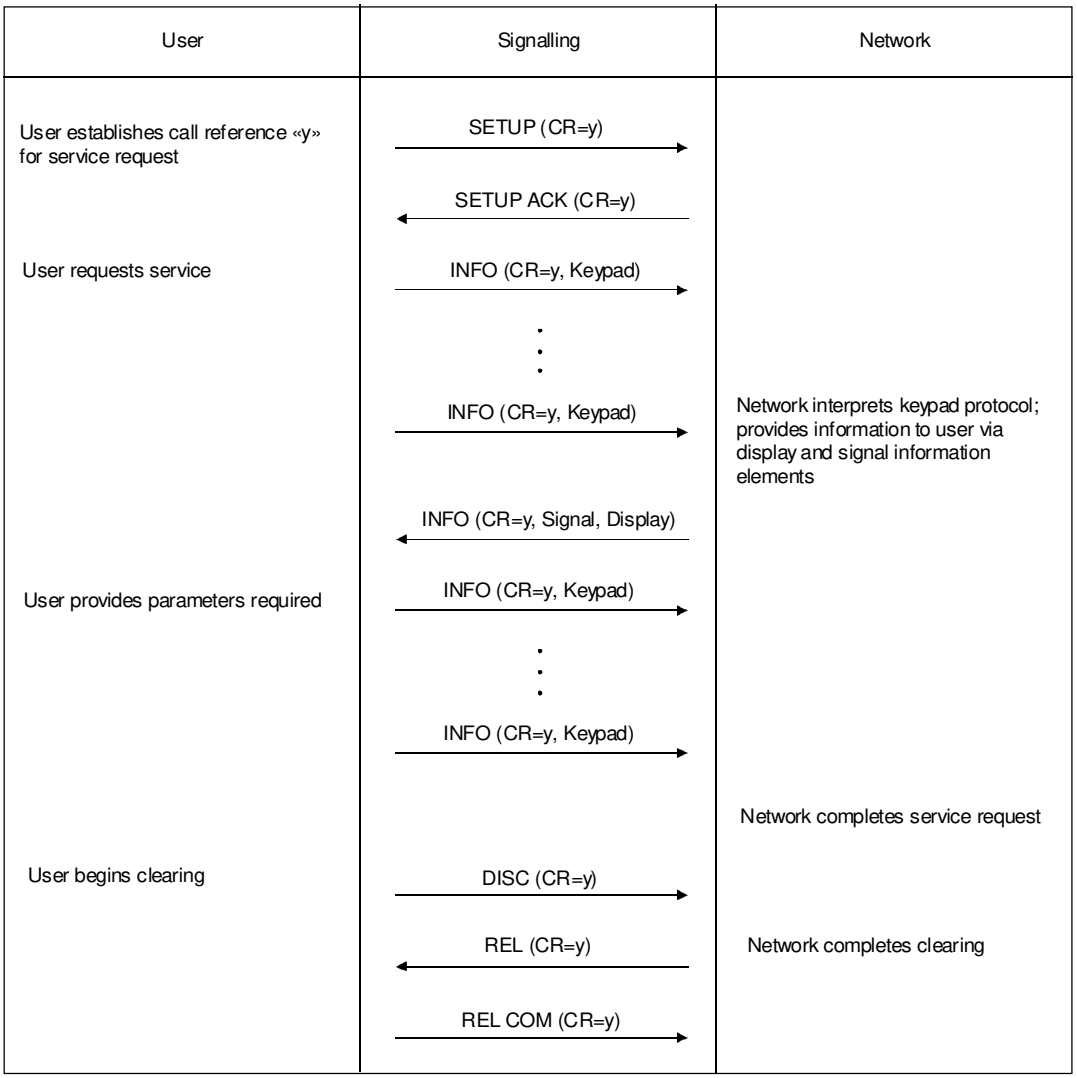
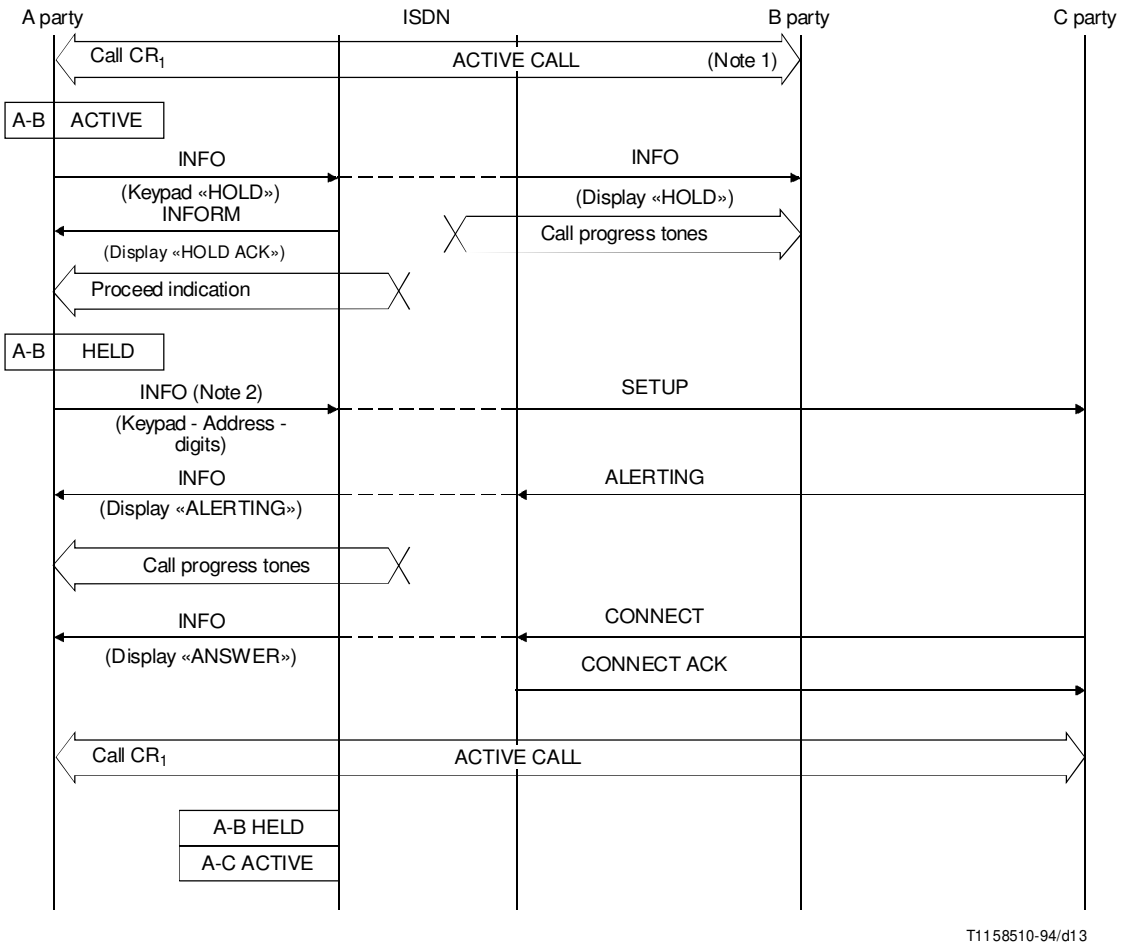


FIGURE I.1/Q.932

A generic example of the use of the keypad protocol



NOTES

1 The first call is established using the normal call establishment procedures specified in Recommendation Q.931.

2 The same call reference as that of the active call is used to establish the enquiry call. The characteristics of the second call are assumed to be identical to the first call (e.g. same bearer capability, low layer compatibility, transit network selection, information elements, etc.).

