

Intel[®] NetStructure[™] SS7 Protocols IS-41 Programmer's Manual

Document Reference: U17SSS

Disclaimer

The product may contain design defects or errors known as errata, which may cause the product to deviate from published specifications.

Information in this document is provided in connection with Intel® products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not designed, intended or authorized for use in any medical, life saving, or life sustaining applications or for any other application in which the failure of the Intel product could create a situation where personal injury or death may occur. Intel may make changes to specifications and product descriptions at any time, without notice.

Intel and Intel NetStructure are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

* Other names and brands may be claimed as the property of others.

Copyright © 1993-2003 Intel Corporation. All rights reserved. No part of this document may be copied, or reproduced in any form, or by any means without prior written consent of Intel.

REVISION HISTORY

ISSUE	DATE	BY	CHANGES
1	08-Feb-99	RBP	Initial release.
2	14-Jun-99	JET	Changes to reflect new message ids. Addition of a table of messages.
3	16-Jul-03	ML	Branding changed: references to System7 removed.

CONTENTS

1. INTRODUCTION	6
2. ABBREVIATIONS	7
3. RELATED DOCUMENTATION	7
4. FEATURE OVERVIEW.....	7
5. GENERAL DESCRIPTION	8
5.1 Module Overview	8
5.2 Module Configuration.....	8
5.3 Dialogue ID assignment.....	9
5.4 Application Context.....	10
5.5 Invoke ID.....	10
5.6 Constant definitions.	10
5.7 Module Dimensions.	10
6. INTERFACE TO SYSTEM SERVICES.....	11
6.1 System Functions	11
6.2 Timer Operation.....	11
7. INTERFACE TO TCAP	12
7.1 Dialogue handling :	12
7.2 Component handling :.....	12
8. INTERFACE TO USER APPLICATION.....	13
8.1 Introduction	13
8.2 Primitive parameters.....	14
8.3 Dialogue Primitive Types	15
8.4 Service Primitive Types	16
8.5 IS-41 Dialogue Request.....	18
8.6 IS-41 Dialogue Indication.....	21
8.7 IS-41 Dialogue Primitive Parameters.....	24
8.8 IS-41 Service Request	26
8.9 IS-41 Service Indication	33
8.10 IS-41 Service Primitive Parameters	35
9. NON-PRIMITIVE INTERFACE	41
9.1 IS-41 Configuration Request.....	42
9.2 IS-41 Timer Configuration Request	45
9.3 IS-41 Software Event Indication.....	47
9.4 IS-41 Trace Mask Request	49
9.5 Trace Event Indication	53
9.6 IS-41 Maintenance Mask Request.....	54
9.7 IS-41 Software Event Mask Request.....	55
APPENDIX 1.....	56
A1 Tick Timer message format.....	56

APPENDIX 2.....	57
A2 Message Type reference.....	57

1. INTRODUCTION

The IS-41 module is a portable software implementation of the ANSI/TIA/EIA-41.5-D Cellular Radiotelecommunications Intersystem Operations specification.

It implements a “Service Provider” interface to the IS-41 network for User applications. This provides a primitive interface for control of dialogues to remote systems and the invocation of operations at those systems. The IS-41 module interface to the user application is similar to the “MAP Provider block” specified in GSM ETS 300-599 Phase 2.

The IS41 module uses the services provided by the underlying Transaction Capabilities (TCAP) service for the transfer of operations between peer systems.

The IS41 module provides services to User applications whilst remaining independent of both the TCAP layer and the User application.

This Programmer's Manual is intended for users developing their own applications that will interface to and make use of the functionality provided by the IS-41 module.

The IS-41 module is an event driven task that uses standard structured message types for communication with other layers of the protocol stack. These messages are used to convey the protocol primitives between the IS-41 module and the User application and between the IS-41 module and TCAP. Each message contains the primitives and parameters defined in the appropriate ITU or ANSI recommendations.

This manual provides an overview of the internal operation of the IS-41 module and defines the structure of all messages used to interface to the IS-41 module.

2. ABBREVIATIONS

ANSI	American National Standards Institute.
APDU	Application Protocol Data Unit.
CCITT	The International Telegraph & Telephone Consultative Committee.
ITU-T	International Telecommunication Union (formerly CCITT).
MTP	Message Transfer Part.
SCCP	Signalling Connection Control Part.
TCAP	Transaction Capabilities Application Part.
MAP	Mobile Application Part.

3. RELATED DOCUMENTATION

- [1] ANSI/TIA/EIA-41.5-D - Cellular Radiotelecommunications Intersystem Operations
- [2] ETSI ETS 300 599 - Mobile Application Part (MAP) Specification (GSM 09.02)
- [3] U06SSS - TCAP Programmer's Manual.
- [4] U10SSS - Software Environment Programmer's Manual.

4. FEATURE OVERVIEW

Key features of the IS-41 module include:

- Primitive interface providing full control of dialogues to remote systems.
- Full dialogue management and error handling
- Primitive interface providing full access to Short Message Service operations.
- Full operation invocation management and error handling.
- Message oriented interface.

5. GENERAL DESCRIPTION

5.1 Module Overview

The IS-41 module is a partial implementation of the ANSI/TIA/EIA-41.5-D specification signalling protocols. The IS-41 module provides full control of dialogues with a peer IS-41 systems. The IS-41 module currently supports full access to the Short Message Service operations and associated parameters.

The ANSI/TIA/EIA-41.5-D specification signalling procedures are the responsibility of the user application.

Whenever the User application wishes to invoke an operation at a remote peer a dialogue must be opened with the peer. The required operations are invoked via this dialogue. Any results or errors sent by the remote system are also received via this dialogue. Generally the remote system will close the dialogue at this point, however certain scenarios specified in IS41 permit the dialogue to be kept open for subsequent operation invocations. The IS-41 module provides the User application with full control of the dialogues via a dialogue primitive interface. Full access to operation invocation and result parameters is provided via a service primitive interface.

The module is event driven. It has a single input queue into which events from other modules (TCAP, User application, management etc.) are written. The module processes each event in turn until the input queue is empty in which case it will do nothing until the next event is received. Output from the module is directed depending on the type of event to either the TCAP module, the User application module, the Management module or the Maintenance module.

In addition the module requires a periodic timer tick notification be issued to it via the input queue. Typically this is required every tenth of a second. This can either be generated by a timer module or using the services of the selected operating system.

5.2 Module Configuration

The module provides maximum flexibility by allowing a large number of user configuration options to be set up at run time. This allows the users to customise the operation of the module to suit the requirements of the final application. All configuration parameters are sent to the module's input event queue in the same manner as IS-41 protocol messages.

The first message sent to the module must be the global configuration message. Any messages received prior to the global configuration message will be discarded. The global configuration message specifies the module id for all modules to which IS-41 issues messages. It also supplies values for the maximum number of dialogues (incoming and outgoing), base values for the incoming and outgoing ranges of dialogue ids and the maximum number of simultaneously active operation invocations that are required. The module checks that the values requested are compatible with the internal dimensions it has been built with.

