

Splash Reference Manual
1.0(build20030415)

Generated by Doxygen 1.2.18

Wed Apr 16 18:28:50 2003

Contents

1	Splash Main Page	1
2	Splash Data Structure Index	3
2.1	Splash Data Structures	3
3	Splash File Index	5
3.1	Splash File List	5
4	Splash Data Structure Documentation	9
4.1	aftrans_t Struct Reference	9
4.2	aftype Struct Reference	11
4.3	buffer_t Struct Reference	13
4.4	cmdline_args Struct Reference	14
4.5	glob_conf Struct Reference	15
4.6	hash_table_t Struct Reference	16
4.7	heap_obj Struct Reference	18
4.8	heap_t Struct Reference	19
4.9	hwtype Struct Reference	20
4.10	interface Struct Reference	21
4.11	l Struct Reference	26
4.12	label_mapping_t Struct Reference	27
4.13	packet_t Struct Reference	28
4.14	pair_t Struct Reference	31
4.15	rt_lookup Struct Reference	32
4.16	rtentry Struct Reference	33
4.17	snap_htup Struct Reference	36

4.18	snap_hval Struct Reference	37
4.19	snap_svc_ifip_item Struct Reference	39
4.20	snap_svc_rec Struct Reference	40
4.21	snap_hdr Struct Reference	41
4.22	svc_returnitem Struct Reference	43
4.23	svc_returnstruct Struct Reference	45
4.24	user_net_device_stats Struct Reference	46
4.25	yy_buffer_state Struct Reference	49
4.26	yyalloc Union Reference	51
4.27	yystype Union Reference	52
5	Splash File Documentation	53
5.1	doxyintro.c File Reference	53
5.2	snap-1.1-wjdb/lib/af.c File Reference	54
5.3	snap-1.1-wjdb/lib/bytecode.h File Reference	58
5.4	snap-1.1-wjdb/lib/config.h File Reference	85
5.5	snap-1.1-wjdb/lib/consts.h File Reference	87
5.6	snap-1.1-wjdb/lib/d_printf.c File Reference	89
5.7	snap_svc/d_printf.c File Reference	91
5.8	snap-1.1-wjdb/lib/d_printf.h File Reference	94
5.9	snap_svc/d_printf.h File Reference	96
5.10	snap-1.1-wjdb/lib/dyncheck.h File Reference	98
5.11	snap-1.1-wjdb/lib/exception.h File Reference	100
5.12	snap-1.1-wjdb/lib/hashtable.h File Reference	102
5.13	snap-1.1-wjdb/lib/hw.c File Reference	104
5.14	snap-1.1-wjdb/lib/inet.c File Reference	108
5.15	snap-1.1-wjdb/lib/interface.c File Reference	109
5.16	snap-1.1-wjdb/lib/interface.h File Reference	111
5.17	snap-1.1-wjdb/lib/interp.h File Reference	112
5.18	snap-1.1-wjdb/lib/intl.h File Reference	113
5.19	snap-1.1-wjdb/lib/io.h File Reference	114
5.20	snap-1.1-wjdb/lib/libsnap.c File Reference	116
5.21	snap-1.1-wjdb/lib/libsnap.h File Reference	122
5.22	snap-1.1-wjdb/lib/list.h File Reference	124

5.23	snap-1.1-wjdb/lib/loopback.c File Reference	125
5.24	snap-1.1-wjdb/lib/memalloc.h File Reference	127
5.25	snap-1.1-wjdb/lib/myassert.h File Reference	129
5.26	snap-1.1-wjdb/lib/net-support.h File Reference	130
5.27	snap-1.1-wjdb/lib/packet.c File Reference	138
5.28	snap-1.1-wjdb/lib/packet.h File Reference	140
5.29	snap-1.1-wjdb/lib/pathnames.h File Reference	142
5.30	snap-1.1-wjdb/lib/printval.h File Reference	147
5.31	snap-1.1-wjdb/lib/proc.c File Reference	149
5.32	snap-1.1-wjdb/lib/proc.h File Reference	150
5.33	snap-1.1-wjdb/lib/router.c File Reference	151
5.34	snap-1.1-wjdb/lib/router.h File Reference	154
5.35	snap-1.1-wjdb/lib/snap.h File Reference	156
5.36	snap-1.1-wjdb/lib/snap_bytecode.c File Reference	157
5.37	snap-1.1-wjdb/lib/snap_hashtable.c File Reference	161
5.38	snap-1.1-wjdb/lib/snap_interp.c File Reference	163
5.39	snap-1.1-wjdb/lib/snap_io.c File Reference	165
5.40	snap-1.1-wjdb/lib/snap_kern_iface.c File Reference	169
5.41	snap-1.1-wjdb/lib/snap_kern_iface.h File Reference	172
5.42	snap-1.1-wjdb/lib/snap_list.c File Reference	173
5.43	snap-1.1-wjdb/lib/snap_svc_conversion.c File Reference	174
5.44	snap-1.1-wjdb/lib/snap_svc_conversion.h File Reference	176
5.45	snap-1.1-wjdb/lib/snap_svc_handler.c File Reference	178
5.46	snap-1.1-wjdb/lib/snap_svc_handler.h File Reference	179
5.47	snap-1.1-wjdb/lib/snap_svc_library_handler.c File Reference	180
5.48	snap-1.1-wjdb/lib/snap_svc_library_handler.h File Reference	182
5.49	snap-1.1-wjdb/lib/snap_svc_reg_handler.c File Reference	185
5.50	snap-1.1-wjdb/lib/snap_svc_reg_handler.h File Reference	189
5.51	snap-1.1-wjdb/lib/snap_svc_reg_table.c File Reference	192
5.52	snap-1.1-wjdb/lib/snap_svc_reg_table.h File Reference	194
5.53	snap-1.1-wjdb/lib/snapnet.c File Reference	196
5.54	snap-1.1-wjdb/lib/snapnet.h File Reference	198
5.55	snap-1.1-wjdb/lib/sockets.c File Reference	199
5.56	snap-1.1-wjdb/lib/sockets.h File Reference	200