FIGURE I.2/Q.932

Specific example of establishing a second call while holding the first one using the keypad protocol

I.3 Example of use of the Feature key management protocol

This example illustrates the use of the Feature key management protocol for the invocation of a supplementary service by a user having initiated a call establishment by sending a SETUP message with incomplete (or no) address information, after having entered the overlap sending state upon receipt of the SETUP ACKNOWLEDGE message. Figure I.3 depicts the user providing supplementary service parameters. This is accomplished via the Keypad facility information element within INFORMATION messages after having invoked the request of a supplementary service by sending a Feature activation information element contained in an INFORMATION message to the network. The association of the feature identifier number (provided within the Feature activitation information element) with a given supplementary service has to be arranged between the user and the network at subscription time.

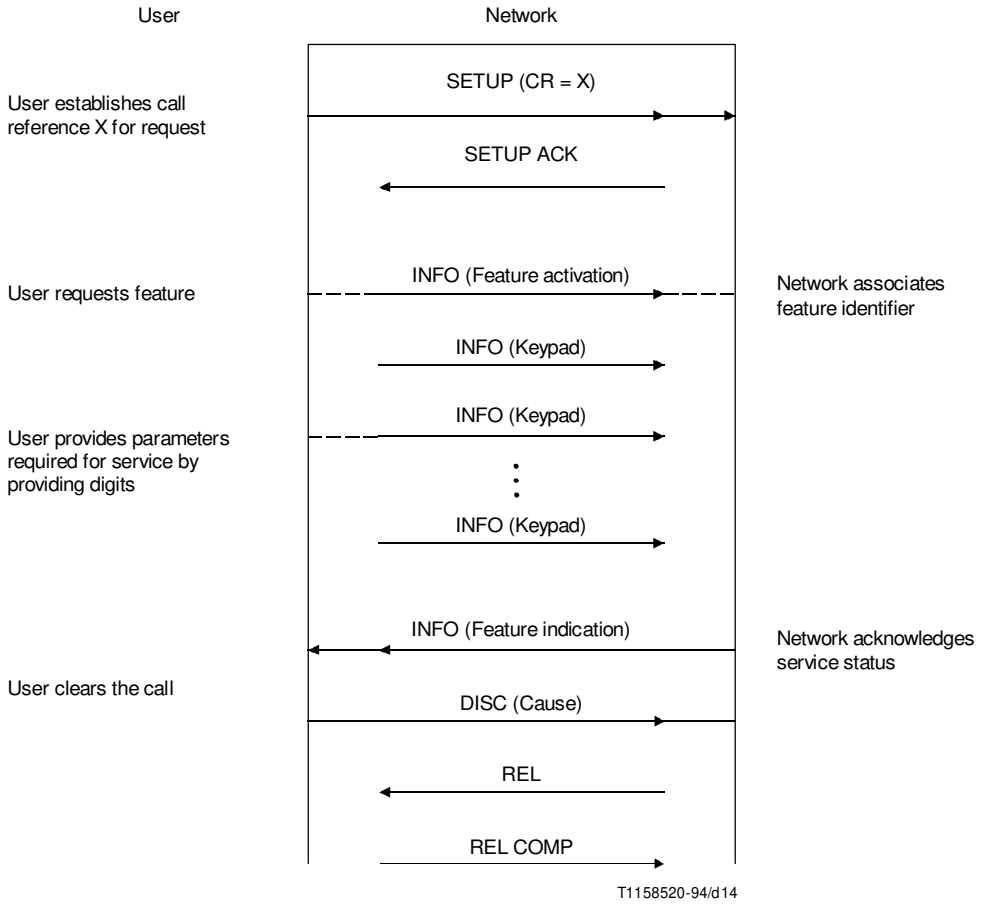
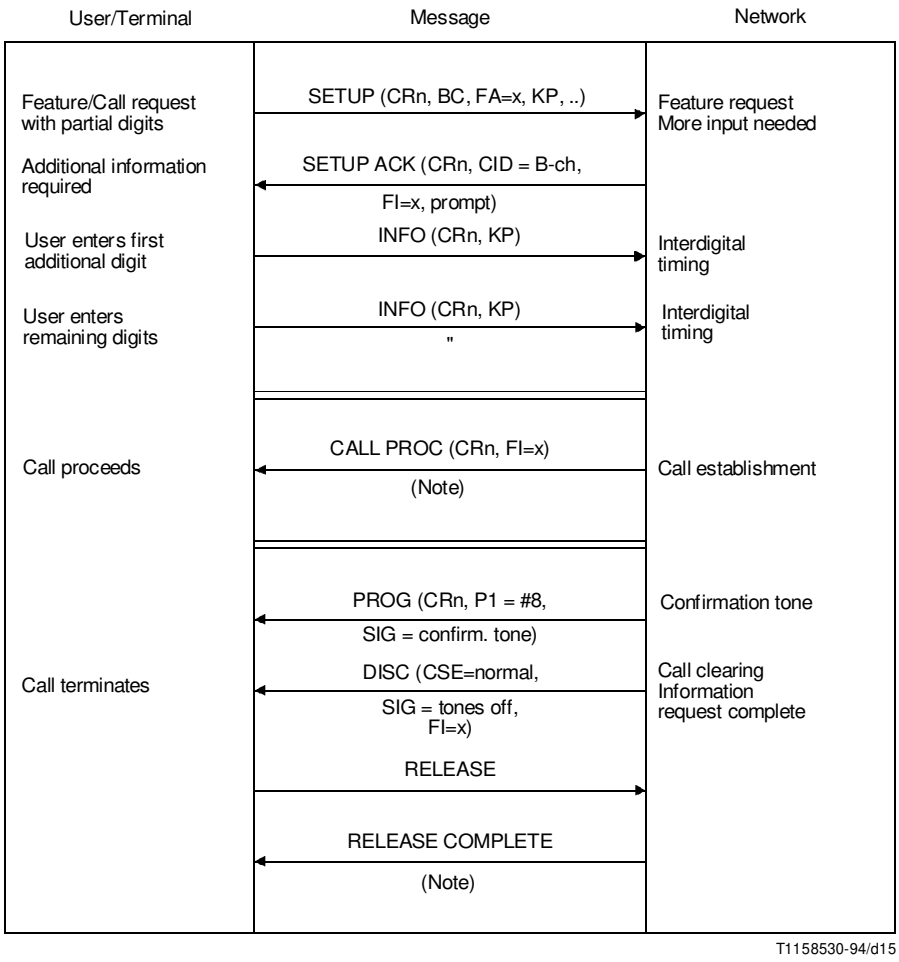


FIGURE I.3/Q.932

A generic example of the use of the Feature key management protocol

Scenario: The network receives a feature request (feature activator #x) with partial supplemental information included in the SETUP message. See Figure I.4.

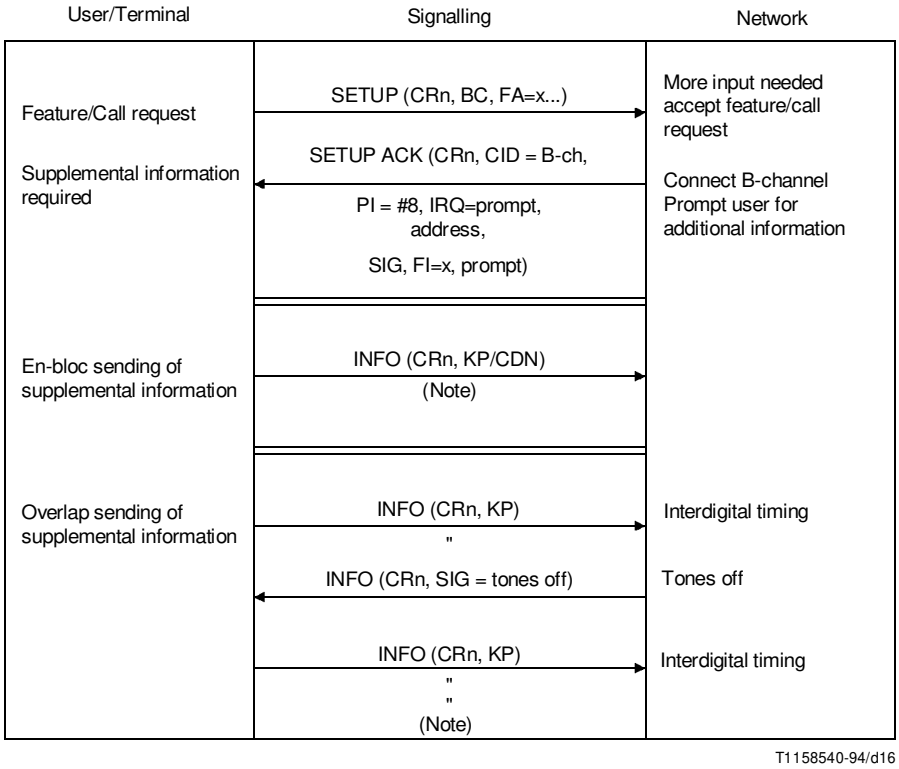


NOTE – Depending on the specific feature activated, the call will proceed or will be cleared.