The IS-41 module must also be sent a timer configuration message. The time for which the IS-41 module will wait for a User application response may be supplied as a message parameter. The message may also be sent without parameters in order to use a default value.

5.3 Dialogue ID assignment

The IS-41 module may support a number of active dialogues at a time. User application primitives are associated with a particular dialogue using a Dialogue ID.

The Dialogue ID is assigned when the opening primitive is exchanged between the User application and the IS-41 module. For a dialogue initiated by the User application (an 'outgoing dialogue'), the value is selected by the User application. For a dialogue initiated by a remote system (an 'incoming dialogue'), the value is selected by the IS-41 module. Once a dialogue has started, all user primitives that refer to this dialogue must include its Dialogue ID value.

The dialogue ID is a 16-bit value. Separate ranges of values must be used for outgoing and incoming dialogues. The global configuration message sent to the module on initialisation specifies the range of IDs available to IS-41 for incoming dialogues. As well as the range of IDs available to the IS-41 User for outgoing dialogues.

The Dialogue ID is of purely local significance between the User application and IS-41.

Maximum values for the number of incoming and outgoing dialogues that the IS-41 can support are set up at configuration time. These values are then fixed for both TCAP and User application interfaces.

The dialogue ID selected by the User application for an outgoing dialogue must lie within the configured range of outgoing dialogue IDs. Dialogue IDs for incoming dialogues are allocated automatically by the IS-41 module from its configured range of incoming dialogue IDs. The incoming dialogue ID that has been unused for the longest period is used next. It's important that IS-41 and the User application use separate ranges of IDs so that is not possible for the User application to select an ID for an outgoing dialogue at the same instant that IS-41 selects the same ID for use with an incoming dialogue.

Dialogue id values also exist on the IS-41 to TCAP interface. The User application is not concerned with these. However both the IS-41 module and the TCAP module must be configured to use separate ranges of dialogue IDs for incoming and outgoing dialogues. The dialogue IDs used on this interface are of entirely local significance to IS-41 and TCAP.

5.4 Application Context

The ANSI/TIA/EIA specifications do not specify the use of application context for identifying TCAP dialogues.

5.5 Invoke ID

When the User application has established a dialogue with a peer, it may invoke an operation at a remote peer by sending a service request primitive to the IS-41 module. Any subsequent response from the peer will be indicated to the user in a service confirmation primitive.

Operation invocations from the remote peer are indicated to the User application in service indication primitives. Any response the user wishes to send in reply should be issued in a service response primitive.

An “invoke id” must be present in a service request or indication primitive. This invoke id serves to identify a particular invocation of IS41 operation.

Service response and confirmation primitives are associated with a particular invocation of an IS41 operation using this invoke id.

The invoke id must be unique for each service request on a dialogue. The invoke id is transported transparently by the IS41 module. The User application is responsible for ensuring unambiguous use of invoke id with the remote system.

5.6 Constant definitions.

To assist the user when writing an application, a ‘C’ language header file (*is41_inc.h*) is available containing all the definitions and constants necessary to interface with the IS-41 module. This file contains definitions for all the mnemonics listed in this Programmer’s Manual.

5.7 Module Dimensions.

Internally there are a number of data structures used by the module.

The maximum dimensions of these structures are determined by compile time constants. The two constants of importance to the user are:

- a) The maximum number of simultaneous dialogues supported by the module.
- b) The maximum number of simultaneous invocations supported by the module.

6. INTERFACE TO SYSTEM SERVICES

6.1 System Functions

In addition to the primitive interface and the management interface to the TCAP module (which are described in later sections) the module requires a few basic system services to be supplied by the underlying operating system.

The following functions are required for inter-task communication:

- | | |
|--------------------|---|
| GCT_send | Sends a message to another task. |
| GCT_receive | Accept next message from input event queue, blocking the task if no message is ready. |
| GCT_grab | As receive but not blocking if no message is ready. |

The following functions are required for allocation of inter-task messages

- | | |
|---------------|---------------------------------------|
| getm() | Allocate a message from the system. |
| relm() | Release block of memory to partition. |

These functions are described in the Software Environment Programmer's Manual.

6.2 Timer Operation

The IS-41 module requires a periodic tick timer message "TIM_EXP" to drive its internal timer mechanism.

This message should be sent to the IS-41 modules input queue at periods that may be selected at compile time. Normally a default period of 100ms is used.

The format of this message is given in Appendix 1.

7. INTERFACE TO TCAP

The IS-41 module is usually used in conjunction with the TCAP module. However, the use of primitives in accordance with Q.771 ensures that it can also be integrated with other Transaction Capabilities implementations if required.

The IS-41 module communicates with the TCAP using the following primitives, all of which are defined in CCITT Recommendation Q.771:

7.1 Dialogue handling :

UNI-REQ	UNI-IND
QUERY-REQ	QUERY-IND
CONVERSATION-REQ	CONVERSATION -IND
RESPONSE-REQ	RESPONSE -IND
U-ABORT-REQ	U-ABORT-IND
	P-ABORT-IND
	NOTICE-IND

The message format used to convey these primitives is defined in the TCAP Programmer's Manual. The following messages are used:

TCP_MSG_DLG_REQ	Messages issued by IS-41
TCP_MSG_DLG_IND	Messages issued to IS-41

7.2 Component handling :

INVOKE-REQ	NULL-IND
RESULT-L-REQ	INVOKE-IND
RESULT-NL-REQ	RESULT-L-IND
U-ERROR-REQ	RESULT-NL-IND
U-CANCEL-REQ	U-ERROR-IND
U-REJECT-REQ	L-CANCEL-IND
	L-REJECT-IND
	R-REJECT-IND
	U-REJECT-IND

The message format used to convey these primitives is defined in the TCAP Programmer's Manual. The following messages are used:

TCP_MSG_CPT_REQ	Messages issued by IS-41
TCP_MSG_CPT_IND	Messages issued to IS-41

8. INTERFACE TO USER APPLICATION

8.1 Introduction

All primitives exchanged between the User application and IS-41 module are passed by inter-process messages.

The following messages are used:

- IS41-DIALOGUE-REQ Transfers dialogue request primitives and dialogue response primitives from User application to IS-41.
- IS41-DIALOGUE-IND Transfers dialogue indication primitives and dialogue confirmation primitives from IS-41 to User application.
- IS41-SERVICE-REQ Transfers service request primitives and service response primitives from User application to IS-41.
- IS41-SERVICE-IND Transfers service indication primitives and service confirmation primitives from IS-41 to User application.

The basic structure of each message (irrespective of the IS-41 primitive contained within it) is the same and is described in Appendix A of the Software Environment Programmers' Manual.

The message must be contained in a single buffer which should be allocated by the sending module (using the **getm** function) and either released (using the **relm** function) or passed to another module by the receiving module. The **getm** and **relm** functions are described in Section 6.

