

NetX Services

© 2006 by Express Logic, Inc.

Address Resolution Protocol (ARP)

UINT	nx_arp_dynamic_entries_invalidate (NX_IP *ip_ptr);
UINT	nx_arp_dynamic_entry_set (NX_IP *ip_ptr, ULONG ip_address, ULONG physical_msw, ULONG physical_lsw);
UINT	nx_arp_enable (NX_IP *ip_ptr, VOID *arp_cache_memory, ULONG arp_cache_size);
UINT	nx_arp_gratuitous_send (NX_IP *ip_ptr, VOID (*response_handler)(NX_IP *ip_ptr, NX_PACKET *packet_ptr));
UINT	nx_arp_hardware_address_find (NX_IP *ip_ptr, ULONG ip_address, ULONG *physical_msw, ULONG *physical_lsw);
UINT	nx_arp_info_get (NX_IP *ip_ptr, ULONG *arp_requests_sent, ULONG *arp_requests_received, ULONG *arp_responses_sent, ULONG *arp_responses_received, ULONG *arp_dynamic_entries, ULONG *arp_static_entries, ULONG *arp_aged_entries, ULONG *arp_invalid_messages);
UINT	nx_arp_ip_address_find (NX_IP *ip_ptr, ULONG *ip_address, ULONG physical_msw, ULONG physical_lsw);
UINT	nx_arp_static_entries_delete (NX_IP *ip_ptr);
UINT	nx_arp_static_entry_create (NX_IP *ip_ptr, ULONG ip_address, ULONG physical_msw, ULONG physical_lsw);
UINT	nx_arp_static_entry_delete (NX_IP *ip_ptr, ULONG ip_address, ULONG physical_msw, ULONG physical_lsw);

Internet Control Message Protocol (ICMP)

UINT	nx_icmp_enable (NX_IP *ip_ptr);
UINT	nx_icmp_info_get (NX_IP *ip_ptr, ULONG *pings_sent, ULONG *ping_timeouts, ULONG *ping_threads_suspended, ULONG *ping_responses_received, ULONG *icmp_checksum_errors, ULONG *icmp_unhandled_messages);
UINT	nx_icmp_ping (NX_IP *ip_ptr, ULONG ip_address, CHAR *data, ULONG data_size, NX_PACKET **response_ptr, ULONG wait_option);

Internet Group Management Protocol (IGMP)

UINT	nx_igmp_enable (NX_IP *ip_ptr);
UINT	nx_igmp_info_get (NX_IP *ip_ptr, ULONG *igmp_reports_sent, ULONG *igmp_queries_received, ULONG *igmp_checksum_errors, ULONG *current_groups_joined);
UINT	nx_igmp_loopback_disable (NX_IP *ip_ptr);
UINT	nx_igmp_loopback_enable (NX_IP *ip_ptr);
UINT	nx_igmp_multicast_join (NX_IP *ip_ptr, ULONG group_address);
UINT	nx_igmp_multicast_leave (NX_IP *ip_ptr, ULONG group_address);

Internet Protocol (IP)

UINT	nx_ip_address_change_notify (NX_IP *ip_ptr, VOID (*ip_address_change_notify)(NX_IP *, VOID *), VOID *additional_info);
UINT	nx_ip_address_get (NX_IP *ip_ptr, ULONG *ip_address, ULONG *network_mask);
UINT	nx_ip_address_set (NX_IP *ip_ptr, ULONG ip_address, ULONG network_mask);
UINT	nx_ip_create (NX_IP *ip_ptr, CHAR *name, ULONG ip_address, ULONG network_mask, NX_PACKET_POOL *default_pool, VOID (*ip_link_driver)(NX_IP_DRIVER *), VOID *memory_ptr, ULONG memory_size, UINT priority);
UINT	nx_ip_delete (NX_IP *ip_ptr);
UINT	nx_ip_driver_direct_command (NX_IP *ip_ptr, UINT command, ULONG *return_value_ptr);
UINT	nx_ip_forwarding_disable (NX_IP *ip_ptr);
UINT	nx_ip_forwarding_enable (NX_IP *ip_ptr);
UINT	nx_ip_fragment_disable (NX_IP *ip_ptr);
UINT	nx_ip_fragment_enable (NX_IP *ip_ptr);
UINT	nx_ip_gateway_address_set (NX_IP *ip_ptr, ULONG ip_address);

