

## Appendix A: PICA data structures

Name	Windows definition	Linux definition	Description	reference in manual	Reference in PICA code
DEVLIST	<pre>typedef struct _DEVLIST {     int num_devices;     char     dev_names[MAXDEVS][MAXDEVSIZE]; } DEVLIST;</pre>	<pre>typedef struct _DEVLIST {     int num_devices;     char     dev_names[MAXDEVS][MAXDEVSIZE]; } DEVLIST;</pre>	list of available devices		devs.h
DevAttrs	<pre>typedef struct _DevAttrs {     unsigned char ha[6];     UINT32 ip_addr; } DevAttrs;</pre>	<pre>typedef struct _DevAttrs {     unsigned char ha[6];     UINT32 ip_addr;     UINT32 mask;     long index; } DevAttrs;</pre>	Contains Mac and Ip addresses;		devs.h
FDesc	<pre>typedef HANDLE FDesc;</pre>	<pre>typedef int FDesc;</pre>	file descriptor		log.h
PICApacket	<pre>typedef struct _PICApacket {     int packet_size;     void * data;     struct _PICApacket * next; } PICApacket;</pre>	<pre>typedef struct _PICApacket {     int packet_size;     void * data;     struct _PICApacket * next; } PICApacket;</pre>	Represents a network packet storing information about data, data size and which is next packet.		mem.h
PICAbuffer	<pre>typedef struct _PICAbuffer {     int tot_queues;     HANDLE * buf_mut;     struct _PICApacket **     packet_queues; } PICAbuffer;</pre>	<pre>typedef struct _PICAbuffer {     int tot_queues;     pthread_mutex_t * buf_mut;     struct _PICApacket **     packet_queues; } PICAbuffer;</pre>	Allows to manage one or more packets queues		mem.h
PICApipes	<pre>typedef struct _PICApipes {     HANDLE pipe;     HANDLE event;</pre>	<pre>typedef int PICApipes;</pre>	represents a pipe		mem.h

	<code>} PICAPipe;</code>				
PICA_IO_DEVICE	<code>typedef LPADAPTER PICA_IO_DEVICE;</code>	<code>typedef int PICA_IO_DEVICE;</code>	represents the device on which send and receive pakets.		packet.h
RTLines	<code>typedef struct _RTLines {     UINT32 dest;     UINT32 mask;     UINT32 gw;     int metric;     char * device;     struct _RTLines * next; } RTLines;</code>	<code>typedef struct _RTLines {     UINT32 dest;     UINT32 mask;     UINT32 gw;     int metric;     char * device;     struct _RTLines * next; } RTLines;</code>	Represents an entry in the routing table		rtmanager.h
RTInfo	<code>typedef struct _RTInfo {     int entry_count;     RTLines * lines; } RTInfo;</code>	<code>typedef struct _RTInfo {     int entry_count;     RTLines * lines; } RTInfo;</code>	Represents a routing table as a list of entries.		rtmanager.h
PICASocket	<code>typedef SOCKET PICASocket;</code>	<code>typedef int PICASocket;</code>			sock.h
THRID	<code>typedef HANDLE THRID;</code>	<code>typedef pthread_t THRID;</code>			thr.h
PICAMutex	<code>typedef HANDLE PICAMutex;</code>	<code>typedef pthread_mutex_t PICAMutex;</code>			thr.h
PICASemaphore	<code>typedef HANDLE PICASemaphore;</code>	<code>typedef struct _PICASemaphore {     sem_t sem;     pthread_mutex_t sem_mutex;     int max_count; } PICASemaphore;</code>			thr.h
PICAdescList	<code>typedef struct _PICAdescList {     int type;     void * desc;     struct _PICAdescList * next; } PICAdescList;</code>	<code>typedef struct _PICAdescList {     int type;     void * desc;     struct _PICAdescList * next; } PICAdescList;</code>	list of descriptors on which it is possible to call the select function		thr.h
PICAselResult	<code>typedef struct _PICAselResult {     int type;</code>	<code>typedef struct _PICAselResult {     int type;</code>	result of the select		thr.h

	<pre>void * desc; } PICAselResult;</pre>	<pre>void * desc; } PICAselResult;</pre>	function		
Victim_t		<pre>typedef struct { int inuse; pthread_t in; } Victim_t;</pre>	store information about threads in suspend state		thr.c
UINT64	<pre>typedef unsigned __int64 UINT64;</pre>	<pre>typedef u_int64_t UINT64;</pre>			timer.h
<pre>struct prioqent</pre>	<pre>struct prioqent { UINT64 tv; void (*callback)(void *); void *data; struct prioqent *pqe; };</pre>	<pre>struct prioqent { UINT64 tv; void (*callback)(void *); void *data; struct prioqent *pqe; };</pre>	represents an element of the timer's queue		timer.h
<pre>struct prioq</pre>	<pre>struct prioq { struct prioqent *pqe; };</pre>	<pre>struct prioq { struct prioqent *pqe; };</pre>	represents the timer's queue		timer.h
<pre>struct itimerval</pre>	<pre>struct itimerval { struct timeval it_interval; struct timeval it_value; };</pre>				timer.h