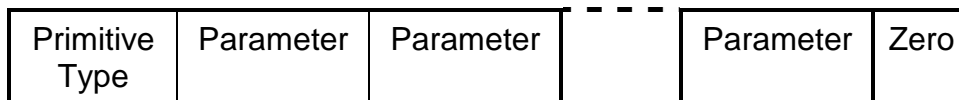
The message header contains a 'type', the value of this parameter indicating the primitive that is being conveyed by the message. The following message types have been defined:

Primitive	Message type	Value
IS41-SERVICE-REQ	IS41_MSG_SRV_REQ	0xc7b0
IS41-SERVICE-IND	IS41_MSG_SRV_IND	0x87b1
IS41-DIALOGUE-REQ	IS41_MSG_DLG_REQ	0xc7b2
IS41-DIALOGUE-IND	IS41_MSG_DLG_IND	0x87b3

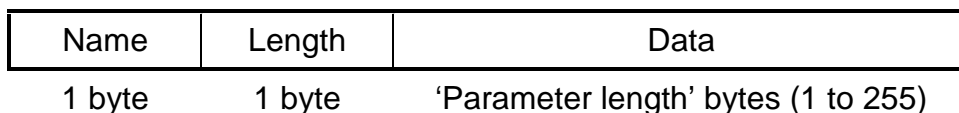
8.2 Primitive parameters

Each User application primitive includes a number of parameters. These parameters are conveyed in the parameter area of the message that conveys the primitive.

The first byte in the parameter area is the primitive type octet and the last byte is a zero byte to indicate that there are no further parameters in the parameter area. Any parameters associated with the message are placed between the message type code and the final (zero) byte. Therefore the parameter area is formatted as follows:



The parameters may be placed in any order. The first byte of a parameter is the parameter name, the second byte is the length of the parameter data to follow (excluding the parameter name and the length byte itself), this is followed by the parameter data. The encoding of the parameter data aligns exactly with the parameter format specified in the appropriate ANSI recommendation whenever possible. Therefore each parameter is formatted as follows:



Within each message there are mandatory parameters which must always be present and optional parameters which may or may not be present.

8.3 Dialogue Primitive Types

Dialogue handling primitives allow the User application to establish and control dialogues with remote systems.

Dialogue primitives sent by the User application to the IS-41 module. These convey a dialogue request or response:

Primitive	Mnemonic	Value (dec)	Value (hex)
IS41-OPEN-REQ	IS41DT_OPEN_REQ	1	0x01
IS41-CLOSE-REQ	IS41DT_CLOSE_REQ	2	0x02
IS41-DELIMITER-REQ	IS41DT_DELIMITER_REQ	3	0x03
IS41-U-ABORT-REQ	IS41DT_U_ABORT_REQ	4	0x04
IS41-OPEN-RSP	IS41DT_OPEN_RSP	129	0x81

Dialogue primitives sent by the IS-41 module to the User application. These convey a dialogue indication, or confirmation:

Primitive	Mnemonic	Value (dec)	Value (hex)
IS41-OPEN-IND	IS41DT_OPEN_IND	1	0x01
IS41-CLOSE-IND	IS41DT_CLOSE_IND	2	0x02
IS41-DELIMITER-IND	IS41DT_DELIMITER_IND	3	0x03
IS41-U-ABORT-IND	IS41DT_U_ABORT_IND	4	0x04
IS41-OPEN-CNF	IS41DT_OPEN_CNF	129	0x81
IS41-P-ABORT-IND	IS41DT_P_ABORT_IND	5	0x05
IS41-NOTICE-IND	IS41DT_NOTICE_IND	6	0x06

The following sections define the message format and content of the parameter area for each of the messages exchanged between the User application and the IS-41 module.

8.4 Service Primitive Types

Once a dialogue is established between two IS-41 peers, they may invoke operations of each other.

The User application invokes an operation using a service request primitive.

The User application responds to an operation invoked at the User application by a remote IS-41 system using a service response primitive.

The following table details service request and response primitives sent by the User application to the IS-41 module.

Primitive	Mnemonic	Value (dec)	Value (hex)
IS41-SMS-DELIVERY- BACKWARD-REQ	IS41ST_SMS_DB_REQ	1	0x01
IS41-SMS-DELIVERY- BACKWARD-RSP	IS41ST_SMS_DB_RSP	129	0x81
IS41-SMS-DELIVERY- FORWARD-REQ	IS41ST_SMS_DF_REQ	2	0x02
IS41-SMS-DELIVERY- FORWARD-RSP	IS41ST_SMS_DF_RSP	130	0x82
IS41-SMS-DELIVERY-POINT- TO-POINT-REQ	IS41ST_SMS_DPTP_REQ	3	0x03
IS41-SMS-DELIVERY-POINT- TO-POINT-RSP	IS41ST_SMS_DPTP_RSP	131	0x83
IS41-SMS-NOTIFICATION- REQ	IS41ST_SMS_N_REQ	4	0x04
IS41-SMS-NOTIFICATION- RSP	IS41ST_SMS_N_RSP	132	0x84
IS41-SMS-REQUEST-REQ	IS41ST_SMS_R_REQ	5	0x05
IS41-SMS-REQUEST-RSP	IS41ST_SMS_R_RSP	133	0x85

The User application receives operation invocations in a service indication primitive.

The User application receives the results or any errors from invoking an operation at a remote IS-41 system in a service confirmation primitive.

The following table details service indication and confirmation primitives sent by the IS-41 module to the User application.

Primitive	Mnemonic	Value (dec)	Value (hex)
IS41-SMS-DELIVERY- BACKWARD-IND	IS41ST_SMS_DB_IND	1	0x01
IS41-SMS-DELIVERY- BACKWARD-CNF	IS41ST_SMS_DB_CNF	129	0x81
IS41-SMS-DELIVERY- FORWARD-IND	IS41ST_SMS_DF_IND	2	0x02
IS41-SMS-DELIVERY- FORWARD-CNF	IS41ST_SMS_DF_CNF	130	0x82
IS41-SMS-DELIVERY-POINT- TO-POINT-IND	IS41ST_SMS_DPTP_IND	3	0x03
IS41-SMS-DELIVERY-POINT- TO-POINT-CNF	IS41ST_SMS_DPTP_CNF	131	0x83
IS41-SMS-NOTIFICATION- IND	IS41ST_SMS_N_IND	4	0x04
IS41-SMS-NOTIFICATION- CNF	IS41ST_SMS_N_CNF	132	0x84
IS41-SMS-REQUEST-IND	IS41ST_SMS_R_IND	5	0x05
IS41-SMS-REQUEST-CNF	IS41ST_SMS_R_CNF	133	0x85

8.5 IS-41 Dialogue Request

Synopsis:

Message sent from the User application to the IS-41 module containing a dialogue request or response primitive.

Message Format:

MESSAGE HEADER		
FIELD NAME	MEANING	
type	IS41_MSG_DLG_REQ (0xc7b2)	
id	Dialogue_ID	
src	Sending module_id	
dst	IS41_TASK_ID	
rsp_req	0	
hclass	0	
status	0	
err_info	0	
len	Number of bytes of user data	
PARAMETER AREA		
OFFSET	SIZE	NAME
0	1	Dialogue primitive type octet.
1	len - 2	Parameters in Name-Length-Data format.
len - 1	1	Set to zero indicating end of message.

