TEXAS INSTRUMENTS

Service Access Point

TCP/IP SAP

Department:	TI Berlin
Creation Date:	18 August 2003
Last Modified:	05 September 2003 by Juergen Nickelsen
ID and Version:	8462.000.03.005
Status:	Draft

0   Document Control

0.1   Document History

ID	Author	Date	Status
8462.000.03.001	NI	18 August 2003	Work in progress
Initial version.			
8462.000.03.002	NI	21 August 2003	Work in progress
First compilable version.			
8462.000.03.003	NI	26 August 2003	Work in progress
Result parameters for a few CNFs.			
8462.000.03.004	NI	01 September 2003	Draft
Renumbered primitives; introduced data and flow control primitives.			
8462.000.03.005	NI	05 September 2003	Draft
Changed task handle to communication handle.			

0.2   References, Abbreviations, Terms

[C_8462.601]   Socket API for GPF-based TCP/IP. Detailed specification. Texas Instruments, August 2003.

[TI 7010.801]   7010.801, References and Vocabulary, Texas Instruments

[RFC 1034]   P.V. Mockapetris: Domain names – concepts and facilities. Internet RFC 1034, November 1987.

Table of Contents

TEXAS INSTRUMENTS

1   Introduction

The Service Access Point (SAP) for an entity defines the interface supported by that entity in order to gain access to its services. This document contains a specification of the TCPIP SAP for the TCP/IP entity of the Texas Instruments GSM/GPRS protocol stack.

The TCPIP SAP is based on the exchange of primitives between the TCP/IP entity and the application entities on one hand, and between the TCP/IP entity and the ACI on the other hand (see Figure 1-1). The method used for exchange of primitives is message passing. The terms "message" and "primitive" are used interchangeably in this document.

This document has three primary sections. The first (section 2) contains a specification of constants used on the interface, which are not specific to the value of a particular parameter. Such constants are typically size fields, specifying array or element sizes.

The second primary section (section 3) contains the top-level description of the primitives defined for the interface. For each primitive its function is described, and a parameter list is given.

For any primitive using complex structures or parameters with identified values, references will be given to subsections of the third primary section (section 0). This section contains specifications of the complex types used in the primitives of the interface. It also contains specifications of parameters with predefined values or value ranges. Cross-references to subtypes may be used within this section for complex type declarations. Types, which are common for more SAP's or are defined as ASN1 types, are specified in common include-files, and references to these files will be given instead.

As the world of mobile communication evolves, some changes to this interface may be necessary. However, as this document is the basis for the TCPIP SAP it will be fully maintained and all necessary changes will result in updated versions of this document.

1.1   Interfaces of the TCP/IP Entity
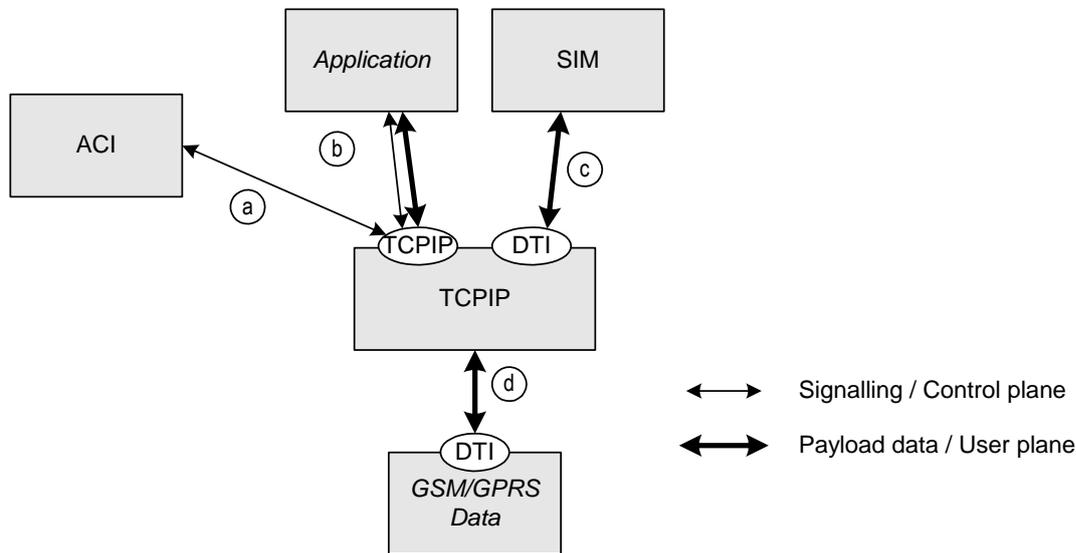
Figure 1-1: Interface of the TCP/IP entity

(a) The ACI controls the TCP/IP entity over the TCPIP SAP.