5.57	snap-1.1-wjdb/lib/timers.h File Reference	202
5.58	snap-1.1-wjdb/lib/unix.c File Reference	205
5.59	snap-1.1-wjdb/lib/version.h File Reference	206
5.60	snap-1.1-wjdb/lib/warn.h File Reference	207
5.61	snap-1.1-wjdb/lib/wassert.h File Reference	208
5.62	snap-1.1-wjdb/src/snap_demux_handler.c File Reference	209
5.63	snap-1.1-wjdb/src/snap_demux_handler.h File Reference	214
5.64	snap-1.1-wjdb/src/snap_exec.c File Reference	217
5.65	snap-1.1-wjdb/src/snap_sendandreceive.c File Reference	222
5.66	snap-1.1-wjdb/src/snapd.c File Reference	227
5.67	snap-1.1-wjdb/utils/consts.c File Reference	228
5.68	snap-1.1-wjdb/utils/kinject.c File Reference	229
5.69	snap-1.1-wjdb/utils/labels.c File Reference	233
5.70	snap-1.1-wjdb/utils/labels.h File Reference	235
5.71	snap-1.1-wjdb/utils/snapas.c File Reference	237
5.72	snap-1.1-wjdb/utils/snapdis.c File Reference	242
5.73	snap-1.1-wjdb/utils/snaplex.c File Reference	244
5.74	snap-1.1-wjdb/utils/snapparse.c File Reference	256
5.75	snap-1.1-wjdb/utils/snapparse.tab.h File Reference	278
5.76	snap_svc/snap_svc.c File Reference	290
5.77	snap_svc/snap_svc.h File Reference	292
5.78	snap_svc/snap_svc_if.c File Reference	297
5.79	snap_svc/snap_svc_if.h File Reference	302
5.80	snap_svc/snap_svc_memmap.c File Reference	305
5.81	snap_svc/snap_svc_memmap.h File Reference	308
5.82	snap_svc/snap_svc_memmap_hash.c File Reference	310
5.83	snap_svc/snap_svc_memmap_hash.h File Reference	312
5.84	snap_svc/snap_svc_memmap_hash_list.c File Reference	314
5.85	snap_svc/snap_svc_memmap_hash_list.h File Reference	315
5.86	snap_svc/snap_svc_proc.c File Reference	317
5.87	snap_svc/snap_svc_proc.h File Reference	319
5.88	snap_svc/snap_svc_route.c File Reference	320
5.89	snap_svc/snap_svc_route.h File Reference	323
5.90	snap_svc/snap_svc_snmp.c File Reference	324

5.91	snap_svc/snap_svc_snmp.h File Reference	332
5.92	snap_svc/snap_svc_TEMPLATE.c File Reference	337
5.93	snap_svc/snap_svc_TEMPLATE.h File Reference	339
5.94	snap_svc/snap_svc_test.c File Reference	340
5.95	snap_svc/snap_svc_test.h File Reference	342

Chapter 1

Splash Main Page

Auto generated Reference Manual

1.0.1 Notice

the following reference material is created from the sourcecode using `doxygen`. please note that not all C style comments have been translated into doxygen style yet, therefore autogenerated commentary may be very brief. View the complete code for further information.