FIGURE I.4/Q.932

Single overlap sequence

Scenario: The network receives a feature request in the SETUP message for which additional information is requested. See Figure I.5.

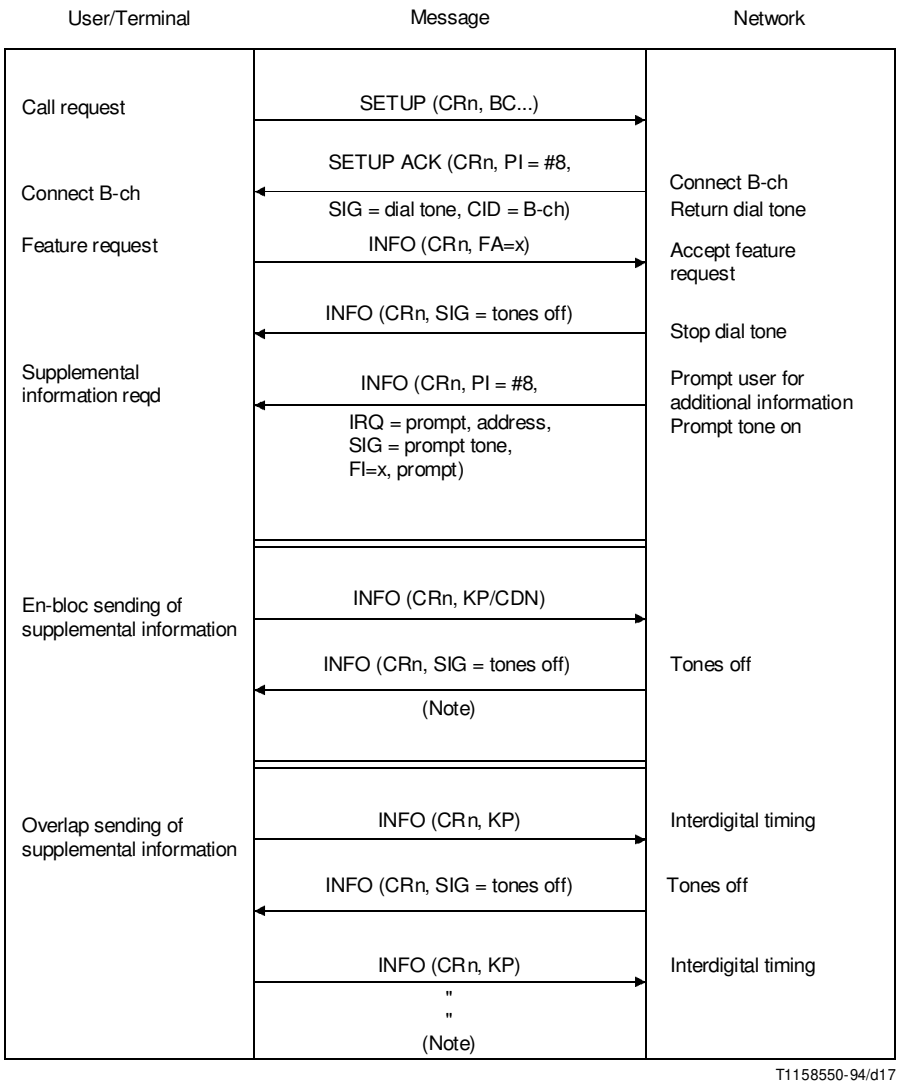


NOTE – Depending on the specific feature activated, the call will proceed or be cleared.

FIGURE I.5/Q.932

En-bloc followed by en-bloc/overlap sequence

Scenario: The network receives a feature request (feature activator #x) subsequent to returning a SETUP ACKnowledge message. The network prompts for additional information. See Figure I.6.



NOTE – Depending on the specific feature activated, the call will proceed or be cleared.
See Table 4-1 for examples of stages for sending of information.

FIGURE I.6/Q.932

Overlap followed by en-bloc/overlap sequence

Scenario: A user attempts to activate a feature, using feature activator #x, during the Active phase of a call. See Figure I.7.

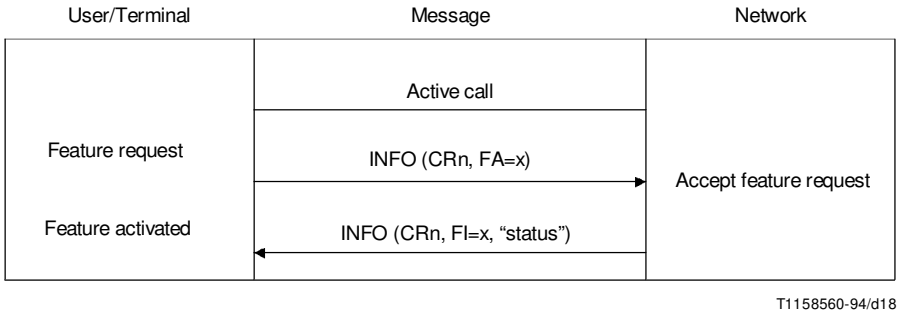


FIGURE I.7/Q.932

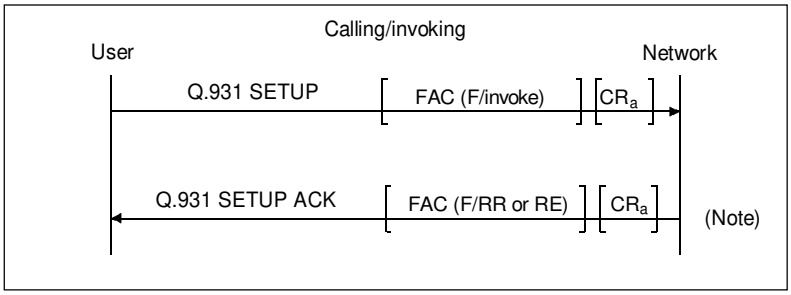
Feature key access – Call progress/Active phase

I.4 Examples of use of the Functional protocol

I.4.1 Call related supplementary service procedures

I.4.1.1 Invocation with call establishment

The example message sequence shows the initiation of a call establishment simultaneously with a supplementary service invocation. See Figure I.8.



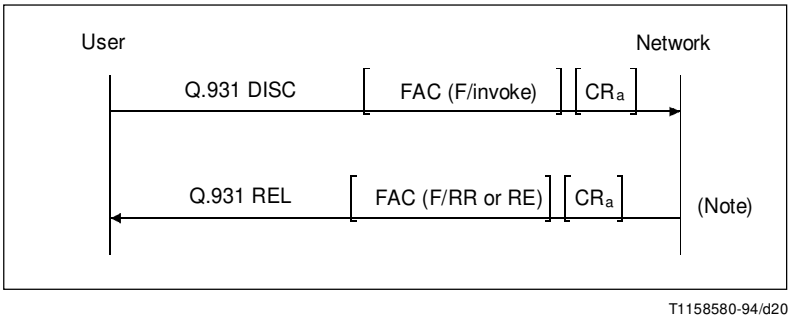
NOTE – Depending on the invoked supplementary service, and the basic call control procedure one of the Recommendation Q.931 messages in the network-to-user direction may be used to carry return result, return error or reject indication or even an invoke for further information.