(b) Most applications do TCP/IP-related signalling and payload data exchange over the TCPIP SAP.

(c) The SIM entity exchanges payload data with the TCP/IP entity over the DTI SAP; the signalling is done by the ACI in this case..

(d) The TCP/IP entity exchanges IP packet data with the GSM/GPRS data entities over the DTI SAP.

1.2   Description of Parameters

The mapping of the primitives to the Socket API and a more detailed description of the primitive parameters can be found in the document "Socket API for GPF-based TCP/IP" [C_8462.601].

2   Constants

Definition:

Name	Value	Comment
TCPIP_HNAMELEN	256	Maximum length of the full-qualified domain name of an Internet host (as defined in [RFC 1034]) plus one byte for the zero termination.
TCPIP_DEFAULT_MTU_SIZE	1500	Default size of the maximum transfer unit of the network connection. To be used if no other information is available.
TCPIP_UNSPECIFIED_IPADDR	0	The unspecified IP address (0.0.0.0)
TCPIP_UNSPECIFIED_PORT	0	The unspecified port number.

History:

21 August 2003   NI   Initial

3   Primitives

3.1   TCPIP_INITIALIZE_REQ

Description:

The ACI requests initialization of TCPIP. TCPIP will initialize its internal data structures, including memory allocation.

Definition:

Short Name	ID	Direction
TCPIP_INITIALIZE_REQ	0x80000048	ACI → TCPIP

Elements:

Long Name	Short Name	Ref	Type

History:

21 August 2003   NI   Initial

3.2   TCPIP_INITIALIZE_CNF

Description:

Response to TCPIP_INITIALIZE_REQ.

Definition:

Short Name	ID	Direction
TCPIP_INITIALIZE_CNF	0x80004048	TCPIP → ACI

Elements:

Long Name	Short Name	Ref	Type
Result code	result	4.5	U8

History:

21 August 2003   NI   Initial

3.3   TCPIP_SHUTDOWN_REQ

Description:

The ACI requests shutdown of TCPIP. TCPIP will stop its operation and will free all allocated resources.

Definition:

Short Name	ID	Direction
TCPIP_SHUTDOWN_REQ	0x80010048	ACI → TCPIP

Elements:

Long Name	Short Name	Ref	Type

TEXAS INSTRUMENTS

History:

21 August 2003   NI   Initial

3.4   TCPIP_SHUTDOWN_CNF

Description:

Response to TCPIP_SHUTDOWN_REQ.

Definition:

Short Name	ID	Direction
TCPIP_SHUTDOWN_CNF	0x80014048	TCPIP → ACI

Elements:

Long Name	Short Name	Ref	Type
Result code	result	4.5	U8

History:

21 August 2003   NI   Initial

3.5   TCPIP_IFCONFIG_REQ

Description:

The ACI notifies TCPIP about a network interface (i. e. CSD connection or PDP context) coming up or going down.

Definition:

Short Name	ID	Direction
TCPIP_IFCONFIG_REQ	0x80020048	ACI → TCPIP

Elements:

Long Name	Short Name	Ref	Type
Direction of interface transition (coming up or going down)	if_up	4.12	U8
MTU size of interface (unused when interface goes down)	mtu_size	4.11	U16
IP address of interface (unused when interface goes down)	ipaddr	4.8	U32
First DNS address (unused when interface goes down)	dnsaddr1	4.13	U32
Second DNS address (unused when interface goes down)	dnsaddr2	4.14	U32

History:

21 August 2003   NI   Initial

3.6   TCPIP_IFCONFIG_CNF

Description:

Response to a TCPIP_IFCONFIG_REQ.

Definition:

Short Name	ID	Direction
TCPIP_IFCONFIG_CNF	0x80024048	TCPIP → ACI

Elements:

Long Name	Short Name	Ref	Type
Result code	result	4.5	U8

History:

21 August 2003   NI   Initial

3.7   TCPIP_DTI_REQ

Description:

The ACI requests TCPIP to establish a link with the specified neighbor entity. The neighbor entity can be a protocol stack entity or an application entity (SIM).