Chapter 2

Splash Data Structure Index

2.1 Splash Data Structures

Here are the data structures with brief descriptions:

aftrans_t	9
aftype	11
buffer_t	13
cmdline_args	14
glob_conf	15
hash_table_t	16
heap_obj	18
heap_t	19
hwtype	20
interface	21
l	26
label_mapping_t	27
packet_t	28
pair_t	31
rt_lookup	32
rtentry	33
snap_htup	36
snap_hval	37
snap_svc_ifip_item	39
snap_svc_rec	40
snaphdr	41
svc_returnitem	43
svc_returnstruct	45
user_net_device_stats	46
yy_buffer_state	49
yyalloc	51
yystype	52

Chapter 3

Splash File Index

3.1 Splash File List

Here is a list of all files with brief descriptions:

doxyintro.c	53
snap-1.1-wjdb/lib/ af.c	54
snap-1.1-wjdb/lib/ bytecode.h	58
snap-1.1-wjdb/lib/ config.h	85
snap-1.1-wjdb/lib/ consts.h	87
snap-1.1-wjdb/lib/ d_printf.c	89
snap-1.1-wjdb/lib/ d_printf.h	94
snap-1.1-wjdb/lib/ dyncheck.h	98
snap-1.1-wjdb/lib/ exception.h	100
snap-1.1-wjdb/lib/ hashtable.h	102
snap-1.1-wjdb/lib/ hw.c	104
snap-1.1-wjdb/lib/ inet.c	108
snap-1.1-wjdb/lib/ interface.c	109
snap-1.1-wjdb/lib/ interface.h	111
snap-1.1-wjdb/lib/ interp.h	112
snap-1.1-wjdb/lib/ intl.h	113
snap-1.1-wjdb/lib/ io.h	114
snap-1.1-wjdb/lib/ libsnap.c	116
snap-1.1-wjdb/lib/ libsnap.h	122
snap-1.1-wjdb/lib/ list.h	124
snap-1.1-wjdb/lib/ loopback.c	125
snap-1.1-wjdb/lib/ memalloc.h	127
snap-1.1-wjdb/lib/ myassert.h	129
snap-1.1-wjdb/lib/ net-support.h	130
snap-1.1-wjdb/lib/ packet.c	138
snap-1.1-wjdb/lib/ packet.h	140
snap-1.1-wjdb/lib/ pathnames.h	142
snap-1.1-wjdb/lib/ printval.h	147
snap-1.1-wjdb/lib/ proc.c	149

snap-1.1-wjdb/lib/ proc.h	150
snap-1.1-wjdb/lib/ router.c	151
snap-1.1-wjdb/lib/ router.h	154
snap-1.1-wjdb/lib/ snap.h	156
snap-1.1-wjdb/lib/ snap_bytecode.c	157
snap-1.1-wjdb/lib/ snap_hashtable.c	161
snap-1.1-wjdb/lib/ snap_interp.c	163
snap-1.1-wjdb/lib/ snap_io.c	165
snap-1.1-wjdb/lib/ snap_kern_iface.c	169
snap-1.1-wjdb/lib/ snap_kern_iface.h	172
snap-1.1-wjdb/lib/ snap_list.c	173
snap-1.1-wjdb/lib/ snap_svc_conversion.c	174
snap-1.1-wjdb/lib/ snap_svc_conversion.h	176
snap-1.1-wjdb/lib/ snap_svc_handler.c	178
snap-1.1-wjdb/lib/ snap_svc_handler.h	179
snap-1.1-wjdb/lib/ snap_svc_library_handler.c	180
snap-1.1-wjdb/lib/ snap_svc_library_handler.h	182
snap-1.1-wjdb/lib/ snap_svc_reg_handler.c	185
snap-1.1-wjdb/lib/ snap_svc_reg_handler.h	189
snap-1.1-wjdb/lib/ snap_svc_reg_table.c	192
snap-1.1-wjdb/lib/ snap_svc_reg_table.h	194
snap-1.1-wjdb/lib/ snapnet.c	196
snap-1.1-wjdb/lib/ snapnet.h	198
snap-1.1-wjdb/lib/ sockets.c	199
snap-1.1-wjdb/lib/ sockets.h	200
snap-1.1-wjdb/lib/ timers.h	202
snap-1.1-wjdb/lib/ unix.c	205
snap-1.1-wjdb/lib/ version.h	206
snap-1.1-wjdb/lib/ warn.h	207
snap-1.1-wjdb/lib/ wassert.h	208
snap-1.1-wjdb/src/ snap_demux_handler.c	209
snap-1.1-wjdb/src/ snap_demux_handler.h	214
snap-1.1-wjdb/src/ snap_exec.c	217
snap-1.1-wjdb/src/ snap_sendandreceive.c	222
snap-1.1-wjdb/src/ snaped.c	227
snap-1.1-wjdb/utls/ consts.c	228
snap-1.1-wjdb/utls/ kinject.c	229
snap-1.1-wjdb/utls/ labels.c	233
snap-1.1-wjdb/utls/ labels.h	235
snap-1.1-wjdb/utls/ snapas.c	237
snap-1.1-wjdb/utls/ snapdis.c	242
snap-1.1-wjdb/utls/ snaplex.c	244
snap-1.1-wjdb/utls/ snapparse.c	256
snap-1.1-wjdb/utls/ snapparse.tab.h	278
snap_svc/ d_printf.c	91
snap_svc/ d_printf.h	96
snap_svc/ snap_svc.c	290
snap_svc/ snap_svc.h	292
snap_svc/ snap_svc_if.c	297
snap_svc/ snap_svc_if.h	302