FIGURE I.8/Q.932

Invocation with call establishment

I.4.1.2 Invocation with call clearing

The example message sequence shows the initiation of normal call clearing simultaneously with a supplementary service invocation. See Figure I.9.



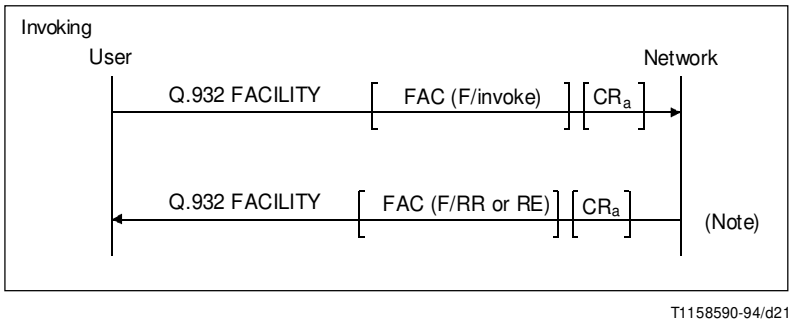
NOTE – Assume the signalling association, CR_a can be cleared together with the connection for the invoked supplementary service, otherwise a FACILITY message may be used instead.

FIGURE I.9/Q.932

Invocation with call clearing

I.4.1.3 Invocation during the active phase of a call

The example message sequence shows the initiation of a supplementary service via the established signalling association CR_a at any time during the active phase of a call. See Figure I.10.



NOTE – This sequence may occur several times during the active phase of a call, utilizing the existing signalling association.

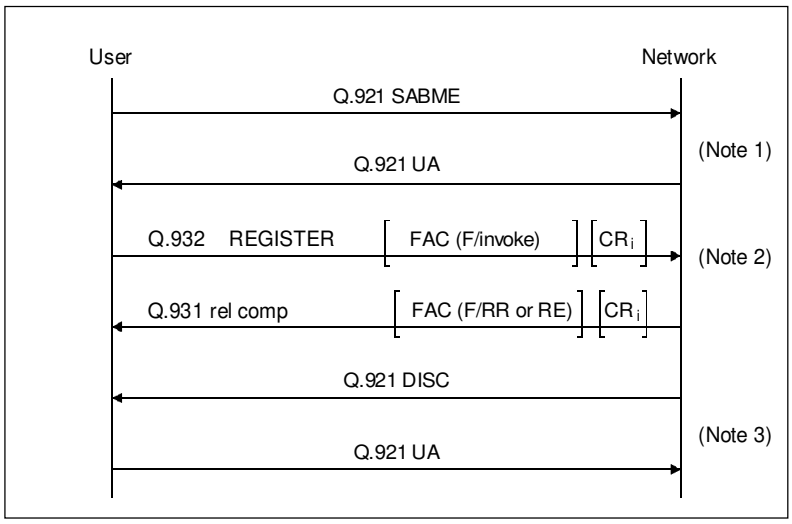
FIGURE I.10/Q.932

Invocation during the active phase of a call

I.4.2 Call independent supplementary service procedures

I.4.2.1 Establishment of a user-to-network transaction for supplementary service control

See Figure I.11.



NOTES

1 Establishment of layer 2 connection if not yet established.

2 If the procedure is used in the network-to-user direction, additional address information may be required. This requires further study.

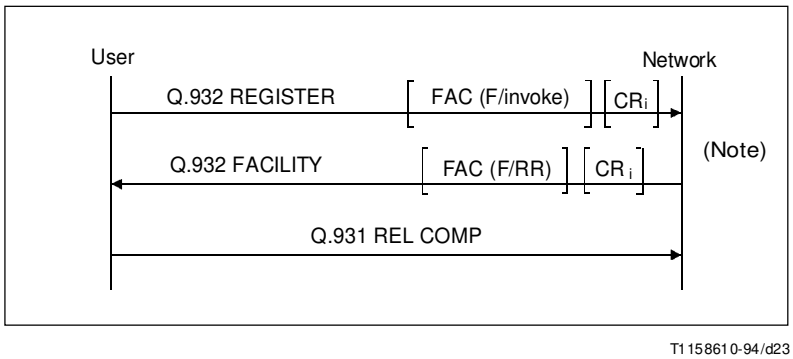
3 Depending on the invoked supplementary service, the layer 2 connection may be kept or cleared.

FIGURE I.11/Q.932

Establishment of a user-to-network transaction for supplementary service control

I.4.2.2 Clearing of a user-to-network transaction for supplementary service control

See Figure I.12.



NOTE – After receiving the last return the receiving side may initiate clearing of the layer 2 connection.

FIGURE I.12/Q.932

Clearing of a user-to-network transaction for supplementary service control

TABLE I.1/Q.932

Key to Figures I.1 to I.12

Layer 2 frames	
SABME	Set asynchronous balance mode extended
UA	Unnumbered acknowledgement frame
DISC	Disconnect frame
Layer 3 messages:	
INFO	Information message
SETUP ACK	Setup acknowledge
DISC	Disconnect
REL	Release
REL COMP	Release complete
Layer 3 message information elements/parameters:	
FAC	Facility information element
F	Facility identifier
Invoke	Invoke operation type
RR	Return result operation type
RE	Return error operation type
CR_a	Call reference of an active call
CR_1	Call reference assigned call independently
BC	Bearer Capability information element
CDN	Called Party Number information element
CRn	Call Reference information element
FA	Feature Activator information element
IRQ	Information Request information element
KP	Keypad facility information element
SIG	Signal information element