Definition:

Short Name	ID	Direction
TCPIP_DTI_REQ	0x80030048	ACI → TCPIP

Elements:

Long Name	Short Name	Ref	Type
DTI connect	dti_conn	4.18	U8
Neighbor entity name	entity_name	4.16	U32
DTI link identifier	link_id	4.15	U32
DTI link direction	dti_direction	4.17	U8

History:

21 August 2003   NI   Initial

3.8   TCPIP_DTI_CNF

Description:

The TCPIP confirms establishment of a link with a neighbor entity.

Definition:

Short Name	ID	Direction
TCPIP_DTI_CNF	0x80034048	ACI → TCPIP

Elements:

Long Name	Short Name	Ref	Type
DTI connect	dti_conn	4.18	U8
DTI link identifier	link_id	4.15	U32

History:

TEXAS INSTRUMENTS

21 August 2003   NI   Initial

3.9   TCPIP_CREATE_REQ

Description:

The application requests a newly created socket from TCPIP.

Definition:

Short Name	ID	Direction
TCPIP_CREATE_REQ	0x80040048	appl → TCPIP

Elements:

Long Name	Short Name	Ref	Type
Application communication handle	app_handle	4.1	U32
IP protocol type	ipproto	4.2	U8
Request ID (arbitrary value chosen by the application)	request_id	4.3	U32

History:

21 August 2003   NI   Initial
05 September 2003   NI   Changed task handle to communication handle

3.10   TCPIP_CREATE_CNF

Description:

The TCPIP entity returns a newly created socket or an error code to the application.

Definition:

Short Name	ID	Direction
TCPIP_CREATE_CNF	0x80044048	TCPIP → appl

Elements:

Long Name	Short Name	Ref	Type
Type of the event as passed to the application (TCPIP_CREATE_CNF)	event_type	4.4	U32
Result code	result	4.5	U8
Descriptor of new socket (if result is TCPIP_RESULT_OK)	socket	4.6	U32
Request ID as passed in TCPIP_CREATE_REQ	request_id	4.3	U32

History:

21 August 2003   NI   Initial

3.11   TCPIP_CLOSE_REQ

Description:

The application requests TCPIP to close a socket.

Definition:

![Texas Instruments logo]

Short Name	ID	Direction
TCPIP_CLOSE_REQ	0x80050048	app1 → TCPIP

Elements:

Long Name	Short Name	Ref	Type
Application communication handle	app_handle	4.1	U32
Socket descriptor	socket	4.6	U32

History:

21 August 2003	NI	Initial
05 September 2003	NI	Changed task handle to communication handle

3.12 TCPIP_CLOSE_CNF

Description:

TCPIP returns the result of a TCPIP_CLOSE_REQ.

Definition:

Short Name	ID	Direction
TCPIP_CLOSE_CNF	0x80054048	TCPIP → app1

Elements:

Long Name	Short Name	Ref	Type
Type of the event as passed to the application (TCPIP_CLOSE_CNF)	event_type	4.4	U32
Result code	result	4.5	U8
Socket descriptor	socket	4.6	U32

History:

21 August 2003	NI	Initial

3.13 TCPIP_BIND_REQ

Description:

The application requests TCPIP to bind a socket to the specified port.

Definition:

Short Name	ID	Direction
TCPIP_BIND_REQ	0x80060048	app1 → TCPIP

Elements:

Long Name	Short Name	Ref	Type
Application communication handle	app_handle	4.1	U32
Socket descriptor	socket	4.6	U32
Port number to bind the socket to	port	4.7	U16

History:

21 August 2003	NI	Initial
05 September 2003	NI	Changed task handle to communication handle

3.14 TCPIP_BIND_CNF

Description:

TCPIP returns the result of a TCPIP_BIND_REQ.

Definition:

Short Name	ID	Direction
TCPIP_BIND_CNF	0x80064048	TCPIP → appl

Elements:

Long Name	Short Name	Ref	Type
Type of the event as passed to the application (TCPIP_BIND_CNF)	event_type	4.4	U32
Result code	result	4.5	U8
Socket descriptor	socket	4.6	U32

History:

21 August 2003	NI	Initial

3.15 TCPIP_LISTEN_REQ

Description:

The application requests TCPIP to listen for incoming connections on a socket.

Definition:

Short Name	ID	Direction
TCPIP_LISTEN_REQ	0x80070048	appl → TCPIP

Elements:

Long Name	Short Name	Ref	Type
Application communication handle	app_handle	4.1	U32
Socket descriptor	socket	4.6	U32

History:

21 August 2003	NI	Initial
05 September 2003	NI	Changed task handle to communication handle

3.16 TCPIP_LISTEN_CNF

Description:

TCPIP returns the result of a TCPIP_LISTEN_REQ.