snap_svc/snap_svc_memmap.c	305
snap_svc/snap_svc_memmap.h	308
snap_svc/snap_svc_memmap_hash.c	310
snap_svc/snap_svc_memmap_hash.h	312
snap_svc/snap_svc_memmap_hash_list.c	314
snap_svc/snap_svc_memmap_hash_list.h	315
snap_svc/snap_svc_proc.c	317
snap_svc/snap_svc_proc.h	319
snap_svc/snap_svc_route.c	320
snap_svc/snap_svc_route.h	323
snap_svc/snap_svc_snmp.c	324
snap_svc/snap_svc_snmp.h	332
snap_svc/snap_svc_TEMPLATE.c	337
snap_svc/snap_svc_TEMPLATE.h	339
snap_svc/snap_svc_test.c	340
snap_svc/snap_svc_test.h	342

Chapter 4

Splash Data Structure Documentation

4.1 aftrans_t Struct Reference

Data Fields

- char * **alias**
- char * **name**
- int * **flag**

4.1.1 Field Documentation

4.1.1.1 char* aftrans_t::alias

Definition at line 41 of file af.c.

Referenced by aftrans_opt().

4.1.1.2 int* aftrans_t::flag

Definition at line 43 of file af.c.

Referenced by aftrans_opt().

4.1.1.3 char* aftrans_t::name

Definition at line 42 of file af.c.

Referenced by aftrans_opt().

The documentation for this struct was generated from the following file:

- `snap-1.1-wjdb/lib/af.c`

4.2 aftype Struct Reference

```
#include <net-support.h>
```

Data Fields

- char * **name**
- char * **title**
- int **af**
- int **alen**
- char *(* **print**)(unsigned char *)
- char *(* **sprint**)(struct sockaddr *, int numeric)
- int(* **input**)(int type, char *bufp, struct sockaddr *)
- void(* **herror**)(char *text)
- int(* **rprint**)(int options)
- int(* **rinput**)(int typ, int ext, char **argv)
- int(* **getmask**)(char *src, struct sockaddr *mask, char *name)

4.2.1 Field Documentation

4.2.1.1 int aftype::af

Definition at line 38 of file net-support.h.

4.2.1.2 int aftype::alen

Definition at line 39 of file net-support.h.

4.2.1.3 int(* aftype::getmask)(char *src, struct sockaddr *mask, char *name)

4.2.1.4 void(* aftype::herror)(char *text)

4.2.1.5 int(* aftype::input)(int type, char *bufp, struct sockaddr *)

4.2.1.6 char* aftype::name

Definition at line 36 of file net-support.h.

4.2.1.7 `char>(* aftype::print)(unsigned char *)`

4.2.1.8 `int(* aftype::rinput)(int typ, int ext, char **argv)`

4.2.1.9 `int(* aftype::rprint)(int options)`

4.2.1.10 `char(* aftype::sprint)(struct sockaddr *, int numeric)`

4.2.1.11 `char* aftype::title`

Definition at line 37 of file net-support.h.

Referenced by `afinit()`.

The documentation for this struct was generated from the following file:

- `snap-1.1-wjdb/lib/net-support.h`

4.3 `buffer_t` Struct Reference

```
#include <io.h>
```

Data Fields

- `int lenb`
- `char * s`

4.3.1 Field Documentation

4.3.1.1 `int buffer_t::lenb`

Definition at line 17 of file `io.h`.

Referenced by `file_to_str()`, `init_request()`, `main()`, `marshal_packet()`, `newwho()`, and `sendpkt()`.

4.3.1.2 `char* buffer_t::s`

Definition at line 18 of file `io.h`.

Referenced by `file_to_str()`, `main()`, `marshal_packet()`, `newwho()`, `sendpkt()`, and `yyparse()`.

The documentation for this struct was generated from the following file:

- `snap-1.1-wjdb/lib/io.h`

4.4 cmdline_args Struct Reference

Data Fields

- int **argc**
- char ** **argv**

4.4.1 Field Documentation

4.4.1.1 int cmdline_args::argc

Definition at line 60 of file libsnap.c.

Referenced by `init_snap()`, and `snap()`.