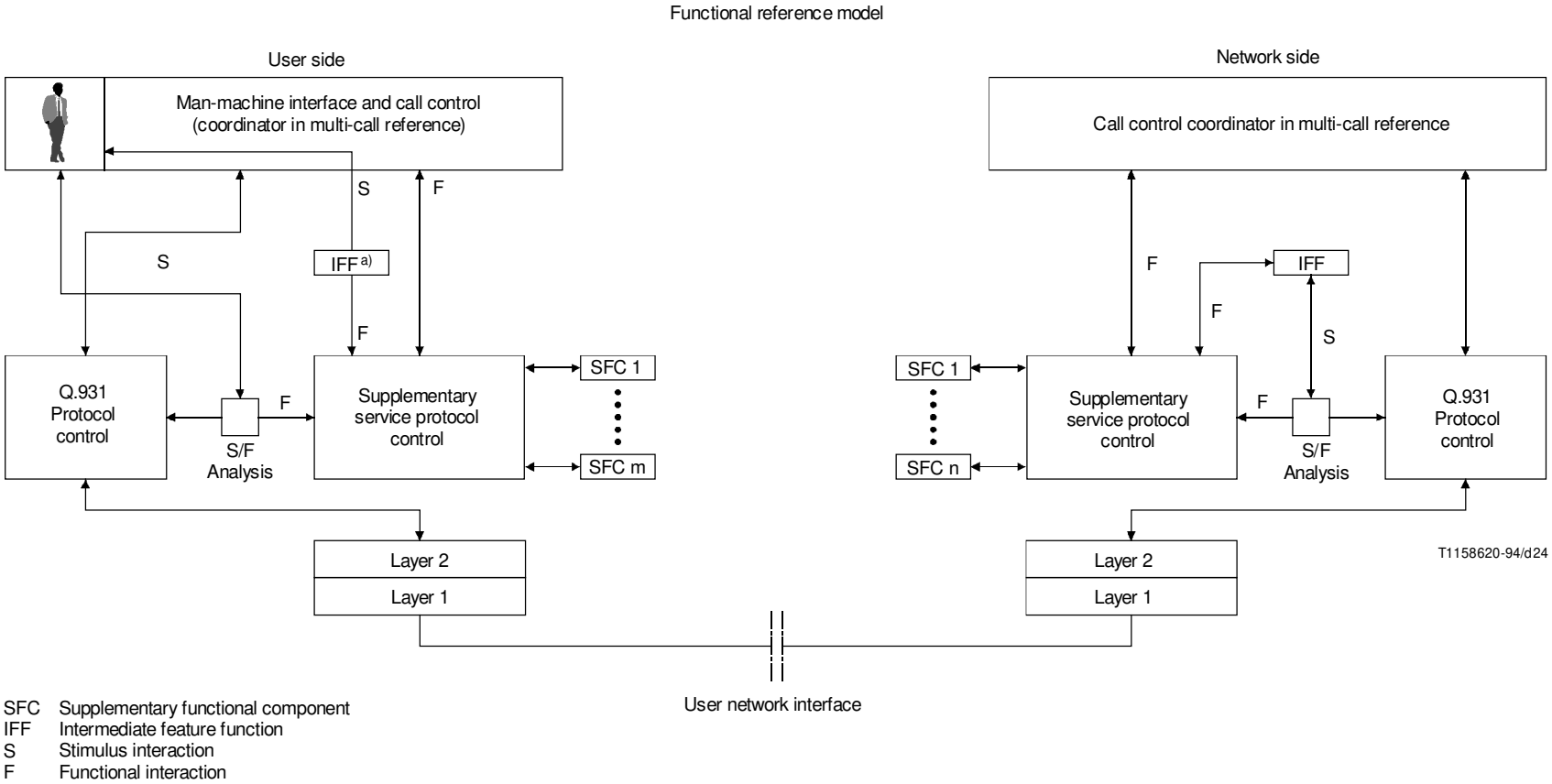
Appendix II

Functional reference model for the operation of supplementary services

(This appendix does not form an integral part of this Recommendation)

This appendix provides a functional model intended to show how the supplementary services can be operated by combining stimulus or Functional protocol types to interact with a unique supplementary service protocol controller which interfaces with the relevant supplementary functional components which provides and coordinates the required functions associated to each supplementary service (e.g. control of resources).

The intermediate feature function performs the necessary conversions between stimulus protocols and the supplementary service functional primitives which are the only ones treated and known from the supplementary service protocol controller. As an example, the intermediate feature function translates an access code received within the Keypad facility information element or a feature identifier number within a Feature activation information element to a supplementary service priority such as hold or retrieve request. See Figure II.1.



a) This is optional at the user side and is implementation dependent.

FIGURE II.1/Q.932

Protocol architecture model

Appendix III

General description of component encoding rules

(This appendix does not form an integral part of this Recommendation)

III.1 General component structure

Each data element within a component has the same structure. A data 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 data element, containing the primary information the data element is intended to convey. Figure III.1 shows an overview of a component and a data element.

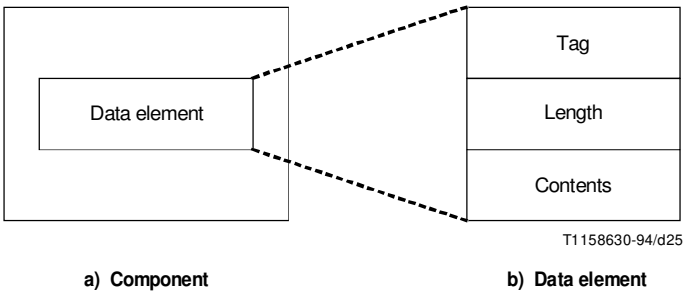


FIGURE III.1/Q.932

Structure of component and data element

Each field is coded using one or more octets. Octets are labelled as shown in Figure III.2. The first octet is the first transmitted. Bits in an octet are labelled as shown in Figure III.3, with bit 1 the least significant and the first transmitted.

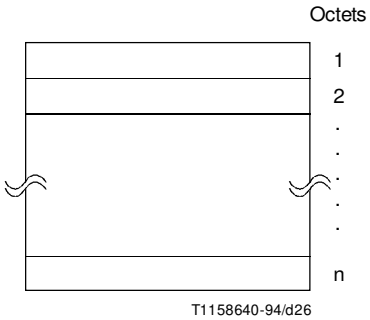


FIGURE III.2/Q.932

Octet labelling scheme

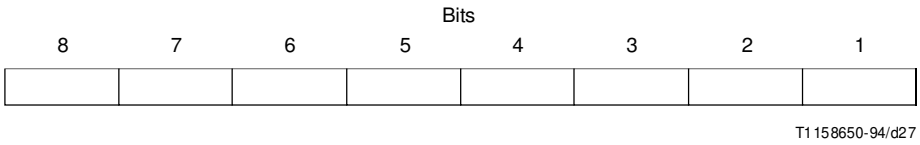


FIGURE III.3/Q.932

Bit labelling scheme

The contents of each data element is either one value (primitive) or one or more data elements (constructor), as shown in Figure III.4.

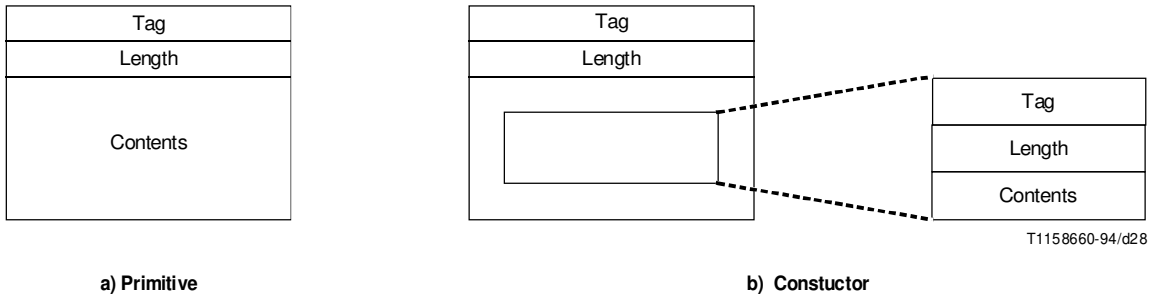
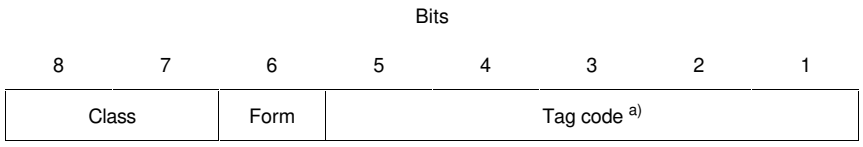


FIGURE III.4/Q.932

Types of contents

III.2 Tag

A data element is first interpreted according to its position within the syntax of the message. The tag distinguishes one data element from another and governs the interpretation of the contents. It is one or more octets in length. The tag is composed of “class”, “form” and “tag code”, as shown in Figure III.5.



a) The tag code may be extended to the following octet(s) as discussed in III.2.3.

FIGURE III.5/Q.932

Format of tag

III.2.1 Tag class

All tags use the two most significant bits (8 and 7) to indicate the tag class. These bits are coded as shown in Table III.1.

TABLE III.1/Q.932

Coding of tag class

Class	Coding (87)
Universal	00
Application-wide	01
Context-specific	10
Private use	11

The universal class is used for tags that are exclusively standardized in Recommendation X.209 and are application independent types. Universal tags may be used anywhere a universal data element type is used. The universal class applies across all CCITT Recommendations, i.e., across Q.932 facility information elements, Signalling System No. 7 ASEs, X.400 MHS, X.500 Directory Services, etc.

The application-wide class is used for data elements that are standardized across all applications (ASEs) using Q.932 facility procedures for supplementary services.

The context-specific class is used for data elements that are specified within the context of the next higher construction and take into account the sequence of other data elements within the same construction. This class may be used for tags in a construction, and the tags may be re-used in any other construction.