Definition:

Short Name	ID	Direction
TCPIP_LISTEN_CNF	0x80074048	TCPIP → appl

Elements:

Long Name	Short Name	Ref	Type
Type of the event as passed to the application (TCPIP_LISTEN_CNF)	event_type	4.4	U32
Result code	result	4.5	U8
Socket descriptor	socket	4.6	U32

History:

21 August 2003   NI   Initial

3.17 TCPIP_CONNECT_REQ

Description:

The application requests TCPIP to connect a socket to a remote peer.

Definition:

Short Name	ID	Direction
TCPIP_CONNECT_REQ	0x80080048	appl → TCPIP

Elements:

Long Name	Short Name	Ref	Type
Application communication handle	app_handle	4.1	U32
Socket descriptor	socket	4.6	U32
IP address of remote peer to connect to	ipaddr	4.8	U32
Port number of remote peer to connect to	port	4.7	U16

History:

21 August 2003   NI   Initial
05 September 2003   NI   Changed task handle to communication handle

3.18 TCPIP_CONNECT_CNF

Description:

TCPIP returns the result of a TCPIP_CLOSE_REQ.

Definition:

Short Name	ID	Direction
TCPIP_CONNECT_CNF	0x80084048	TCPIP → appl

Elements:

Long Name	Short Name	Ref	Type

Type of the event as passed to the application (TCPIP_CONNECT_CNF)	event_type	4.4	U32
Result code	result	4.5	U8
Socket descriptor	socket	4.6	U32

History:

21 August 2003   NI   Initial

3.19   TCPIP_DATA_REQ

Description:

The application requests TCPIP to send payload data on a socket. Destination IP address and destination port number must be zero for TCP sockets; they must also be zero for connected UDP sockets if the destination address and port number shall be used that the socket has been connected to.

Definition:

Short Name	ID	Direction
TCPIP_DATA_REQ	0x80090048	app1 → TCPIP

Elements:

Long Name	Short Name	Ref	Type
Application communication handle	app_handle	4.1	U32
Socket descriptor	socket	4.6	U32
Destination IP address	ipaddr	4.8	U32
Destination port number	port	4.7	U16
Data buffer length	buflen	4.21	U16
Payload data	data	4.19	U32

History:

01 September 2003   NI   Initial
05 September 2003   NI   Changed task handle to communication handle

3.20   TCPIP_DATA_CNF

Description:

TCPIP acknowledges the TCPIP_DATA_REQ and signals to the application how much data the application is allowed to send (window size). The application may now send data on this socket again; it may send as much data as the "window" parameter specifies, but at least one TCPIP_DATA_REQ.

Definition:

Short Name	ID	Direction
TCPIP_DATA_CNF	0x80094048	TCPIP → app1

Elements:

Long Name	Short Name	Ref	Type
Type of the event as passed to the application (TCPIP_DATA_CNF)	event_type	4.4	U32
Result code	result	4.5	U8
Socket descriptor	socket	4.6	U32
Flow control window size	window	4.20	U32

History:

01 September 2003   NI   Initial

3.21 TCPIP_DATA_IND

Description:

TCPIP sends payload data to the application for the specified socket. Destination IP address and destination port number are zero for connected sockets.

Definition:

Short Name	ID	Direction
TCPIP_DATA_IND	0x800a4048	TCPIP → appl

Elements:

Long Name	Short Name	Ref	Type
Type of the event as passed to the application (TCPIP_DATA_IND)	event_type	4.4	U32
Result code	result	4.5	U8
Socket descriptor	socket	4.6	U32
Source IP address	ipaddr	4.8	U32
Source port number	port	4.7	U16
Data buffer length	buflen	4.21	U16
Payload data	data	4.19	U32

History:

01 September 2003   NI   Initial

3.22 TCPIP_DATA_RES

Description:

The application acknowledges a TCPIP_DATA_IND and signals to TCPIP how much data TCPIP is allowed to send (window size). TCPIP may now send data on this socket again; it may send as much data as the "window" parameter specifies, but at least one TCPIP_DATA_IND.

Definition:

Short Name	ID	Direction
TCPIP_DATA_RES	0x800a0048	appl → TCPIP

Elements:

Long Name	Short Name	Ref	Type
Application communication handle	app_handle	4.1	U32
Socket descriptor	socket	4.6	U32
Flow control window size	window	4.20	U32

History:

01 September 2003	NI	Initial
05 September 2003	NI	Changed task handle to communication handle

3.23 TCPIP_SOCKNAME_REQ

Description:

The application requests TCPIP to retrieve information about the local socket.