UINT	nx_ip_info_get (NX_IP *ip_ptr, ULONG *ip_total_packets_sent, ULONG *ip_total_bytes_sent, ULONG *ip_total_packets_received, ULONG *ip_total_bytes_received, ULONG *ip_invalid_packets, ULONG *ip_receive_packets_dropped, ULONG *ip_receive_checksum_errors, ULONG *ip_send_packets_dropped, ULONG *ip_total_fragments_sent, ULONG *ip_total_fragments_received);
UINT	nx_ip_mtu_get (NX_IP *ip_ptr);
UINT	nx_ip_raw_packet_disable (NX_IP *ip_ptr);
UINT	nx_ip_raw_packet_enable (NX_IP *ip_ptr);
UINT	nx_ip_raw_packet_receive (NX_IP *ip_ptr, NX_PACKET **packet_ptr, ULONG wait_option);
UINT	nx_ip_raw_packet_send (NX_IP *ip_ptr, NX_PACKET *packet_ptr, ULONG destination_ip, ULONG type_of_service);
UINT	nx_ip_status_check (NX_IP *ip_ptr, ULONG needed_status, ULONG *actual_status, ULONG wait_option);

Packet Management

UINT	nx_packet_allocate (NX_PACKET_POOL *pool_ptr, NX_PACKET **packet_ptr, ULONG packet_type, ULONG wait_option);
UINT	nx_packet_copy (NX_PACKET *packet_ptr, NX_PACKET **new_packet_ptr, NX_PACKET_POOL *pool_ptr, ULONG wait_option);
UINT	nx_packet_data_append (NX_PACKET *packet_ptr, VOID *data_start, ULONG data_size, NX_PACKET_POOL *pool_ptr, ULONG wait_option);
UINT	nx_packet_data_retrieve (NX_PACKET *packet_ptr, VOID *buffer_start, ULONG *bytes_copied);
UINT	nx_packet_length_get (NX_PACKET *packet_ptr, ULONG *length);
UINT	nx_packet_pool_create (NX_PACKET_POOL *pool_ptr, CHAR *name, ULONG payload_size, VOID *memory_ptr, ULONG memory_size);
UINT	nx_packet_pool_delete (NX_PACKET_POOL *pool_ptr);
UINT	nx_packet_pool_info_get (NX_PACKET_POOL *pool_ptr, ULONG *total_packets, ULONG *free_packets, ULONG *empty_pool_requests, ULONG *empty_pool_suspensions, ULONG *invalid_packet_releases);
UINT	nx_packet_release (NX_PACKET *packet_ptr);
UINT	nx_packet_transmit_release (NX_PACKET *packet_ptr);

Reverse Address Resolution Protocol (RARP)

UINT	nx_rarp_disable (NX_IP *ip_ptr);
UINT	nx_rarp_enable (NX_IP *ip_ptr);
UINT	nx_rarp_info_get (NX_IP *ip_ptr, ULONG *rarp_requests_sent, ULONG *rarp_responses_received, ULONG *rarp_invalid_messages);

System Initialization

VOID	nx_system_initialize (VOID);
------	-------------------------------------

Transmission Control Protocol (TCP)

UINT	nx_tcp_client_socket_bind (NX_TCP_SOCKET *socket_ptr, UINT port, ULONG wait_option);
UINT	nx_tcp_client_socket_connect (NX_TCP_SOCKET *socket_ptr, ULONG server_ip, UINT server_port, ULONG wait_option);
UINT	nx_tcp_client_socket_port_get (NX_TCP_SOCKET *socket_ptr, UINT *port_ptr);
UINT	nx_tcp_client_socket_unbind (NX_TCP_SOCKET *socket_ptr);
UINT	nx_tcp_enable (NX_IP *ip_ptr);
UINT	nx_tcp_free_port_find (NX_IP *ip_ptr, UINT port, UINT *free_port_ptr);
UINT	nx_tcp_info_get (NX_IP *ip_ptr, ULONG *tcp_packets_sent, ULONG *tcp_bytes_sent, ULONG *tcp_packets_received, ULONG *tcp_bytes_received, ULONG *tcp_invalid_packets, ULONG *tcp_receive_packets_dropped, ULONG *tcp_checksum_errors, ULONG *tcp_connections, ULONG *tcp_disconnections, ULONG *tcp_connections_dropped, ULONG *tcp_retransmit_packets);
UINT	nx_tcp_server_socket_accept (NX_TCP_SOCKET *socket_ptr, ULONG wait_option);