The private use class is reserved for data elements specific to a nation, a network or a private user. Such data elements are beyond the scope of this Recommendation.

The Tag codes of the application-wide class not assigned in this Recommendation are reserved for future use.

III.2.2 Form of the data element

Bit 6 is used to indicate whether the data element is “primitive” or “constructor”, as shown in Table III.2. A primitive element is one whose structure is atomic (i.e. one value only). A constructor element is one whose content is one or more data elements which may themselves be constructor elements.

Both forms of elements are shown in Figure III.4.

TABLE III.2/Q.932

Coding of element form

Element form	Coding (6)
Primitive	0
Constructor	1

III.2.3 Tag code

Bits 1 to 5 of the first octet of the tag plus any extension octets represent a tag code that distinguishes one element type from another of the same class. Tag codes in the range 00000 to 11110 (0 to 30 decimal) are provided in one octet.

The extension mechanism is to code bits 1 to 5 of the first octet as 11111. Bit 8 of the following octet serves as an extension indication. If bit 8 of the extension octet is set to 0, then no further octets for this tag are used. If bit 8 is set to 1, the following octet is also used for extension of the tag code. The resultant tag consists of bits 1 to 7 of each extension octet with bit 7 of the first extension octet being most significant and bit 1 of the last extension octet being least significant. Tag code 31 is encoded as 0011111 in bits 7 to 1 of a single extension octet. Higher tag codes continue from this point using the minimum possible number of extension octets.

Figure III.6 shows the detailed format of the tag code.

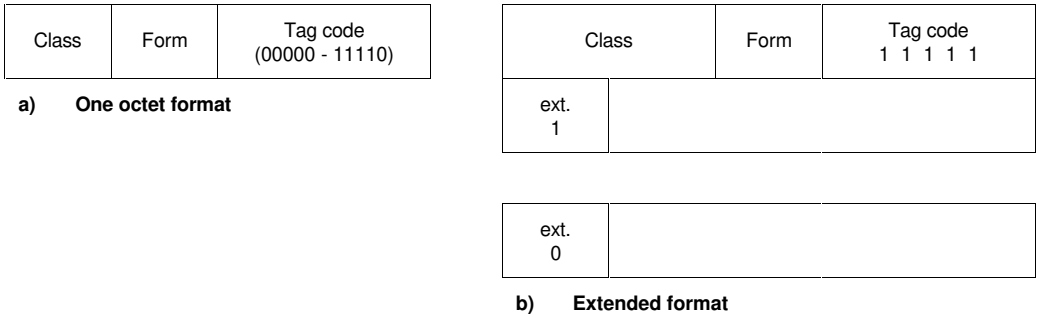


FIGURE III.6/Q.932

Format of the tag code

III.3 Length of the contents

The length of the contents is coded to indicate the number of octets in the contents. The length does not include the tag nor the length of the contents octets.

The length of the contents uses the short, long or indefinite form. If the length is less than 128 octets, the short form is used. In the short form, bit 8 is coded 0, and the length is encoded as a binary number using bits 1 to 7.

If the length of the contents is greater than 127 octets, then the long form of the length of the contents is used. The long form length is from 2 to 127 octets long. Bit 8 of the first octet is coded 1, and bits 1 to 7 of the first octet encode a number one less than the size of the length in octets as an unsigned binary number whose MSB and LSB are bits 7 and 1, respectively. The length itself is encoded as an unsigned binary number whose MSB and LSB are bit 8 of the second octet and bit 1 of the last octet, respectively. This binary number should be encoded in the fewest possible octets, with no leading octets having the value 0.

The indefinite form is one octet long and may (but need not) be used in place of the short or long form, whenever the element is a constructor. It has the value 10000000. When this form is employed, a special end-of-contents (EOC) indicator terminates the contents.

There is no notation for the end-of-contents indicator. Although considered part of the contents syntactically, the end-of-contents indicator has no semantic significance.

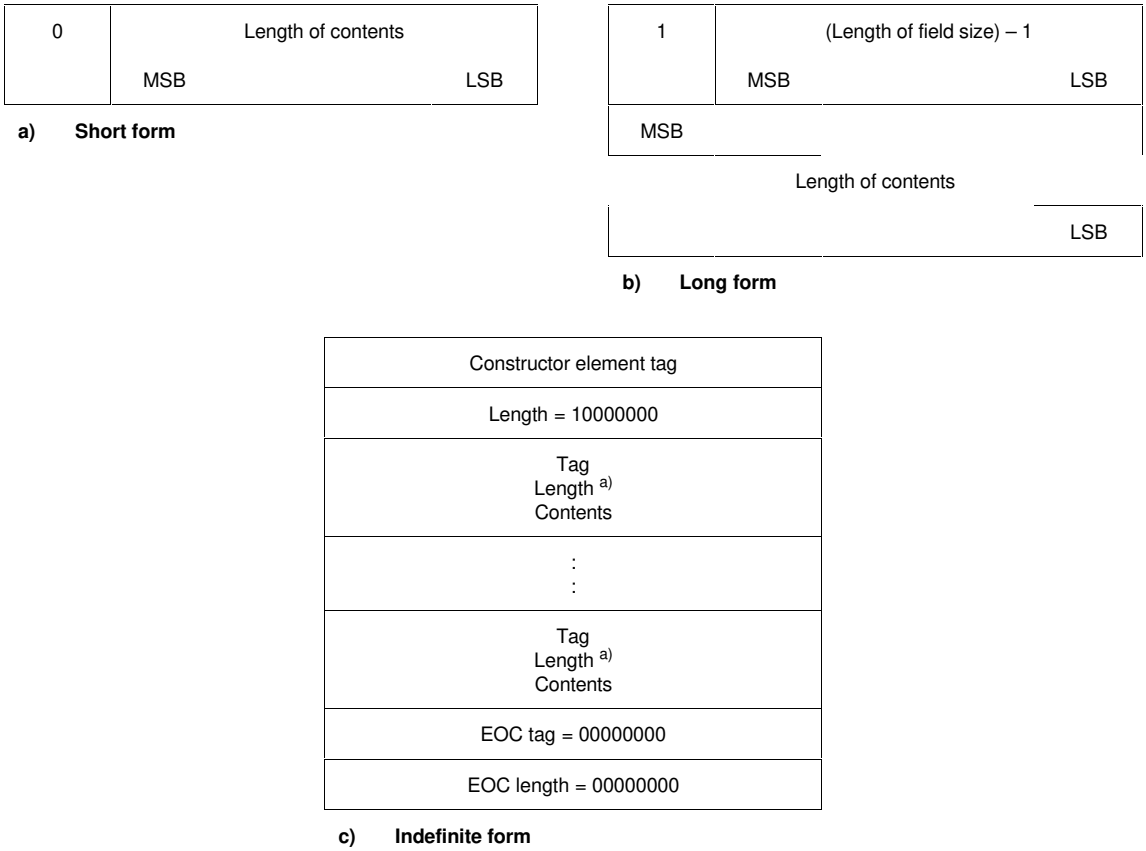
The representation for the end-of-contents indicator is an element whose class is universal, whose form is primitive, whose identifier code has the value 0, and whose contents are unused and absent (see Table III.3).

TABLE III.3/Q.932

Representation for the end-of-contents indicator

EOC	Length	Contents
00 (hex)	00 (hex)	Absent

Figure III.7 shows the formats of the length field described above. The maximum value that may be encoded is constrained by Q.931 information element size limitations.



a) The length may take any of three forms: short, long and indefinite.

FIGURE III.7/Q.932

Format of length field

III.4 Contents

The contents are the substance of the data element and contains the information the data element is intended to convey. Its length is variable, but always an integral number of octets. The contents are interpreted in a type-dependent manner, i.e., according to the tag value.

Appendix IV

Definition of operations, errors and data types

(This appendix does not form an integral part of this Recommendation)

IV.1 Table IV.1 provides a formal definition of operation and error macros. This ASN.1 notation is the same as the one provided in Recommendation X.219 and is only provided here for convienence to the reader.