Definition:

Short Name	ID	Direction
TCPIP_SOCKNAME_REQ	0x800b0048	appl → TCPIP

Elements:

Long Name	Short Name	Ref	Type
Application communication handle	app_handle	4.1	U32
Socket descriptor	socket	4.6	U32

History:

21 August 2003	NI	Initial
05 September 2003	NI	Changed task handle to communication handle

3.24 TCPIP_SOCKNAME_CNF

Description:

TCPIP returns the result of a TCPIP_SOCKNAME_REQ.

Definition:

Short Name	ID	Direction
TCPIP_SOCKNAME_CNF	0x800b4048	TCPIP → appl

Elements:

Long Name	Short Name	Ref	Type
Type of the event as passed to the application (TCPIP_SOCKNAME_CNF)	event_type	4.4	U32
Result code	result	4.5	U8
Socket descriptor	socket	4.6	U32
IP address of socket	ipaddr	4.8	U32
Port number of socket	port	4.7	U16

History:

21 August 2003   NI   Initial

3.25   TCPIP_PEERNAME_REQ

Description:

The application requests TCPIP to retrieve information about the peer of a connected socket.

Definition:

Short Name	ID	Direction
TCPIP_PEERNAME_REQ	0x800c0048	appl → TCPIP

Elements:

Long Name	Short Name	Ref	Type
Application communication handle	app_handle	4.1	U32
Socket descriptor	socket	4.6	U32

History:

21 August 2003   NI   Initial
05 September 2003   NI   Changed task handle to communication handle

3.26   TCPIP_PEERNAME_CNF

Description:

TCPIP returns the result of a TCPIP_PEERNAME_REQ.

Definition:

Short Name	ID	Direction
TCPIP_PEERNAME_CNF	0x800c4048	TCPIP → appl

Elements:

Long Name	Short Name	Ref	Type
Type of the event as passed to the application (TCPIP_PEERNAME_CNF)	event_type	4.4	U32
Result code	result	4.5	U8
Socket descriptor	socket	4.6	U32
IP address of remote peer	ipaddr	4.8	U32
Port number of remote peer	port	4.7	U16

History:

21 August 2003   NI   Initial

3.27   TCPIP_HOSTINFO_REQ

Description:

The application requests TCPIP to retrieve DNS information a host or IP address. Of IP address and hostname, exactly one must be specified; if the IP address is specified, the DNS is queried for the hostname, and if the hostname is specified, the DNS is queried for the IP address.

Definition:

Short Name		ID	Direction
TCPIP_HOSTINFO_REQ		0x800d0048	app1 → TCPIP

Elements:

Long Name	Short Name	Ctrl	Ref	Type
Application communication handle	app_handle		4.1	U32
IP address of the host	ipaddr		4.8	U32
Full-qualified domain name of the host	hostname	[TCPIP_HNAMELEN]	4.9	UBYTE
Request ID (arbitrary value chosen by the application)	request_id		4.3	U32

History:

21 August 2003	NI	Initial
05 September 2003	NI	Changed task handle to communication handle

3.28   TCPIP_HOSTINFO_CNF

Description:

TCPIP returns the result of a TCPIP_HOSTINFO_REQ.

Definition:

Short Name		ID	Direction
TCPIP_HOSTINFO_CNF		0x800d4048	TCPIP → app1

Elements:

Long Name	Short Name	Ctrl	Ref	Type
Type of the event as passed to the application (TCPIP_SOCKNAME_CNF)	event_type		4.4	U32
Result code	result		4.5	U8
Socket descriptor (unused)	socket		4.6	U32
Full-qualified domain name of the host	hostname	[TCPIP_HNAMELEN]	4.9	UBYTE
IP address of the host	ipaddr		4.8	U32
Request ID as passed by in TCPIP_HOSTINFO_REQ	request_id		4.3	U32

History:

21 August 2003	NI	Initial

3.29   TCPIP_MTU_SIZE_REQ

Description:

The application requests TCPIP to indicate the MTU size of the network connection.