Description:

This message is used by the User application to send dialogue primitives to the IS-41 module.

All dialogue primitives contain a dialogue ID which is encoded in the message header. It does not form part of the parameter area. It must be provided by the User application with the IS41-OPEN-REQ primitive and used in the message header of all subsequent dialogue & service primitives associated with that dialogue.

Parameter area contents:

The dialogue primitive type octet is coded as defined in Section 8.3 Dialogue Primitive Types.

The coding of the dialogue primitive parameters is given in Section

8.7 IS-41 Dialogue Primitive Parameters

The following table lists the parameters associated with each dialogue request primitive and shows whether the parameter is Mandatory (M), in which case the message will be discarded if the parameter is omitted, or Optional (O), in which case the parameter is not essential.

Parameter	IS-41 Primitive				
	O P E N - R E Q	C L O S E - R E Q	D E L I M I T E R - R E Q	U - A B O R T - R E Q	O P E N - R S P
Destination address	M				
Originating address	O				
Result					M
Release method		M			
User reason				M	O

8.6 IS-41 Dialogue Indication

Synopsis:

Protocol message sent from IS-41 to the User application containing a dialogue indication or confirmation primitive.

Message Format:

MESSAGE HEADER		
FIELD NAME	MEANING	
type	IS41_MSG_DLG_IND (0x87b3)	
id	Dialogue_ID	
src	IS41_TASK_ID	
dst	Sending module_id	
rsp_req	0	
hclass	0	
status	0	
err_info	0	
len	Number of bytes of user data	
PARAMETER AREA		
OFFSET	SIZE	NAME
0	1	Dialogue primitive type octet.
1	len - 2	Parameters in Name-Length-Data format.
len - 1	1	Set to zero indicating end of message.

Description:

The IS-41 sends dialogue indication and confirmation primitives to the User application. The primitives that may be sent are detailed in section 8.3 Dialogue Primitive Types.

Indication primitives are sent to indicate incoming dialogue control and problem reports. The IS-41 module indicates incoming dialogue open, close, abort, notice (problem report) and delimiter (end of a group of service requests) primitives.

The User application must confirm the open dialogue primitive with the remote peer before accepting service requests from it.

All dialogue primitives must contain the dialogue ID of the dialogue to which they refer. This is encoded in the message header. It does not form part of the parameter area. Incoming dialogue requests will have a dialogue id assigned by the IS-41 module.

Parameter area contents:

The dialogue primitive type octet is coded as defined in Section 8.3 Dialogue Primitive Types.

The coding of the dialogue primitive parameters is given in Section

8.7 IS-41 Dialogue Primitive Parameters

The following table lists the parameters associated with each dialogue indication or confirmation primitive and shows whether the parameter is Mandatory (M), in which case the message will be discarded if the parameter is omitted, or Optional (O), in which case the parameter is not essential.

Parameter	IS-41 Primitive						
	O P E N - I N D	C L O S E - I N D	D E L I M I T E R - I N D	U - A B O R T - I N D	P - A B O R T - I N D	O P E N - C N F	N O T I C E - I N D
Destination address	M						
Originating address	O						
Result						M	
User reason				M			
Provider reason					M	O	
Source					M		
Problem diagnostic							M

Key

- M Mandatory The message will be discarded if the corresponding parameter is omitted
- O Optional The parameter is not essential

8.7 IS-41 Dialogue Primitive Parameters

The following parameter names are defined for use in dialogue primitive messages:

Parameter	Mnemonic	Value (dec)	Value (hex)
Destination address	IS41PN_dest_address	1	0x01
Originating address	IS41PN_orig_address	3	0x03
Result	IS41PN_result	5	0x05
Release method	IS41PN_release_method	7	0x07
User reason	IS41PN_user_rsn	8	0x08
Provider reason	IS41PN_prov_rsn	9	0x09
Source	IS41PN_source	12	0x0c
Problem diagnostic	IS41PN_prob_diag	13	0x0d

The coding for each parameter type is given in the following tables:

Parameter name	IS41PN_dest_address
Parameter length	Variable, in the range 2 to 18
Parameter data	Destination address parameter encoded in the format expected by the network layer (e.g. when using SCCP, in accordance with Q.713 definition of "Called party address", starting with the address indicator and containing, optionally, signalling point code, subsystem number and global title).

Parameter name	IS41PN_orig_address
Parameter length	Variable, in the range 2 to 18
Parameter data	Origination address parameter encoded in the format expected by the network layer (e.g. when using SCCP, in accordance with Q.713 definition of "Called party address", starting with the address indicator and containing, optionally, signalling point code and global title).

Parameter name	IS41PN_result
Parameter length	Fixed, set to 1
Parameter data	0 – accept 1 – dialogue refused

Parameter name	IS41PN_release_method
Parameter length	Fixed, set to 1
Parameter data	0 – normal release 1 – prearranged end

Parameter name	IS41PN_user_rsn
Parameter length	Variable
Parameter data	Any user specific data

Parameter name	IS41PN_prov_rsn
Parameter length	Fixed, set to 1
Parameter data	0 – provider malfunction 1 – supporting dialogue/transaction released 2 – resource limitation 3 – maintenance activity 4 – version incompatibility 5 – abnormal IS-41 dialogue

Parameter name	IS41PN_source
Parameter length	Fixed, set to 1
Parameter data	0 – IS-41 problem 1 - TC problem 2 – network service problem

Parameter name	IS41PN_prob_diag
Parameter length	Fixed, set to 1
Parameter data	0 – abnormal event detected by peer 1 – response rejected by peer 2 – abnormal event received from peer

8.8 IS-41 Service Request

Synopsis:

Protocol message sent from the User application to the IS-41 module containing a single service specific request primitive.

Message Format:

MESSAGE HEADER		
FIELD NAME	MEANING	
type	IS41_MSG_SRV_REQ (0xc7b0)	
id	Dialogue_ID	
src	Sending module_id	
dst	IS41_TASK_ID	
rsp_req	0	
class	0	
status	0	
err_info	0	
len	Number of bytes of user data	
PARAMETER AREA		
OFFSET	SIZE	NAME
0	1	Primitive type octet.
1	len – 2	Parameters in Name-Length-Data format.
Len – 1	1	Set to zero indicating end of message.

Description:

The User application uses this message to send service specific request and response primitives to the IS-41 module. These primitives correspond directly to the operations specified in IS-41.

Once a dialogue has been opened with a peer system, the User application may send service specific primitives to the IS-41 module. IS-41 buffers the service primitives until a dialogue primitive requesting the transfer of the service requests to the peer is issued by the User application.

The buffered services are transferred as IS41 operations, coded into the component of the appropriate TCAP package.

The **IS41-CLOSE** primitive transfers the buffered services and closes the dialogue. The **IS41-DELIMITER** primitive transfers the buffered service requests and the dialogue remains open.