IV.2 Components

A component is a sequence of data elements each of which is made up of a tag, a length and contents. The component type is indicated by the first octet of the Facility information element component. The component types defined for the Facility information element and their usage are defined in accordance with Recommendations X.219 and X.229 as follows:

- Invoke: The Invoke component is used to initiate the invocation of an operation to be performed by the performing entity (the performer).
- Return result: The Return Result component is used to report a successful outcome of a previously invoked operation.
- Return error: The Return Error component is used to report an unsuccessful outcome of a previously invoked operation. The operation was interpreted successfully and attempted, but the outcome of the operation was unsuccessful.
- Reject: The Reject component is used to reject a request for invocation if a problem was detected. The operation was either not performed at all or was disrupted and terminated abnormally. The reject component may also be used to reject a reply to the invocation (i.e. Return Result or Return Error components). A Reject component should not be sent when a Reject component problem is detected.

IV.2.1 Length of each component or of their data elements

Lengths up to 127 octets are coded using the short form of Recommendation X.209: bit 8 is set to zero and the remaining seven bits are a binary encoding of the length, with bit 1 the least significant bit. (This length encoding is identical to that of Recommendation Q.931 for lengths up to 127 octets.) This is illustrated in Figure IV.1.

If the length of the contents is greater than 127 octets, then the long form of the length of the contents is used. The long form length is from 2 to 127 octets long. Bit 8 of the first octet is coded 1, and bits 1 to 7 of the first octet encode a number one less than the size of the length in octets as an unsigned binary number whose MSB and LSB are bits 7 and 1, respectively. The length itself is encoded as an unsigned binary number whose MSB and LSB are bit 8 of the second octet and bit 1 of the last octet, respectively. This binary number should be encoded in the fewest possible octets, with no leading octets having the value 0. This is illustrated in Figure IV.2.

TABLE IV.1/Q.932

Formal definition of data types

```
Remote-Operation-Notation { joint-iso-ccitt x 219 remote-operations (4) notation (0) }
DEFINITIONS                                ::=
BEGIN

EXPORTS                          OPERATION, ERROR;

OPERATION MACRO                  ::=
BEGIN
TYPE NOTATION                    ::= Argument Result Errors LinkedOperations
VALUE NOTATION          ::= value (VALUE CHOICE {
                                                  localValue INTEGER,
                        -- used for operations defined in
                        -- this series of recommendations

globalValue OBJECT IDENTIFIER
                        -- used for national-specific or
                        -- network specific operations
                                                                  } )

Argument                ::= "ARGUMENT"NamedType | empty
Result     ::= "RESULT" ResultType | empty
Errors     ::= "ERRORS" " { " ErrorNames " } " | empty
LinkedOperations        ::= "LINKED" " { " LinkedOperationNames " " } " | empty
NamedType               ::= identifier type | type
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

END      -- of OPERATION macro

ERROR MACRO             ::=
BEGIN
TYPE NOTATION           ::= Parameter
VALUE NOTATION ::= value (VALUE CHOICE {
                                localValue INTEGER,
                                -- used for errors defined in this series of
                                -- recommendations

globalValue OBJECT IDENTIFIER
          -- used for national-specific or network-specific
          -- errors

Parameter                       ::= "PARAMETER" NamedType | empty

NamedType                       ::= identifier type | type

END          -- of ERROR macro

END          -- of Q.932 Functional Protocol definitions
```

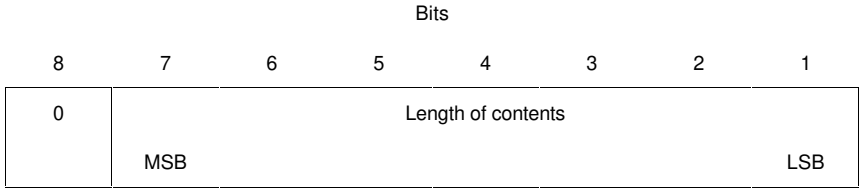
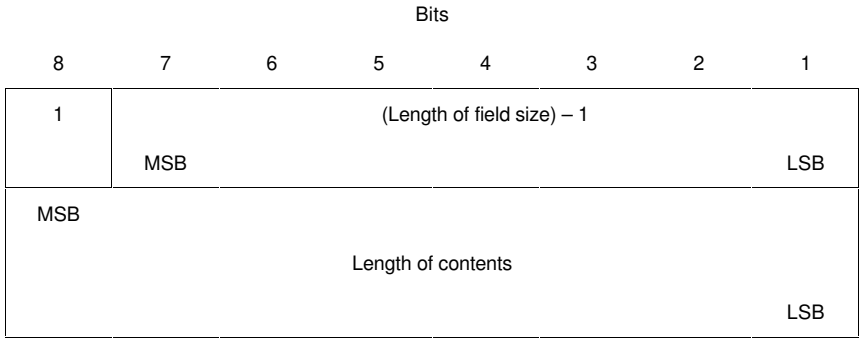


FIGURE IV.1/Q.932

Format of the length field (short form)



NOTE – The application of the indefinite form of the length is not precluded depending on future applications (see III.3).

FIGURE IV.2/Q.932

Format of the length field (long form)

IV.2.2 Component type tag

The coding of the component type tag is shown in Table IV.2.

TABLE IV.2/Q.932

Component type tag

Component type tag	Bits							
	8	7	6	5	4	3	2	1
Invoke	1	0	1	0	0	0	0	1
Return result	1	0	1	0	0	0	1	0
Return error	1	0	1	0	0	0	1	1
Reject	1	0	1	0	0	1	0	0

IV.2.3 Component identifier tags

An invoke identifier is used to identify an operation invocation and is reflected in the return result or return error that responds to it. An invoke may refer to another invoke through the linked identifier. When a protocol error occurs, the invoke identifier is reflected in the reject component, but if it is not available, a null is returned. Invoke and linked identifiers are one octet long. The null has zero length. The coding of the component identifier tags is shown in Table IV.3.

TABLE IV.3/Q.932

Coding of component identifier tag

	Bits							
	8	7	6	5	4	3	2	1
Invoke identifier	0	0	0	0	0	0	1	0
Linked identifier	1	0	0	0	0	0	0	0
Null	0	0	0	0	0	1	0	1

IV.3 Operation value tag

The operation value specifies the facility or supplementary service application and operation being requested. Values are encoded as integers or object identifiers. The value of the operation value is supplementary service specific and will be specified in future Recommendations which contain the protocol for individual supplementary services. The coding for the operation value tag is shown in Table IV.4.

TABLE IV.4/Q.932

Coding of operation value tag

Operation value tag	8	7	6	5	4	3	2	1
Integer (Note 1)	0	0	0	0	0	0	1	0
Object identifier (Note 2)	0	0	0	0	0	1	1	0

NOTES

1 Operation values are defined for individual supplementary services in other Recommendations in this series.

2 Object identifiers are used for national specific, network specific, manufacturer specific or user-specific operations. Object identifiers are used according to Recommendation X.208, Annexes B or C. The notation for object identifiers is defined in 28/X.208 and the coding is defined in 20/X.209.

IV.4 Error value tag

Operations report errors as specified for each individual operation. Values are encoded as integers or object identifiers. The coding for the error value tag is shown in Table IV.5.

TABLE IV.5/Q.932

Coding of error value tag

Error value tag	8	7	6	5	4	3	2	1
Integer (Note 1)	0	0	0	0	0	0	1	0
Object identifier (Note 2)	0	0	0	0	0	1	1	0

NOTES

1 Error values are defined for individual supplementary services in other Recommendations in this series.

2 Object identifiers are used for national specific, network specific, manufacturer specific or user-specific operations. Object identifiers are used according to Annexes B/X.208 or C/X.208. The notation for object identifiers is defined in 28/X.208 and the coding is defined in 20/X.209.