Definition:

Short Name	ID	Direction
TCPIP_MTU_SIZE_REQ	0x800e0048	TCPIP → appl

Elements:

Long Name	Short Name	Ref	Type
Application communication handle	app_handle	4.1	U32
Socket descriptor	socket	4.6	U32

History:

21 August 2003	NI	Initial
05 September 2003	NI	Changed task handle to communication handle

3.30 TCPIP_MTU_SIZE_CNF

Description:

TCPIP returns the result of a TCPIP_MTU_SIZE_REQ.

Definition:

Short Name	ID	Direction
TCPIP_MTU_SIZE_CNF	0x800e4048	TCPIP → appl

Elements:

Long Name	Short Name	Ref	Type
Type of the event as passed to the application (TCPIP_MTU_SIZE_CNF)	event_type	4.4	U32
Result code	result	4.5	U8
Socket descriptor	socket	4.6	U32
MTU size	mtu_size	4.11	U16

History:

21 August 2003	NI	Initial

3.31 TCPIP_CONNECT_IND

Description:

TCPIP indicates a connection by a remote peer.

Definition:

Short Name	ID	Direction
TCPIP_CONNECT_IND	0x800f4048	TCPIP → appl

Elements:

Long Name	Short Name	Ref	Type
Type of the event as passed to the application	event_type	4.4	U32

(TCPIP_CONNECT_IND)			
Result code (TCPIP_RESULT_OK)	result	4.5	U8
Socket on which the connection has been accepted	socket	4.6	U32
New socket allocated for the connection	new_socket	4.10	U32
IP address of the remote peer	ipaddr	4.8	U32
Port number on the remote side	port	4.7	U16

History:

21 August 2003   NI   Initial

3.32   TCPIP_CONN_CLOSED_IND

Description:

TCPIP indicates that a connection has been close by the remote peer.

Definition:

Short Name	ID	Direction
TCPIP_CONN_CLOSED_IND	0x80104048	TCPIP → app1

Elements:

Long Name	Short Name	Ref	Type
Type of the event as passed to the application (TCPIP_CONN_CLOSED_IND)	event_type	4.4	U32
Result code (TCPIP_RESULT_OK)	result	4.5	U8
Socket of the connection that has been closed	socket	4.6	U32

History:

21 August 2003   NI   Initial

3.33   TCPIP_ERROR_IND

Description:

TCPIP indicates an asychronous error on a socket connection.

Definition:

Short Name	ID	Direction
TCPIP_ERROR_IND	0x80114048	TCPIP → app1

Elements:

Long Name	Short Name	Ref	Type
Type of the event as passed to the application (TCPIP_ERROR_IND)	event_type	4.4	U32
Result code (one of TCPIP_RESULT_CONN_RESET, TCPIP_RESULT_CONN_ABORTED, or TCPIP_RESULT_UNREACHABLE)	result	4.5	U8

Socket for which the error occured	socket	4.6	U32

History:

21 August 2003   NI   Initial

3.34 TCPIP_INTERNAL_IND

Description:

TCPIP sends a message to itself..

Definition:

Short Name	ID	Direction
TCPIP_INTERNAL_IND	0x800f0048	TCPIP → TCPIP

Elements:

Long Name	Short Name	Ref	Type
Message pointer	msg_p	4.22	U32
Message ID	msg_id	4.23	U32

History:

16 October 2003   NI   Initial

4 Parameters

Parameters shall be part of the primitives described below and if applied the parameters shall contain the values specified here..

4.1 Application Communication Handle

Description:

> The application communication handle is used to identify a task to TCPIP. The value is passed to the task as the argument to pei_init().

Definition:

Type	Short Name	Comment
U32	app_handle	Application communication handle

History:

> 21 August 2003   NI   Initial
> 05 September 2003   NI   Changed task handle to communication handle

4.2 IP Protocol

Description:

> This parameter indicates which IP protocol (TCP or UDP) shall be used on the socket.

Definition:

Type	Short Name	Comment
U8	ipproto	IP protocol identifier

Values:

Value	C-Macro	Comment
6	TCPIP_IPPROTO_TCP	Value for TCP
17	TCPIP_IPPROTO_UDP	Value for UDP

History:

> 21 August 2003   NI   Initial

4.3 Request ID

Description:

> The request ID identifies associates a response to the corresponding request. The value is chosen by the application and passed through transparently by TCPIP. The application, or rather the Socket API, may choose whatever it sees fit – array indices, values of a counter, or pointers, for example.

Definition:

Type	Short Name	Comment
U32	request_id	Request ID

History:

> 21 August 2003   NI   Initial

4.4   Event Type

Description:

> This is the type of the event as passed to the application. It is not identical to the primitive type, as there are some events generated by the Socket API. The Socket API *could*, in theory, prepend the event type field itself, but that would involve a copy operation and, worse, the allocation of another memory block. In order to save that, the event type field is included in the primitive.