Note that although many service requests may be transferred using one **IS41-CLOSE** or **IS41-DELIMITER** dialogue primitive, only one service primitive may be present in each **IS41_MSG_SRV_REQ** message.

All service specific request primitives must contain the dialogue ID of the dialogue to which they belong. This is encoded in the message header and does not form part of the parameter area.

Parameter area contents:

The specific service primitive type octet is coded as defined in 8.4 Service Primitive Types.

The coding of service primitive parameters is given in 8.10 IS-41 Service Primitive Parameters

The following tables show the parameters associated with each service request primitive and shows whether the parameter is

- M - MANDATORY in which case the message will be discarded if the parameter is omitted.
- C - CONDITIONAL in which case the parameter is mandatory in some circumstances only. For example a result service may include a user error parameter or data parameters but not both.
- O - OPTIONAL in which case the parameter is not essential.

The “ACK” table for each service indicates parameters for the Response primitive.

IS41-SMS-DELIVERY-BACKWARD-REQ	
Parameter	Class
Primitive type octet	M
Invoke ID	M
InterMSCCircuitID	M
MobileIdentificationNumber	M
SMS_BearerData	M
SMS_TeleserviceIdentifier	M
ElectronicSerialNumber	O
SMS_ChargeIndicator	O
SMS_DestinationAddress	O
SMS_OriginalDestinationAddress	O
SMS_OriginalDestinationSubaddress	O
SMS_OriginalOriginationAddress	O
SMS_OriginalOriginationSubaddress	O
SMS_OriginationAddress	O
Timeout	O

IS41-SMS-DELIVERY-BACKWARD-ACK	
Parameter	Class
Primitive type octet	M
Invoke ID	M
SMS_BearerData	O
SMS_CauseCode	O
SMS_FaultyParameter	O
User error	C
Provider error (CNF primitive only)	C

IS41-SMS-DELIVERY-FORWARD-REQ	
Parameter	Class
Primitive type octet	M
Invoke ID	M
InterMSCCircuitID	M
MobileIdentificationNumber	M
SMS_BearerData	M
SMS_TeleserviceIdentifier	M
ElectronicSerialNumber	O
SMS_ChargeIndicator	O
SMS_DestinationAddress	O
SMS_OriginalDestinationAddress	O
SMS_OriginalDestinationSubaddress	O
SMS_OriginalOriginationAddress	O
SMS_OriginalOriginationSubaddress	O
SMS_OriginationAddress	O
Timeout	O

IS41-SMS-DELIVERY-FORWARD-ACK	
Parameter	Class
Primitive type octet	M
Invoke ID	M
SMS_BearerData	O
SMS_CauseCode	O
SMS_FaultyParameter	O
User error	C
Provider error (CNF primitive only)	C

IS41-SMS-DELIVERY-POINT-TO-POINT-REQ	
Parameter	Class
Primitive type octet	M
Invoke ID	M
SMS_BearerData	M
SMS_TeleserviceNameIdentifier	M
ElectronicSerialNumber	O
MobileIdentificationNumber	O
SMS_ChargeIndicator	O
SMS_DestinationAddress	O
SMS_MessageCount	O
SMS_NotificationIndicator	O
SMS_OriginalDestinationAddress	O
SMS_OriginalDestinationSubaddress	O
SMS_OriginalOriginationAddress	O
SMS_OriginalOriginationSubaddress	O
SMS_OriginationAddress	O
Timeout	O

IS41-SMS-DELIVERY-POINT-TO-POINT-ACK	
Parameter	Class
Primitive type octet	M
Invoke ID	M
SMS_BearerData	O
SMS_CauseCode	O
SMS_FaultyParameter	O
User error	C
Provider error (CNF primitive only)	C

IS41-SMS-NOTIFICATION-REQ	
Parameter	Class
Primitive type octet	M
Invoke ID	M
ElectronicSerialNumber	O
MobileIdentificationNumber	O
SMS_AccessDeniedReason	O
SMS_Address	O
Timeout	O

IS41-SMS-NOTIFICATION-ACK	
Parameter	Class
Primitive type octet	M
Invoke ID	M
User error	O
Provider error (CNF primitive only)	O

IS41-SMS-REQUEST-REQ	
Parameter	Class
Primitive type octet	M
Invoke ID	M
MobileIdentificationNumber	M
ElectronicSerialNumber	O
SMS_NotificationIndicator	O
SMS_TeleserviceIdentifier	O
Timeout	O

IS41-SMS-REQUEST-ACK	
Parameter	Class
Primitive type octet	M
Invoke ID	M
ElectronicSerialNumber	O
SMS_AccessDeniedReason	O
SMS_Address	O
User error	C
Provider error (CNF primitive only)	C

8.9 IS-41 Service Indication

Synopsis:

Protocol message sent from the IS-41 module to the User application containing a service specific primitive.

Message Format:

MESSAGE HEADER		
FIELD NAME	MEANING	
type	IS41_MSG_SRV_IND (0x87b1)	
id	Dialogue_ID	
src	IS41_TASK_ID	
dst	Sending module_id	
rsp_req	0	
hclass	0	
status	0	
err_info	0	
len	Number of bytes of user data	
PARAMETER AREA		
OFFSET	SIZE	NAME
0	1	Primitive type octet.
1	len - 2	Parameters in Name-Length-Data format.
len - 1	1	Set to zero indicating end of message.

Description:

The IS-41 module uses this message to send service specific indication and confirmation primitives to the User application. The primitives that may be sent are detailed in section 8.4 Service Primitive Types.

All specific service indication primitives contain the dialogue ID of the dialogue to which they belong. It is encoded in the message header and does not form part of the parameter area.

Parameter area contents:

The parameter area is coded as defined for the IS41-SERVICE-REQUEST message.

The coding of service primitive parameters is given in section 8.10 IS-41 Service Primitive Parameters

The parameters included in each primitive are as defined for the IS41-SERVICE-REQUEST message. The parameters for the Indication primitive are exactly the same as the Request primitive. The parameters for the Confirmation primitive are the same as the Response primitive with the addition of the optional "Provider error" parameter.