IV.5 Problem tag

Protocol problems are indicated in groups. Table IV.6 indicates the tags for these groups. The contents for each of these tags is indicated in Tables IV.7 to IV.10. The contents of these tags are defined in Table IV.11.

TABLE IV.6/Q.932

Coding of problem tags

Problem	Bits							
	8	7	6	5	4	3	2	1
General problem	1	0	0	0	0	0	0	0
Invoke problem	1	0	0	0	0	0	0	1
Return result problem	1	0	0	0	0	0	1	0
Return error problem	1	0	0	0	0	0	1	1

TABLE IV.7/Q.932

Coding of general problem

	Bits							
	8	7	6	5	4	3	2	1
Unrecognized component	0	0	0	0	0	0	0	0
Mistyped component	0	0	0	0	0	0	0	1
Badly structured component	0	0	0	0	0	0	1	0

NOTE – ROSE uses the term application protocol data unit (APDU) in place of component.

TABLE IV.8/Q.932

Coding of invoke problem

	Bits							
	8	7	6	5	4	3	2	1
Duplicate invocation	0	0	0	0	0	0	0	0
Unrecognized operation	0	0	0	0	0	0	0	1
Mistyped argument	0	0	0	0	0	0	1	0
Resource limitation	0	0	0	0	0	0	1	1
Initiator releasing	0	0	0	0	0	1	0	0
Unrecognized linked identifier	0	0	0	0	0	1	0	1
Linked response unexpected	0	0	0	0	0	1	1	0
Unexpected child operation	0	0	0	0	0	1	1	1

TABLE IV.9/Q.932

Coding of return result problem

	Bits							
	8	7	6	5	4	3	2	1
Unrecognized invocation	0	0	0	0	0	0	0	0
Result response unexpected	0	0	0	0	0	0	0	1
Mistyped result	0	0	0	0	0	0	1	0

TABLE IV.10/Q.932

Coding of return error problem

	Bits							
	8	7	6	5	4	3	2	1
Unrecognized invocation	0	0	0	0	0	0	0	0
Error response unexpected	0	0	0	0	0	0	0	1
Unrecognized error	0	0	0	0	0	0	1	0
Unexpected error	0	0	0	0	0	0	1	1
Mistyped parameter	0	0	0	0	0	1	0	0

TABLE IV.11/Q.932

Problem code definitions

General-problem:	
– unrecognized-component:	Signifies that the type of the component, as evidenced by its type identifier, is not one of the four defined by Recommendation X.229 [10]
– mistyped-component:	Signifies that the structure of the component does not conform to Recommendation X.229
– badly-structured-component:	Signifies that the structure of the component does not conform to the standard notation and encoding, defined in Recommendations X.208 [7] and X.209 [8]
Invoke problem:	
– duplicate-invocation:	Signifies that the invoke-identifier parameter violates the assignment rules of Recommendation X.219 [9]
– unrecognized-operation:	Signifies that the operation is not one of those agreed between the user and the network
– mistyped-argument:	Signifies that the type of the operation argument supplied is not that agreed between the user and the network
– resource-limitation:	The performing user or network is not able to perform the invoked operation due to resource limitation
– initiator-releasing:	The association-initiator is not willing to perform the invoked operation because it is about to attempt to release the application-association
– unrecognized-linked-identifier:	Signifies that there is no operation in progress with an invoke-identifier equal to the specified linked-identifier
– linked-response-unexpected:	Signifies that the invoked operation referred to by linked-identifier is not a parent-operation
– unexpected-child-operation:	Signifies that the invoked child-operation is not one that the invoked parent-operation referred to by the linked-identifier allows
Return-result-problem:	
– unrecognized-invocation:	Signifies that no operation with the specified Invoke-identifier is in progress
– result-response-unexpected:	Signifies that the invoke operation does not report a result
– mistyped-result:	Signifies that the type of the result parameter supplied is not that agreed between the user and the network
Return-error-problem:	
– unrecognized-invocation:	Signifies that no operation with the specified invoke-identifier is in progress
– error-response-unexpected:	Signifies that the invoked operation does not report failure
– unrecognized-error:	Signifies that the reported error is not one of those agreed between the user and the network
– unexpected-error:	Signifies that the reported error is not one that the invoked operation may report
– mistyped-parameter:	Signifies that the type of the error parameters supplied is not that agreed between the user and the network

NOTE – The former definitions are adapted from 7.4.4.2/X.229 and 7.5.4.2/X.229 (Remote operations: protocol specification).

IV.6 Parameters

The parameters included with a component (i.e. the argument with an invoke, the result with a return result or the parameter with a return error) are indicated in the specification of the operation. They may include optional and default parameters. Parameters shall be one of the following:

– a sequence of parameters;

– a set of parameters;

– a specific parameter with its own tag;

– nothing at all (i.e. absent).

When more than one parameter is required, they shall follow a sequence or set tag as specified in the specification of the operation. (The usage of the sequence and set tags is defined in Recommendations X.208 and X.209.)

Sequences and sets of parameters may contain further sequences and sets as specified for the operation to be performed. Table IV.12 indicates the coding of the sequence and set tags.

TABLE IV.12/Q.932

Coding of sequence and set tags

	Bits							
	8	7	6	5	4	3	2	1
Sequence tag	0	0	1	1	0	0	0	0
Set tag	0	0	1	1	0	0	0	1

Appendix V

List of object identifiers defined in Recommendation Q.932

(This appendix does not form an integral part of this Recommendation)

Object identifier	*Reference*
facility-information-element-component (3)	Table 8-7
explicit-network-controlled-channel-reservation (4)	Table 6-1
embedded-q931 types (5)	Table 8-9
notification-data-structure (6)	Table 8-15
addressing-data-elements (7)	Annex C

Acronyms used in this Recommendation

	Meaning
ACSE	Association Control Service Element
APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation One (see Recommendations X.208 and X.209)
CEI	Connection Endpoint Identifier (see Recommendation Q.920)
CES	Connection Endpoint Suffix (see Recommendation Q.920)

IA5	International Alphabet No. 5
ISDN	Integrated Services Digital Network
LSB	Least Significant Bit
MSB	Most Significant Bit
NT2	Network Termination Type Two (see Recommendation I.411)
ROSE	Remote Operations Service Element (see Recommendations X.219 and X.229)
SAPI	Service Access Point Identifier (see Recommendation Q.920)
SPID	Service Profile Identifier
TEI	Terminal Endpoint Identifier (see Recommendation Q.920)
TID	Terminal Identifier
USID	User Service Identifier

References

[1] CCITT Recommendation *Basic user-network interface – Layer 1 specification*, Rec. I.430.

[2] CCITT Recommendation *Primary rate user-network interface – Layer 1 specification*, Rec. I.431.

[3] CCITT Recommendation *ISDN user-network interface – Data link layer specification*, Rec. Q.921.

[4] CCITT Recommendation *ISDN user-network interface layer 3 specification for basic call control*, Rec. Q.931.

[5] CCITT Recommendation *ISDN user-network interface layer 3 – General aspects*, Rec. Q.930.

[6] CCITT Recommendation *ISDN user-network interface data link layer – General aspects*, Rec. Q.920.

[7] CCITT Recommendation *Specification of abstract syntax notation specification of abstract syntax notation one (ASN.1)*, Rec. X.208.

[8] CCITT Recommendation *Specification of basic encoding rules for abstract syntax notation one (ASN.1)*, Rec. X.209.

[9] CCITT Recommendation *Remote operations: model, notation and service definition*, Rec. X.219.

[10] CCITT Recommendation *Remote operations: protocol specification*, Rec. X.229.

[11] CCITT Recommendation *ISDN user-network interface protocol profile for management*, Rec. Q.941.

[12] CCITT Recommendation *Information technology – Open systems interconnection – Service definition for the association control service element*, Rec. X.217.

[13] CCITT Recommendation *Information technology – Open systems interconnection – Protocol specification for the association control service element*, Rec. X.227.