Definition:

Type	Short Name	Comment
U32	event_type	Type of the event

Values:

Value	C-Macro	Comment
1	TCPIP_EVT_CREATE_CNF	Result of TCPIP_CREATE_REQ
2	TCPIP_EVT_CLOSE_CNF	Result of TCPIP_CLOSE_REQ
3	TCPIP_EVT_BIND_CNF	Result of TCPIP_BIND_REQ
4	TCPIP_EVT_LISTEN_CNF	Result of TCPIP_LISTEN_REQ
5	TCPIP_EVT_CONNECT_CNF	Result of TCPIP_CONNECT_REQ
6	TCPIP_EVT_SOCKNAME_CNF	Result of TCPIP_SOCKNAME_REQ
7	TCPIP_EVT_PEERNAME_CNF	Result of TCPIP_PEERNAME_REQ
8	TCPIP_EVT_HOSTINFO_CNF	Result of TCPIP_HOSTINFO_REQ
9	TCPIP_EVT_MTU_SIZE_CNF	Result of TCPIP_MTU_SIZE_REQ
10	TCPIP_EVT_RECV_IND	Indication of incoming data
11	TCPIP_EVT_CONNECT_IND	Indication of incoming connection
12	TCPIP_EVT_CONN_CLOSED_IND	Indication of remotely closed connection
13	TCPIP_EVT_ERROR_IND	Indication of asynchronous error
14	TCPIP_EVT_FLOW_READY_IND	Flow Control: application can send again (not used by a TCPIP primitive!)

History:

> 21 August 2003   NI   Initial

4.5   Result Code

Description:

> The result code indicates if an operation was successful or not, and if not, what kind of error occured.

Definition:

Type	Short Name	Comment
U8	result	Result code

Values:

Value	C-Macro	Comment
0	TCPIP_RESULT_OK	Operation was successful.
1	TCPIP_RESULT_INVALID_PARAMETER	A parameter was invalid.
2	TCPIP_RESULT_INTERNAL_ERROR	An internal error has happened.
3	TCPIP_RESULT_ADDR_IN_USE	The address or port is already in use.
4	TCPIP_RESULT_OUT_OF_MEMORY	There is not enough memory to fulfill the request.
5	TCPIP_RESULT_NOT_SUPPORTED	The socket is not of a type that can support this operation.
6	TCPIP_RESULT_UNREACHABLE	The specified host cannot be reached.
7	TCPIP_RESULT_CONN_REFUSED	The connection to the specified address was refused by the remote host.
8	TCPIP_RESULT_TIMEOUT	The connection attempt timed out without establishing a connection.
9	TCPIP_RESULT_IS_CONNECTED	The request could not be fulfilled because the socket is already connected.
10	TCPIP_RESULT_HOST_NOT_FOUND	The specified host could not be found in the DNS.
11	TCPIP_RESULT_DNS_TEMP_ERROR	A temporary DNS error has occurred. Retrying the query may be successful.
12	TCPIP_RESULT_DNS_PERM_ERROR	A permanent DNS error has occurred.
13	TCPIP_RESULT_NO_IPADDR	The specified name has been found in the DNS, but no IP address is available.
14	TCPIP_RESULT_NOT_CONNECTED	The socket has not been connected yet.
15	TCPIP_RESULT_MSG_TOO_BIG	The size of the data buffer is too large for a UDP socket.
16	TCPIP_RESULT_CONN_RESET	The connection has been reset by the remote peer.
17	TCPIP_RESULT_CONN_ABORTED	The connection was aborted due to timeout or some other error condition.
18	TCPIP_RESULT_NO_BUFSPACE	Sending failed temporarily because the space to buffer the message was exhausted. The application should wait for a TCPIP_FLOW_READY_IND event to retry the operation.
19	TCPIP_RESULT_NETWORK_LOST	The operation failed because TCP/IP's network connection has been disconnected.
20	TCPIP_RESULT_NOT_READY	The operation failed because the TCP/IP entity was not yet initialised.

History:

21 August 2003   NI   Initial

4.6   Socket Descriptor

Description:

The identifier of a socket. The value is chosen by TCPIP when a socket is created. The application SHALL NOT use the value other than for comparison of equality.