UINT	nx_tcp_server_socket_listen (NX_IP *ip_ptr, UINT port, NX_TCP_SOCKET *socket_ptr, UINT listen_queue_size, VOID (*tcp_listen_callback)(NX_TCP_SOCKET *socket_ptr, UINT port));
UINT	nx_tcp_server_socket_relisten (NX_IP *ip_ptr, UINT port, NX_TCP_SOCKET *socket_ptr);
UINT	nx_tcp_server_socket_unaccept (NX_TCP_SOCKET *socket_ptr);
UINT	nx_tcp_server_socket_unlisten (NX_IP *ip_ptr, UINT port);
UINT	nx_tcp_socket_create (NX_IP *ip_ptr, NX_TCP_SOCKET *socket_ptr, CHAR *name, ULONG type_of_service, ULONG fragment, UINT time_to_live, ULONG window_size, VOID (*tcp_urgent_data_callback)(NX_TCP_SOCKET *socket_ptr), VOID (*tcp_disconnect_callback)(NX_TCP_SOCKET *socket_ptr));
UINT	nx_tcp_socket_delete (NX_TCP_SOCKET *socket_ptr);
UINT	nx_tcp_socket_disconnect (NX_TCP_SOCKET *socket_ptr, ULONG wait_option);
UINT	nx_tcp_socket_info_get (NX_TCP_SOCKET *socket_ptr, ULONG *tcp_packets_sent, ULONG *tcp_bytes_sent, ULONG *tcp_packets_received, ULONG *tcp_bytes_received, ULONG *tcp_retransmit_packets, ULONG *tcp_packets_queued, ULONG *tcp_checksum_errors, ULONG *tcp_socket_state, ULONG *tcp_transmit_queue_depth, ULONG *tcp_transmit_window, ULONG *tcp_receive_window);
UINT	nx_tcp_socket_mss_get (NX_TCP_SOCKET *socket_ptr, ULONG *mss);
UINT	nx_tcp_socket_mss_peer_get (NX_TCP_SOCKET *socket_ptr, ULONG *peer_mss);
UINT	nx_tcp_socket_mss_set (NX_TCP_SOCKET *socket_ptr, ULONG mss);
UINT	nx_tcp_socket_receive (NX_TCP_SOCKET *socket_ptr, NX_PACKET **packet_ptr, ULONG wait_option);
UINT	nx_tcp_socket_receive_notify (NX_TCP_SOCKET *socket_ptr, VOID (*tcp_receive_notify)(NX_TCP_SOCKET *socket_ptr));
UINT	nx_tcp_socket_send (NX_TCP_SOCKET *socket_ptr, NX_PACKET *packet_ptr, ULONG wait_option);
UINT	nx_tcp_socket_state_wait (NX_TCP_SOCKET *socket_ptr, UINT desired_state, ULONG wait_option);
UINT	nx_tcp_socket_transmit_configure (NX_TCP_SOCKET *socket_ptr, ULONG max_queue_depth, ULONG timeout, ULONG max_retries, ULONG timeout_shift);

User Datagram Protocol (UDP)

UINT	nx_udp_enable (NX_IP *ip_ptr);
UINT	nx_udp_free_port_find (NX_IP *ip_ptr, UINT port, UINT *free_port_ptr);
UINT	nx_udp_info_get (NX_IP *ip_ptr, ULONG *udp_packets_sent, ULONG *udp_bytes_sent, ULONG *udp_packets_received, ULONG *udp_bytes_received, ULONG *udp_invalid_packets, ULONG *udp_receive_packets_dropped, ULONG *udp_checksum_errors);
UINT	nx_udp_socket_bind (NX_UDP_SOCKET *socket_ptr, UINT port, ULONG wait_option);
UINT	nx_udp_socket_checksum_disable (NX_UDP_SOCKET *socket_ptr);
UINT	nx_udp_socket_checksum_enable (NX_UDP_SOCKET *socket_ptr);
UINT	nx_udp_socket_create (NX_IP *ip_ptr, NX_UDP_SOCKET *socket_ptr, CHAR *name, ULONG type_of_service, ULONG fragment, UINT time_to_live, ULONG queue_maximum);
UINT	nx_udp_socket_delete (NX_UDP_SOCKET *socket_ptr);
UINT	nx_udp_socket_info_get (NX_UDP_SOCKET *socket_ptr, ULONG *udp_packets_sent, ULONG *udp_bytes_sent, ULONG *udp_packets_received, ULONG *udp_bytes_received, ULONG *udp_packets_queued, ULONG *udp_receive_packets_dropped, ULONG *udp_checksum_errors);
UINT	nx_udp_socket_port_get (NX_UDP_SOCKET *socket_ptr, UINT *port_ptr);
UINT	nx_udp_socket_receive (NX_UDP_SOCKET *socket_ptr, NX_PACKET **packet_ptr, ULONG wait_option);
UINT	nx_udp_socket_receive_notify (NX_UDP_SOCKET *socket_ptr, VOID (*udp_receive_notify)(NX_UDP_SOCKET *socket_ptr));
UINT	nx_udp_socket_send (NX_UDP_SOCKET *socket_ptr, NX_PACKET *packet_ptr, ULONG ip_address, UINT port);
UINT	nx_udp_socket_unbind (NX_UDP_SOCKET *socket_ptr);
UINT	nx_udp_source_extract (NX_PACKET *packet_ptr, ULONG *ip_address, UINT *port);