8.10 IS-41 Service Primitive Parameters

The following parameter names are defined for use in service primitive messages:

Parameter	Mnemonic	Value	
		(Dec)	(Hex)
Invoke ID	IS41PN_invoke_id	14	0x0e
Linked ID	IS41PN_linked_id	15	0x0f
Timeout	IS41PN_timeout	16	0x10
InterMSCCircuitID	IS41PN_interMSCcctId	17	0x11
MobileIdentification Number	IS41PN_mobileIdNum	18	0x12
SMS_BearerData	IS41PN_SMSbearerdata	19	0x13
SMS_Teleservice Identifier	IS41PN_SMStelesrvId	20	0x14
ElectronicSerialNumber	IS41PN_esn	21	0x15
SMS_ChargeIndicator	IS41PN_SMSchargeInd	22	0x16
User error	IS41PN_user_err	23	0x17
Provider error	IS41PN_prov_err	24	0x18
SMS_DestinationAddress	IS41PN_SMSdestAddr	25	0x19
SMS_OriginalDestination Address	IS41PN_SMSorigDestAddr	26	0x1a
SMS_OriginalDestination Subaddress	IS41PN_SMSorigDestSubAddr	27	0x1b
SMS_OriginationAddress	IS41PN_SMSorigAddr	28	0x1c
SMS_OriginalOriginationAddress	IS41PN_SMSorigOrigAddr	29	0x1d
SMS_OriginalOriginationSubaddress	IS41PN_SMSorigOrigSubAddr	30	0x1e
SMS_CauseCode	IS41PN_SMScauseCode	31	0x1f
SMS_MessageCount	IS41PN_SMSmsgCount	32	0x20
SMS_NotificationIndicator	IS41PN_SMSnotelnd	33	0x21
SMS_AccessDenied Reason	IS41PN_SMSaccessDeniedRsn	34	0x22
SMS_Address	IS41PN_SMSaddress	35	0x23
SMS_FaultyParameter	IS41PN_faultyParam	36	0x24

Service specific parameters are transferred transparently by the IS-41 module.

Some service specific parameters are specified with a fixed length. If received parameters are not the specified length an error component is returned.

Some service specific parameters have a specified length but additional received octets are specified to be ignored. Additional received octets are ignored and the parameter is handled normally.

The User application is sent a notice indication primitive if a service request containing an error is received from the network. The user is sent an open confirm primitive with a user error parameter if a result containing an error is received.

The coding for each parameter type is given in the following tables:

Parameter name	IS41PN_invoke_id
Parameter length	Fixed, set to 1
Parameter data	Single octet providing a unique invocation ID for each service request. Encoded in the range -128 to +127

Parameter name	IS41PN_linked_id
Parameter length	Fixed, set to 1
Parameter data	Provides the invoke ID of the service to which this service request indication is linked. Used only where an IS41 operation is specified to permit another operation invocation as a response. Encoded in the range -128 to +127

Parameter name	IS41PN_timeout
Parameter length	Fixed, set to 2
Parameter data	Used in service request primitives only. Permits the user to indicate the TCAP timeout period associated with this service. The IS41 module will provide a default timeout if this parameter is not included. Encoded in the range 0 to 409 seconds.

Parameter name	IS41PN_interMSCcctId
Parameter length	Fixed, set to 2
Parameter data	Two octets encoded as specified in TIA/EIA-41.5-D, i.e.: Octet 1 - Trunk Group Member Octet 2 - Trunk Member Number

Parameter name	IS41PN_mobileIdNum
Parameter length	Fixed, set to 5
Parameter data	Five octets encoded as specified in TIA/EIA-41.5-D.

Parameter name	IS41PN_SMSbearerdata
Parameter length	Variable, in the range 1 to 200
Parameter data	Encoded as specified in TIA/EIA-41.5-D. ie User defined.

Parameter name	IS41PN_SMStelesrvId
Parameter length	Fixed, set to 2 (additional octets ignored).
Parameter data	Encoded as specified in TIA/EIA-41.5-D.

Parameter name	IS41PN_esn
Parameter length	Fixed, set to 4.
Parameter data	Encoded as specified in TIA/EIA-41.5-D.

Parameter name	IS41PN_SMSchargeInd
Parameter length	Fixed, set to 1 (additional octets ignored).
Parameter data	Encoded as specified in TIA/EIA-41.5-D.

Parameter name	IS41PN_user_err
Parameter length	Fixed, set to 1
Parameter data	<p>Encoded as specified in TIA/EIA-41.5-D., i.e.:</p> <ul style="list-style-type: none"> 129 - UnrecognisedMIN 130 - UnrecognisedESN 131 - MIN_HLRMismatch 132 - OperationSequenceProblem 133 - ResourceShortage 134 - OperationNotSupported 135 - TrunkUnavailable 136 - ParameterError 137 - SystemFailure 138 - UnrecognisedParameterValue 139 - FeatureInactive 140 - MissingParameter

Parameter name	IS41PN_prov_err
Parameter length	Fixed, set to 1
Parameter data	<p>Single octet coded as follows:</p> <ul style="list-style-type: none"> 1 - duplicated invoke ID 2 - not supported service 3 - mistyped parameter 4 - resource limitation 5 - initiating release 6 - unexpected response from peer 7 - service completion failure 8 - no response from peer 9 - invalid response received

Parameter name	IS41PN_SMSdestAddr
Parameter length	Variable, in the range 4 to 12
Parameter data	Encoded as specified in TIA/EIA-41.5-D

Parameter name	IS41PN_SMSorigDestAddr
Parameter length	Variable, in the range 4 to 133
Parameter data	Encoded as specified in TIA/EIA-41.5-D

Parameter name	IS41PN_SMSorigDestSubAddr
Parameter length	Variable.
Parameter data	Encoded as specified in TIA/EIA-41.5-D

Parameter name	IS41PN_SMSorigAddr
Parameter length	Variable, in the range 4 to 12
Parameter data	Encoded as specified in TIA/EIA-41.5-D

Parameter name	IS41PN_SMSorigOrigAddr
Parameter length	Variable, in the range 4 to 133
Parameter data	Encoded as specified in TIA/EIA-41.5-D

Parameter name	IS41PN_SMSorigOrigSubAddr
Parameter length	Variable.
Parameter data	Encoded as specified in TIA/EIA-41.5-D

Parameter name	IS41PN_SMScauseCode
Parameter length	Fixed, set to 1 (additional octets ignored).
Parameter data	Encoded as specified in TIA/EIA-41.5-D

Parameter name	IS41PN_SMSmsgCount
Parameter length	Fixed, set to 1 (additional octets ignored).
Parameter data	Encoded as specified in TIA/EIA-41.5-D.

Parameter name	IS41PN_SMSnotelnd
Parameter length	Fixed, set to 1 (additional octets ignored).
Parameter data	Encoded as specified in TIA/EIA-41.5-D.

Parameter name	IS41PN_SMSAccessDeniedRsn
Parameter length	Fixed, set to 1 (additional octets ignored).
Parameter data	Encoded as specified in TIA/EIA-41.5-D.

Parameter name	IS41PN_SMSAddress
Parameter length	Variable, in the range 4 to 12
Parameter data	Encoded as specified in TIA/EIA-41.5-D.

Parameter name	IS41PN_faultyParam
Parameter length	Variable in range 1 to 3
Parameter data	Encoded as specified in TIA/EIA-41.5-D.

9. NON-PRIMITIVE INTERFACE

In addition to the primitive interface for passing IS-41 protocol messages between the IS-41 module and the User application, the IS-41 module supports a non-primitive interface for implementation-specific functionality.

The non-primitive interface permits the configuration of the IS-41 module.

It also allows the IS-41 module to supply diagnostic information, protocol error events and software error events to the local system management and maintenance modules.

This section describes the formats of all the messages used in the non-primitive interface.

All messages handled by IS-41 may request a confirmation. This is done by setting the `rsp_req` field in the message header. See Appendix A of the Software Environment Programmers' Manual. The confirmation message consists of the received message echoed back to the sending module. The status field of the confirmation message header may take the following values.