4.4.1.2 char** cmdline_args::argv

Definition at line 61 of file libsnap.c.

Referenced by `init_snap()`.

The documentation for this struct was generated from the following file:

- `snap-1.1-wjdb/lib/libsnap.c`

4.5 glob_conf Struct Reference

Data Fields

- `sockaddr_in herehint`

4.5.1 Field Documentation

4.5.1.1 struct sockaddr_in glob_conf::herehint

Definition at line 56 of file `libsnap.c`.

Referenced by `parse_cmdline_snap()`, and `snap()`.

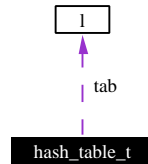
The documentation for this struct was generated from the following file:

- `snap-1.1-wjdb/lib/libsnap.c`

4.6 hash_table_t Struct Reference

```
#include <hashtable.h>
```

Collaboration diagram for hash_table_t:



Data Fields

- `int(* cmp)(const void *, const void *)`
- `int(* hash)(void *)`
- `int max_len`
- `list_t ** tab`
- `int tab_sz`
- `list_t ** tab`

4.6.1 Field Documentation

4.6.1.1 `int(* hash_table_t::cmp)(const void *, const void *)`

Referenced by `ht_create()`, `ht_lookup()`, and `ht_remove()`.

4.6.1.2 `int(* hash_table_t::hash)(void *)`

Referenced by `ht_create()`, `ht_insert()`, `ht_lookup()`, and `ht_remove()`.

4.6.1.3 `int hash_table_t::max_len`

Definition at line 18 of file `snap_svc_memmap_hash.h`.

Referenced by `ht_create()`, and `ht_insert()`.

4.6.1.4 `list_t** hash_table_t::tab`

Definition at line 19 of file `snap_svc_memmap_hash.h`.

4.6.1.5 list_t hash_table_t::tab**

Definition at line 19 of file hashtable.h.

Referenced by ht_create(), ht_insert(), ht_lookup(), and ht_remove().

4.6.1.6 int hash_table_t::tab_sz

Definition at line 20 of file snap_svc_memmap_hash.h.

Referenced by ht_create(), ht_insert(), ht_lookup(), and ht_remove().

The documentation for this struct was generated from the following files:

- snap-1.1-wjdb/lib/hashtable.h
- snap_svc/snap_svc_memmap_hash.h

4.7 heap_obj Struct Reference

```
#include <bytecode.h>
```

Data Fields

- LENTYPE **len**
- unsigned short **flag**
- char **s** [0]

4.7.1 Field Documentation

4.7.1.1 unsigned short heap_obj::flag

Definition at line 67 of file `bytecode.h`.

Referenced by `marshal_packet()`, `newwho()`, `newtup()`, and `snap_svc_convert_returnstruct2stack()`.

4.7.1.2 LENTYPE heap_obj::len

Definition at line 66 of file `bytecode.h`.

Referenced by `fprintf_value_heap()`, `marshal_packet()`, `newwho()`, `newtup()`, and `snap_svc_convert_returnstruct2stack()`.

4.7.1.3 char heap_obj::s[0]

Definition at line 71 of file `bytecode.h`.

Referenced by `fprintf_value_heap()`, `marshal_packet()`, `newwho()`, `newtup()`, and `snap_svc_convert_returnstruct2stack()`.

The documentation for this struct was generated from the following file:

- `snap-1.1-wjdb/lib/bytecode.h`

4.8 heap_t Struct Reference

```
#include <packet.h>
```

Data Fields

- int lenb
- char * h

4.8.1 Field Documentation

4.8.1.1 char* heap_t::h

Definition at line 18 of file packet.h.

4.8.1.2 int heap_t::lenb

Definition at line 17 of file packet.h.

The documentation for this struct was generated from the following file:

- snap-1.1-wjdb/lib/**packet.h**

4.9 hwtype Struct Reference

```
#include <net-support.h>
```

Data Fields

- char * **name**
- char * **title**
- int **type**
- int **alen**
- char>(* **print**)(unsigned char *)
- char>(* **sprint**)(struct sockaddr *)
- int(* **input**)(char *, struct sockaddr *)
- int(* **activate**)(int fd)

4.9.1 Field Documentation

4.9.1.1 int(* hwtype::activate)(int fd)

4.9.1.2 int hwtype::alen

Definition at line 58 of file net-support.h.

4.9.1.3 int(* hwtype::input)(char *, struct sockaddr *)

4.9.1.4 char* hwtype::name

Definition at line 55 of file net-support.h.

4.9.1.5 char>(* hwtype::print)(unsigned char *)

4.9.1.6 char>(* hwtype::sprint)(struct sockaddr *)

4.9.1.7 char* hwtype::title

Definition at line 56 of file net-support.h.

Referenced by hwinit().