Definition:

Type	Short Name	Comment
U32	socket	Socket descriptor

History:

    21 August 2003   NI   Initial

4.7 Port Number

Description:

    The number of a UDP or TCP port.

Definition:

Type	Short Name	Comment
U16	port	Port number

History:

    21 August 2003   NI   Initial

4.8 IP Address

Description:

    The Internet Protocol address of a host.

Definition:

Type	Short Name	Comment
U32	ipaddr	IP address

History:

    21 August 2003   NI   Initial

4.9 Host Name

Description:

    The full-qualified domain name of an Internet host.

Definition:

Type	Short Name	Ctrl	Comment
UBYTE	hostname	[TCPIP_HNAMELEN]	Name of the host

History:

    21 August 2003   NI   Initial

4.10 New Socket

Description:

    The identifier of a new socket return by a TCPIP_CONNECT_IND primitive. The value is chosen
    by TCPIP when a socket is returned in a TCPIP_CONNECT_IND. The application SHALL NOT

use the value other than for comparison of equality.

Definition:

Type	Short Name	Comment
U32	new_socket	Socket descriptor

History:

21 August 2003 NI Initial

4.11 MTU Size

Description:

The size of the Maximum Transfer Unit of a network connection.

Definition:

Type	Short Name	Comment
U16	mtu_size	MTU size

History:

21 August 2003 NI Initial

4.12 Interface Configuration Direction

Description:

This parameter indicates which IP protocol (TCP or UDP) shall be used on the socket.

Definition:

Type	Short Name	Comment
U8	if_up	Interface Configuration direction

Values:

Value	C-Macro	Comment
0	TCPIP_IFCONFIG_DOWN	Interface going down
1	TCPIP_IFCONFIG_UP	Interface coming up

History:

21 August 2003 NI Initial

4.13 First DNS IP Address

Description:

The IP address of the first DNS server.

Definition:

Type	Short Name	Comment
U32	dnsaddr1	IP address of first DNS server

History:

21 August 2003 NI Initial

4.14 Second DNS IP Address

Description:

The IP address of the second DNS server.

Definition:

Type	Short Name	Comment
U32	dnsaddr2	IP address of second DNS server

History:

21 August 2003   NI   Initial

4.15 DTI Link Identifier

Description:

This is a unique identifier used for the specific DTI link.. Used by the DTI library.

Definition:

Type	Short Name	Comment
U32	link_id	DTI link identifier

History:

21 August 2003   NI   Initial

4.16 DTI Neighbor Entity Name

Description:

The DTI neighbor entity as a string. Used by the DTI library.

Definition:

Type	Short Name	Comment
U32	entity_name	DTI neighbor entity name

History:

21 August 2003   NI   Initial

4.17 DTI Direction

Description:

Indicates the direction of the DTI link. Used by the DTI library.

Definition:

Type	Short Name	Comment
U8	dti_direction	Direction of DTI connection

Values:

Value	C-Macro	Comment
0	TCPIP_DTI_TO_HIGHER_LAYER	Connect to higher layer, i. e. an application

| 1 | TCPIP_DTI_TO_LOWER_LAYER | Connect to lower layer, i. e. to protocol stack |

History:

21 August 2003   NI   Initial

4.18   DTI Connect

Description:

Indicates whether the DTI link is to be established or disconnected.

Definition:

Type	Short Name	Comment
U8	dti_conn	Indicates to connect or disconnect DTI link

Values:

Value	C-Macro	Comment
0	TCPIP_CONNECT_DTI	Connect the DTI link
1	TCPIP_DISCONNECT_DTI	Disconnect the DTI link

History:

21 August 2003   NI   Initial

4.19   Payload data

Description:

Address of payload data (actually a pointer).

Definition:

Type	Short Name	Comment
U32	data	Address of payload data

History:

01 September 2003   NI   Initial

4.20   Flow control window size

Description:

Amount of data that may be sent over this socket.

Definition:

Type	Short Name	Comment
U32	window	Flow control window size

History:

01 September 2003   NI   Initial

4.21 Data buffer length

Description:

Amount of data in a payload data buffer.

Definition:

Type	Short Name	Comment
U16	buflen	Data buffer length

History:

01 September 2003   NI   Initial

4.22 Message pointer

Description:

Pointer to message contents.

Definition:

Type	Short Name	Comment
U32	msg_p	Message pointer

History:

16 October 2003   NI   Initial

4.23 Message ID

Description:

Identification of message.

Definition:

Type	Short Name	Comment
U32	msg_id	Message ID

History:

16 October 2003   NI   Initial