Link Driver Constants

NX_LINK_PACKET_SEND	0
NX_LINK_INITIALIZE	1
NX_LINK_ENABLE	2
NX_LINK_DISABLE	3
NX_LINK_PACKET_BROADCAST	4
NX_LINK_ARP_SEND	5
NX_LINK_ARP_RESPONSE_SEND	6
NX_LINK_RARP_SEND	7
NX_LINK_MULTICAST_JOIN	8
NX_LINK_MULTICAST_LEAVE	9
NX_LINK_GET_STATUS	10
NX_LINK_GET_SPEED	11
NX_LINK_GET_DUPLEX_TYPE	12
NX_LINK_GET_ERROR_COUNT	13
NX_LINK_GET_RX_COUNT	14
NX_LINK_GET_TX_COUNT	15
NX_LINK_GET_ALLOC_ERRORS	16
NX_LINK_USER_COMMAND	50

NetX API Parameters

NX_NO_WAIT	0
NX_WAIT_FOREVER	0xFFFFFFFF
NX_TRUE	1
NX_FALSE	0
NX_NULL	0
NX_MAX_PORT	0xFFFF
NX_ANY_PORT	0
NX_IP_PACKET	36
NX_UDP_PACKET	44
NX_TCP_PACKET	56
NX_ICMP_PACKET	36
NX_IGMP_PACKET	36
NX_RECEIVE_PACKET	0
NX_IP_NORMAL	0x00000000UL
NX_IP_MIN_DELAY	0x00100000UL
NX_IP_MAX_DATA	0x00080000UL
NX_IP_MAX_RELIABLE	0x00040000UL
NX_IP_MIN_COST	0x00020000UL
NX_IP_TIME_TO_LIVE	0x00000800UL
NX_IP_INITIALIZE_DONE	0x0001
NX_IP_ADDRESS_RESOLVED	0x0002
NX_IP_LINK_ENABLED	0x0004
NX_IP_ARP_ENABLED	0x0008
NX_IP_UDP_ENABLED	0x0010
NX_IP_TCP_ENABLED	0x0020
NX_IP_IGMP_ENABLED	0x0040
NX_IP_RARP_COMPLETE	0x0080

NetX API Return Values

NX_SUCCESS	0x00
NX_NO_PACKET	0x01
NX_UNDERFLOW	0x02
NX_OVERFLOW	0x03
NX_NO_MAPPING	0x04
NX_DELETED	0x05
NX_POOL_ERROR	0x06
NX_PTR_ERROR	0x07