4.9.1.8 int hwtype::type

Definition at line 57 of file net-support.h.

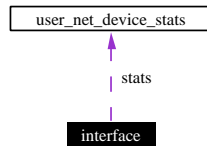
The documentation for this struct was generated from the following file:

- snap-1.1-wjdb/lib/**net-support.h**

4.10 interface Struct Reference

```
#include <interface.h>
```

Collaboration diagram for interface:



Data Fields

- char **name** [IFNAMSIZ]
- short **type**
- short **flags**
- int **metric**
- int **mtu**
- int **tx_queue_len**
- ifmap **map**
- sockaddr **addr**
- sockaddr **dstaddr**
- sockaddr **broadaddr**
- sockaddr **netmask**
- sockaddr **ipxaddr_bb**
- sockaddr **ipxaddr_sn**
- sockaddr **ipxaddr_e3**
- sockaddr **ipxaddr_e2**
- sockaddr **ddpaddr**
- sockaddr **ecaddr**
- int **has_ip**
- int **has_ipx_bb**
- int **has_ipx_sn**
- int **has_ipx_e3**
- int **has_ipx_e2**
- int **has_ax25**
- int **has_ddp**
- int **has_econet**
- char **hwaddr** [32]
- **user_net_device_stats** stats

4.10.1 Field Documentation

4.10.1.1 `struct sockaddr interface::addr`

Definition at line 39 of file interface.h.

Referenced by `if_fetch()`.

4.10.1.2 `struct sockaddr interface::broadaddr`

Definition at line 41 of file interface.h.

Referenced by `if_fetch()`.

4.10.1.3 `struct sockaddr interface::ddpaddr`

Definition at line 47 of file interface.h.

Referenced by `if_fetch()`.

4.10.1.4 `struct sockaddr interface::dstaddr`

Definition at line 40 of file interface.h.

Referenced by `if_fetch()`.

4.10.1.5 `struct sockaddr interface::ecaddr`

Definition at line 48 of file interface.h.

Referenced by `if_fetch()`.

4.10.1.6 `short interface::flags`

Definition at line 34 of file interface.h.

Referenced by `if_fetch()`.

4.10.1.7 `int interface::has_ax25`

Definition at line 54 of file interface.h.

4.10.1.8 `int interface::has_ddp`

Definition at line 55 of file interface.h.

Referenced by `if_fetch()`.

4.10.1.9 int interface::has_econet

Definition at line 56 of file interface.h.

Referenced by if_fetch().

4.10.1.10 int interface::has_ip

Definition at line 49 of file interface.h.

4.10.1.11 int interface::has_ipx_bb

Definition at line 50 of file interface.h.

Referenced by if_fetch().

4.10.1.12 int interface::has_ipx_e2

Definition at line 53 of file interface.h.

Referenced by if_fetch().

4.10.1.13 int interface::has_ipx_e3

Definition at line 52 of file interface.h.

Referenced by if_fetch().

4.10.1.14 int interface::has_ipx_sn

Definition at line 51 of file interface.h.

Referenced by if_fetch().

4.10.1.15 char interface::hwaddr[32]

Definition at line 57 of file interface.h.

Referenced by if_fetch().

4.10.1.16 struct sockaddr interface::ipxaddr_bb

Definition at line 43 of file interface.h.

Referenced by if_fetch().

4.10.1.17 struct sockaddr interface::ipxaddr_e2

Definition at line 46 of file interface.h.

Referenced by if_fetch().

4.10.1.18 struct sockaddr interface::ipxaddr_e3

Definition at line 45 of file interface.h.

Referenced by if_fetch().

4.10.1.19 struct sockaddr interface::ipxaddr_sn

Definition at line 44 of file interface.h.

Referenced by if_fetch().

4.10.1.20 struct ifmap interface::map

Definition at line 38 of file interface.h.

Referenced by if_fetch().

4.10.1.21 int interface::metric

Definition at line 35 of file interface.h.

Referenced by if_fetch().

4.10.1.22 int interface::mtu

Definition at line 36 of file interface.h.

Referenced by if_fetch().

4.10.1.23 char interface::name[IFNAMSIZ]

Definition at line 32 of file interface.h.

Referenced by if_fetch().

4.10.1.24 struct sockaddr interface::netmask

Definition at line 42 of file interface.h.

Referenced by if_fetch().

4.10.1.25 struct user_net_device_stats interface::stats

Definition at line 58 of file interface.h.

4.10.1.26 int interface::tx_queue_len

Definition at line 37 of file interface.h.

Referenced by if_fetch().

4.10.1.27 short interface::type

Definition at line 33 of file interface.h.

Referenced by if_fetch().