Mnemonic	Value	Description
IS41E_OK	0	No error message received OK.
IS41E_BAD_ID	1	Inappropriate or invalid id in request message
IS41E_BAD_STATE	2	Inappropriate or unrecognised message type .
IS41E_BAD_MSG	5	Unsupported message received.
IS41E_BAD_PARAM	6	Invalid parameters contained in message.
IS41E_NO_RESOURCES	7	Insufficient internal message resources.

9.1 IS-41 Configuration Request

Synopsis:

Message used to configure the IS-41 module for operation.

Message Format:

MESSAGE HEADER		
FIELD NAME	MEANING	
type	IS41_MSG_CONFIG (0x77b4)	
id	0	
src	Sending module_id	
dst	IS41_TASK_ID	
rsp_req	Used to request a confirmation	
hclass	0	
status	0	
srr_info	0	
len	40	
PARAMETER AREA		
OFFSET	SIZE	NAME
0	1	user_id
1	1	TCAP_id
2	1	mngt_id
3	1	maint_id
4	1	trace_id
5	1	reserved
6	2	base_usr_ogdlg_id
8	2	base_usr_icdlg_id
10	2	base_tc_ogdlg_id
12	2	base_tc_icdlg_id
14	2	nog_dialogues
16	2	nic_dialogues
18	2	num_invokes
20	4	options
24	16	reserved

Description:

This message is used to configure the IS-41 module for operation. It should be the first message sent to the module (any messages received before a valid configuration message will be discarded). It should only be issued once.

The message parameters relate to the environment in which the IS-41 module is operating.

Confirmation Message:

The module sending the message can optionally request that a confirmation message be returned by the IS-41 module after the message has been processed. This is achieved by setting the **rsp_req** field in the message header. This will cause a confirmation message of the same format to be returned. The **status** field in this message is zero on success or an error code otherwise.

Parameter Description:

user_id

User application module ID.

TCAP_id

TCAP module ID.

mngt_id

Management module ID.

maint_id

Maintenance module ID.

trace_id

Trace module ID.

base_usr_ogdlg_id

The base user dialogue ID for user dialogue ids assigned by the user for outgoing dialogues. The subsequent (**nog_dialogues** - 1) dialogue IDs will also be available to the user. The user must ensure that all values used in the dialogue ID field of protocol messages relating to outgoing dialogues lie within this range. Separate ranges of user dialogue ids must be defined for incoming and outgoing dialogues.

base_usr_icdlg_id

The base user dialogue ID for user dialogue ids assigned by the IS41 module to incoming dialogues. The IS-41 module uses values in the range **base_icdlg_id** to (**base_icdlg_id** + **nic_dialogues** - 1) for this purpose. Separate ranges of user dialogue ids must be defined for incoming and outgoing dialogues.

base_tc_ogdlg_id

The base dialogue ID for outgoing dialogues between IS-41 and TCAP. The subsequent (**nog_dialogues** - 1) dialogue IDs will also be handled by the module. The IS-41 module allocates the dialogue ID for each outgoing TCAP dialogue. It uses values in the range **base_tc_ogdlg_id** to (**base_tc_ogdlg_id** + **nog_dialogues** - 1) for this purpose. Separate ranges of TCAP dialogue ids must be defined for incoming and outgoing dialogues.

base_tc_icdlg_id

The first dialogue ID for incoming dialogues between TCAP and IS-41. The subsequent (**nic_dialogues** - 1) dialogue IDs will also be handled by the module. TCAP allocates the dialogue ID for each incoming dialogue. It must use values in the range **base_tc_icdlg_id** (**base_tc_icdlg_id** + **nic_dialogues** - 1) for this purpose. Separate ranges of TCAP dialogue ids must be defined for incoming and outgoing dialogues.

nog_dialogues

The maximum number of simultaneous outgoing dialogues that the module is required to support. This value is compared with a compile time constant to ensure that the module has sufficient internal resources to handle the requested maximum number of outgoing dialogues.

nic_dialogues

The maximum number of simultaneous incoming dialogues that the module is required to support. This value is compared with a compile time constant to ensure that the module has sufficient internal resources to handle the requested maximum number of incoming dialogues.

num_invokes

The maximum number of simultaneous invocations that the module is required to support. This value is compared with a compile time constant to ensure that the module has sufficient internal resources to handle the requested number of simultaneous invocations.

options

Reserved for future use.

9.2 IS-41 Timer Configuration Request

Synopsis:

Message used to set up default protocol parameters for use by the IS-41 module.

Message Format:

MESSAGE HEADER		
FIELD NAME	MEANING	
type	IS41_MSG_CNF_TIM (0x77b5)	
id	0	
src	Sending module_id	
dst	IS41_TASK_ID	
rsp_req	Used to request a confirmation	
hclass	0	
status	0	
err_info	0	
len	3	
PARAMETER AREA		
OFFSET	SIZE	NAME
0	2	t0

Description:

This message is used to configure the IS-41 protocol timers for operation. It should be sent to the module before any protocol messages are sent or received. It should only be issued once.

Confirmation Message:

The module sending the message can optionally request that a confirmation message be returned by the IS-41 module after the message has been processed. This is achieved by setting the **rsp_req** field in the message header. This will cause a confirmation message of the same format to be returned. See appendix A1. The **status** field in this message is zero on success or an error code otherwise.

Parameter Description:

t0

Timer waiting for a response from the User application. Minimum value 5 seconds. (If the Configure Timers message is not sent, a default value of 5 seconds will be used.)

9.3 IS-41 Software Event Indication

Synopsis:

Message used by IS-41 to indicate an implementation specific software related event to the local management module.

Message Format:

MESSAGE HEADER	
FIELD NAME	MEANING
type	IS41_MSG_ERROR_IND (0x07b9)
id	See below
src	IS41_TASK_ID
dst	Management module id
rsp_req	Used to request a confirmation
hclass	0
status	Software event code (see below)
err_info	0
len	0

Software event code

The **Software event code** contained in the **status** field of the message indicates the type of event. Possible values are listed in the following table that also lists the meaning of the **id** field in each case.

Mnemonic	Code	Id	Description
IS41SWE_NO_MSSM	1	0	Maximum number of active invocations exceeded.
IS41SWE_NO_DLG	2	0	No internal resource to handle dialogue.
IS41SWE_NO_ISM	3	0	Internal pool of structured messages exhausted.
IS41SWE_ISM_LOW	4	0	Internal pool of structured messages running low.
IS41SWE_BAD_MSG	5	Msg_type	Unrecognised inter task message received.
IS41SWE_TX_FMT_ERR	6	0	Internal error during message formatting.
IS41SWE_USER_BAD_FMT	7	0	Badly formatted message received from User application.
IS41SWE_TCAP_BAD_FMT	8	0	Badly formatted IS-41 message received from TCAP.
IS41SWE_UNREC_TYPE	9	Msg_type	Unrecognised IS-41 message received from TCAP.
IS41SWE_INVALID_DLG_ID	10	0	Message received from TCAP with unrecognised dialogue id.