NX_WAIT_ERROR	0x08
NX_SIZE_ERROR	0x09
NX_OPTION_ERROR	0x0a
NX_DELETE_ERROR	0x10
NX_CALLER_ERROR	0x11
NX_INVALID_PACKET	0x12
NX_INVALID_SOCKET	0x13
NX_NOT_ENABLED	0x14
NX_ALREADY_ENABLED	0x15
NX_ENTRY_NOT_FOUND	0x16
NX_NO_MORE_ENTRIES	0x17
NX_ARP_TIMER_ERROR	0x18
NX_RESERVED_CODE0	0x19
NX_WAIT_ABORTED	0x1A
NX_IP_INTERNAL_ERROR	0x20
NX_IP_ADDRESS_ERROR	0x21
NX_ALREADY_BOUND	0x22
NX_PORT_UNAVAILABLE	0x23
NX_NOT_BOUND	0x24
NX_RESERVED_CODE1	0x25
NX_SOCKET_UNBOUND	0x26
NX_NOT_CREATED	0x27
NX_SOCKETS_BOUND	0x28
NX_NO_RESPONSE	0x29
NX_POOL_DELETED	0x30
NX_ALREADY_RELEASED	0x31
NX_RESERVED_CODE2	0x32
NX_MAX_LISTEN	0x33
NX_DUPLICATE_LISTEN	0x34
NX_NOT_CLOSED	0x35
NX_NOT_LISTEN_STATE	0x36
NX_IN_PROGRESS	0x37
NX_NOT_CONNECTED	0x38
NX_WINDOW_OVERFLOW	0x39
NX_ALREADY_SUSPENDED	0x40
NX_DISCONNECT_FAILED	0x41
NX_STILL_BOUND	0x42
NX_NOT_SUCCESSFUL	0x43
NX_UNHANDLED_COMMAND	0x44
NX_NO_FREE_PORTS	0x45
NX_INVALID_PORT	0x46
NX_INVALID_RELISTEN	0x47
NX_CONNECTION_PENDING	0x48
NX_TX_QUEUE_DEPTH	0x49
NX_NOT_IMPLEMENTED	0x80

NetX TCP State Machine Constants

NX_TCP_CLOSED	1
NX_TCP_LISTEN_STATE	2
NX_TCP_SYN_SENT	3
NX_TCP_SYN_RECEIVED	4
NX_TCP_ESTABLISHED	5
NX_TCP_CLOSE_WAIT	6
NX_TCP_FIN_WAIT_1	7
NX_TCP_FIN_WAIT_2	8
NX_TCP_CLOSING	9
NX_TCP_TIMED_WAIT	10
NX_TCP_LAST_ACK	11

ASCII Character Codes in HEX

		most significant nibble							
		0_	1_	2_	3_	4_	5_	6_	7_
0	NUL	DLE	SP	0	@	P	'	p	
1	SOH	DC1	!	1	A	Q	a	q	
2	STX	DC2	"	2	B	R	b	r	
3	ETX	DC3	#	3	C	S	c	s	
4	EOT	DC4	\$	4	D	T	d	t	
5	ENQ	NAK	%	5	E	U	e	u	
6	ACK	SYN	&	6	F	V	f	v	
7	BEL	ETB	'	7	G	W	g	w	
8	BS	CAN	(8	H	X	h	x	
9	HT	EM)	9	I	Y	i	y	
A	LF	SUB	*	:	J	Z	j	z	
B	VT	ESC	+	;	K	[K	}	
C	FF	FS	,	<	L	\	l	l	
D	CR	GS	-	=	M]	m	}	
E	SO	RS	.	>	N	^	n	~	
F	SI	US	/	?	O	_	o	DEL	

Power	Decimal Value	Hexadecimal Value	Power	Decimal Value	Hexadecimal Value
0	1	1	17	131,072	0x20000
1	2	2	18	262,144	0x40000
2	4	4	19	524,288	0x80000
3	8	8	20	1,048,576	0x100000
4	16	0x10	21	2,097,152	0x200000
5	32	0x20	22	4,194,304	0x400000
6	64	0x40	23	8,388,608	0x800000
7	128	0x80	24	16,777,216	0x1000000
8	256	0x100	25	33,554,432	0x2000000
9	512	0x200	26	67,108,864	0x4000000
10	1,024	0x400	27	134,217,728	0x8000000
11	2,048	0x800	28	268,435,456	0x10000000
12	4,096	0x1000	29	536,870,912	0x20000000
13	8,192	0x2000	30	1,073,741,824	0x40000000
14	16,384	0x4000	31	2,147,483,647	0x80000000
15	32,768	0x8000	32	4,294,967,295	0x100000000
16	65,536	0x10000			

ANSI C Format Specifications

%d or %i	decimal integer	%u	unsigned decimal integer
%ld or %li	long decimal integer	%lu	long unsigned decimal integer
%hd or %hi	short decimal integer	%hu	short unsigned decimal integer
%o	octal integer	%c	ASCII character
%lo	long octal integer	%s	ASCII string, null terminated
%ho	short octal integer	%f	floating point
%x or %X	hexadecimal integer	%e or %E	double precision floating point
%lx or %lX	long hexadecimal integer	%g or %G	double precision floating point
%hx or %hX	short hexadecimal integer	%p	void pointer

NETX

Programmers Guide

expresslogic