The documentation for this struct was generated from the following file:

- snap-1.1-wjdb/lib/**interface.h**

4.11 l Struct Reference

```
#include <list.h>
```

Data Fields

- void * v
- l * next
- void * v
- l * next

4.11.1 Field Documentation

4.11.1.1 struct l* l::next

Definition at line 21 of file snap_svc_memmap_hash_list.h.

4.11.1.2 struct l* l::next

Definition at line 13 of file list.h.

Referenced by cons(), free_list(), ht_remove(), length_list(), and newtup().

4.11.1.3 void* l::v

Definition at line 20 of file snap_svc_memmap_hash_list.h.

4.11.1.4 void* l::v

Definition at line 12 of file list.h.

Referenced by cons(), ht_remove(), and newtup().

The documentation for this struct was generated from the following files:

- snap-1.1-wjdb/lib/list.h
- snap_svc/snap_svc_memmap_hash_list.h

4.12 label_mapping_t Struct Reference

Data Fields

- char * **name**
- void * **where**

4.12.1 Field Documentation

4.12.1.1 char* label_mapping_t::name

Definition at line 21 of file labels.c.

4.12.1.2 void* label_mapping_t::where

Definition at line 22 of file labels.c.

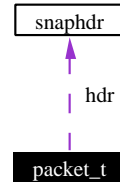
The documentation for this struct was generated from the following file:

- snap-1.1-wjdb/utils/**labels.c**

4.13 packet_t Struct Reference

```
#include <packet.h>
```

Collaboration diagram for packet_t:



Data Fields

- **header_t * hdr**
- unsigned char **rb**
- **instr_t * code_min**
- **instr_t * pc**
- **instr_t * handler**
- **instr_t * code_max**
- **value_t * stack_min**
- **value_t * sp**
- **value_t * stack_max**
- void * **heap_min**
- void * **h_alloc_ptr**
- void * **heap_max**
- void * **h_alloc_heap_max**
- unsigned int **is_contiguous**
- int **resized**
- iphdr * **iph**

4.13.1 Field Documentation

4.13.1.1 instr_t* packet_t::code_max

Definition at line 33 of file packet.h.

Referenced by `fprintf_packet()`, `main()`, `marshal_packet()`, and `unmarshal_packet()`.

4.13.1.2 instr_t* packet_t::code_min

Definition at line 30 of file packet.h.

Referenced by `fprintf_packet()`, `main()`, `marshal_packet()`, `patch_jumps()`, `unmarshal_packet()`, and `yyparse()`.

4.13.1.3 void* packet_t::h_alloc_heap_max

Definition at line 44 of file packet.h.

Referenced by main(), marshal_packet(), and unmarshal_packet().

4.13.1.4 void* packet_t::h_alloc_ptr

Definition at line 40 of file packet.h.

Referenced by main(), newwho(), and newtup().

4.13.1.5 instr_t* packet_t::handler

Definition at line 32 of file packet.h.

Referenced by unmarshal_packet().

4.13.1.6 header_t* packet_t::hdr

Definition at line 27 of file packet.h.

Referenced by fprintf_packet(), main(), marshal_packet(), unmarshal_packet(), and yyparse().

4.13.1.7 void* packet_t::heap_max

Definition at line 41 of file packet.h.

Referenced by main(), marshal_packet(), newwho(), newtup(), and unmarshal_packet().

4.13.1.8 void* packet_t::heap_min

Definition at line 39 of file packet.h.

Referenced by fprintf_packet(), fprintf_value(), main(), marshal_packet(), newwho(), newtup(), snap_svc_convert_returnstruct2stack(), snap_svc_convert_stack2returnstruct(), and unmarshal_packet().

4.13.1.9 struct iphdr* packet_t::iph

Definition at line 54 of file packet.h.

Referenced by marshal_packet(), and unmarshal_packet().

4.13.1.10 unsigned int packet_t::is_contiguous

Definition at line 48 of file packet.h.

Referenced by `main()`, `marshal_packet()`, and `unmarshal_packet()`.

4.13.1.11 `instr_t*` `packet_t::pc`

Definition at line 31 of file `packet.h`.

Referenced by `main()`, `marshal_packet()`, `patch_jumps()`, `unmarshal_packet()`, and `yyparse()`.

4.13.1.12 `unsigned char` `packet_t::rb`

Definition at line 28 of file `packet.h`.

4.13.1.13 `int` `packet_t::resized`

Definition at line 49 of file `packet.h`.

4.13.1.14 `value_t*` `packet_t::sp`

Definition at line 36 of file `packet.h`.

Referenced by `fprintf_packet()`, `main()`, `marshal_packet()`, `patch_jumps()`, `snap_svc_convert_direct2stack()`, `snap_svc_convert_returnstruct2stack()`, `snap_svc_convert_stack2arguments()`, `unmarshal_packet()`, and `yyparse()`.