9.4 IS-41 Trace Mask Request

Synopsis:

Message used to configure IS-41 to send a trace message to the trace module whenever a specific message type is sent or received. The trace module is identified in the IS-41 configuration request message.

Message Format:

MESSAGE HEADER		
FIELD NAME	MEANING	
type	IS41_MSG_TRACE_MASK (0x57b6)	
id	0	
src	Sending module ID	
dst	IS41_TASK_ID	
rsp_req	Used to request a confirmation	
hclass	0	
status	0	
err_info	0	
len	12	
PARAMETER AREA		
OFFSET	SIZE	NAME
0	4	op_evt_mask - Output event trace mask
4	4	ip_evt_mask - Input event trace mask
8	4	non_prim_mask - Non-primitive trace mask

op_evt_mask

The output event trace mask. This is a 32-bit value with bits set to 1 to cause a trace message to be sent to the system trace module when IS-41 sends the associated protocol message.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	TCCMP	TCDLG	IS41SRV	IS41DLG
				_REQ	_REQ	_IND	_IND

IS41DLG_IND – Dialogue indication primitive from IS-41 to User application

IS41SRV_IND – Service indication primitive from IS-41 to User application

TCDLG_REQ – Dialogue request primitive from IS-41 to TCAP

TCCMP_REQ – Component request primitive from IS-41 to TCAP

ip_evt_mask

The input event trace mask. This is a 32-bit value with bits set to 1 to cause a trace message to be sent to the system trace module when IS-41 receives the associated protocol message.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	TCCMP	TCDLG	IS41SRV	IS41DLG
				_IND	_IND	_REQ	_REQ

IS41DLG_REQ – Dialogue request primitive from User application to IS41

IS41SRV_REQ – Service request primitive from User application to IS41

TCDLG_IND – Dialogue Indication primitive from TCAP to IS41

TCCMP_IND – Component Indication primitive from TCAP to IS41

non_prim_mask

The non-primitive trace mask. This is a 32-bit value with bits set to 1 to cause a trace message to be sent to the system trace module when IS-41 receives the associated non-primitive message.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	SSW MSK	SMT MSK	SW_ EVT	MT _EVT	CNF _TIM

CNF_TIM – Timer Configuration Message received by IS-41

MT_EVT – Maintenance Event indication message sent by IS-41 to the maintenance module.

SW_EVT– Software Error Event indication message sent by IS-41 to the management module.

SMT_MSK- Maintenance Mask Request message received by IS-41.

SSW_MSK- Error Mask Request message received by IS-41.

9.5 Trace Event Indication

Synopsis:

The IS-41 trace event masks are used to enable and disable tracing of all protocol and non primitive messages received or sent by the IS41 module. The traced messages are reported as event indications as shown below:

Message Format:

MESSAGE HEADER		
FIELD NAME	MEANING	
type	MGT_MSG_TRACE_EV (0x0003)	
id	0	
src	IS41_TASK_ID	
dst	Management module id	
rsp_req	0	
hclass	0	
status	0	
err_info	0	
len	18 + length of traced data	
PARAMETER AREA		
OFFSET	SIZE	NAME
0	1	source module id
1	1	destination module id
2	2	id
4	2	type
6	2	status
8	4	timestamp
12	4	pointer to the message being traced
16	2	data length
18	0 .. 280	data - Data taken from the contents of the MSG parameter area.

9.6 IS-41 Maintenance Mask Request

Synopsis:

Message used to configure which maintenance events IS-41 will send a maintenance event indication message for.

Message Format:

MESSAGE HEADER		
FIELD NAME	MEANING	
type	IS41_MSG_S_MAINT_MASK (0x57b8)	
id	0	
src	Sending module ID	
dst	IS41_TASK_ID	
rsp_req	Used to request a confirmation	
hclass	0	
status	0	
err_info	0	
len	9	
PARAMETER AREA		
OFFSET	SIZE	NAME
0	9	mask

mask

A bit mask indicating the maintenance events which are active. A '1' indicates that an event is active and a '0' indicates that is not active. The first octet sent is for maintenance event codes 0-7, (bit 0 for event code 0) the second octet for maintenance event codes 8-15, (bit 0 for event code 8) etc.

Pad unused bits with zeros.

No maintenance events are currently defined.

9.7 IS-41 Software Event Mask Request

Synopsis:

Message used to configure which software error events IS-41 will send a software event indication message for.

Message Format:

MESSAGE HEADER		
FIELD NAME	MEANING	
type	IS41_MSG_S_ERROR_MASK (0x57b7)	
id	0	
src	Sending module ID	
dst	IS41_TASK_ID	
rsp_req	Used to request a confirmation	
hclass	0	
status	0	
err_info	0	
len	9	
PARAMETER AREA		
OFFSET	SIZE	NAME
0	9	mask

mask

A bit mask indicating the software events which are active. A '1' indicates that an event is active and a '0' indicates that is not active. The first octet sent is for software event codes 0-7 (bit 0 for event code 0), the second octet for software event codes 8-15 (bit 0 for event code 8), etc.

Pad unused bits with zeros.

All software events currently reported in the IS41_MSG_ERROR_IND message detailed in section 9.3 IS-41 Software Event Indication are enabled by default.

APPENDIX 1

A1 Tick Timer message format

The IS-41 module requires a periodic “tick timer” message. This must be sent at the required frequency, nominally every tenth of a second.

The format of the required message is shown below:

MESSAGE HEADER		
FIELD NAME		MEANING
type		TM_EXP (0xc002)
id		Index to timer table
src		Sending module ID
dst		IS41_TASK_ID
rsp_req		0
class		0
status		0
err_info		0
len		4
PARAMETER AREA		
OFFSET	SIZE	NAME
0	4	Timer type

APPENDIX 2

A2 Message Type reference

The following table, lists by message type all the messages described in this manual.

Value	Mnemonic	Description
0xc7b2	IS41_MSG_DLG_REQ	Used to send a dialogue request from the user to the IS-41 module.
0x87b3	IS41_MSG_DLG_IND	Dialogue message indication sent from the IS-41 module to the user.
0xc7b0	IS41_MSG_SRV_REQ	Used to send a service request from the user to the IS-41 module.
0x87b1	IS41_MSG_SRV_IND	Service request message indication sent from the IS-41 module to the user.
0x77b4	IS41_MSG_CONFIG	Used to configure the IS-41 module
0x77b5	IS41_MSG_CNF_TIM	Configures timer values
0x07b9	IS41_MSG_ERROR_IND	Used to indicate a software related event to the local management module
0x57b6	IS41_MSG_TRACE_MASK	Used to configure module message tracing
0x57b8	IS41_MSG_S_MAINT_MASK	Used to configure maintenance event tracing
0x57b7	IS41_MSG_S_ERROR_MASK	Used to configure software event tracing

Note:

For more information on the MSG message structure see Appendix A of the Software Environment Programmers' Manual.