4.13.1.15 `value_t*` `packet_t::stack_max`

Definition at line 37 of file `packet.h`.

Referenced by `main()`, `marshal_packet()`, `snap_svc_convert_returnstruct2stack()`, `unmarshal_packet()`, and `yyparse()`.

4.13.1.16 `value_t*` `packet_t::stack_min`

Definition at line 35 of file `packet.h`.

Referenced by `fprintf_packet()`, `main()`, `marshal_packet()`, `patch_jumps()`, `snap_svc_convert_stack2arguments()`, and `unmarshal_packet()`.

The documentation for this struct was generated from the following file:

- `snap-1.1-wjdb/lib/packet.h`

4.14 pair_t Struct Reference

```
#include <hashtable.h>
```

Data Fields

- void * **key**
- void * **value**
- void * **key**
- void * **value**

4.14.1 Field Documentation

4.14.1.1 void* pair_t::key

Definition at line 11 of file snap_svc_memmap_hash.h.

4.14.1.2 void* pair_t::key

Definition at line 11 of file hashtable.h.

Referenced by ht_insert(), and ht_remove().

4.14.1.3 void* pair_t::value

Definition at line 12 of file snap_svc_memmap_hash.h.

4.14.1.4 void* pair_t::value

Definition at line 12 of file hashtable.h.

Referenced by ht_insert().

The documentation for this struct was generated from the following files:

- snap-1.1-wjdb/lib/**hashtable.h**
- snap_svc/**snap_svc_memmap_hash.h**

4.15 rt_lookup Struct Reference

```
#include <router.h>
```

Data Fields

- `addr_t hopaddr`
- unsigned int `ifidx`

4.15.1 Field Documentation

4.15.1.1 `addr_t rt_lookup::hopaddr`

Definition at line 20 of file `router.h`.

Referenced by `nexthop()`.

4.15.1.2 `unsigned int rt_lookup::ifidx`

Definition at line 21 of file `router.h`.

Referenced by `nexthop()`.

The documentation for this struct was generated from the following file:

- `snap-1.1-wjdb/lib/router.h`

4.16 rentry Struct Reference

Data Fields

- **addr_t** **rt_dst**
- **addr_t** **rt_gateway**
- **addr_t** **rt_genmask**
- unsigned short int **rt_flags**
- short int **rt_pad2**
- unsigned long int **rt_ifidx**
- unsigned char **rt_tos**
- unsigned char **rt_class**
- short int **rt_pad4**
- short int **rt_metric**
- char * **rt_dev**
- unsigned long int **rt_mtu**
- unsigned long int **rt_window**
- unsigned short int **rt_irtt**

4.16.1 Field Documentation

4.16.1.1 unsigned char rentry::rt_class

Definition at line 50 of file router.c.

4.16.1.2 char* rentry::rt_dev

Definition at line 53 of file router.c.

Referenced by `handle_request()`.

4.16.1.3 addr_t rentry::rt_dst

Definition at line 43 of file router.c.

Referenced by `handle_request()`, `nexthop()`, and `read_routes()`.

4.16.1.4 unsigned short int rentry::rt_flags

Definition at line 46 of file router.c.

Referenced by `handle_request()`, and `read_routes()`.

4.16.1.5 addr_t rentry::rt_gateway

Definition at line 44 of file router.c.

Referenced by `handle_request()`, `nexthop()`, and `read_routes()`.

4.16.1.6 addr_t rentry::rt_genmask

Definition at line 45 of file router.c.

Referenced by handle_request(), nexthop(), and read_routes().

4.16.1.7 unsigned long int rentry::rt_ifidx

Definition at line 48 of file router.c.

Referenced by nexthop(), and read_routes().

4.16.1.8 unsigned short int rentry::rt_irtt

Definition at line 56 of file router.c.

Referenced by read_routes().

4.16.1.9 short int rentry::rt_metric

Definition at line 52 of file router.c.

Referenced by read_routes().

4.16.1.10 unsigned long int rentry::rt_mtu

Definition at line 54 of file router.c.

Referenced by read_routes().

4.16.1.11 short int rentry::rt_pad2

Definition at line 47 of file router.c.

4.16.1.12 short int rentry::rt_pad4

Definition at line 51 of file router.c.

4.16.1.13 unsigned char rentry::rt_tos

Definition at line 49 of file router.c.

4.16.1.14 unsigned long int rentry::rt_window

Definition at line 55 of file router.c.

Referenced by read_routes().

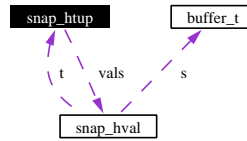
The documentation for this struct was generated from the following file:

- snap-1.1-wjdb/lib/**router.c**

4.17 snap_htup Struct Reference

```
#include <snap_svc_conversion.h>
```

Collaboration diagram for snap_htup:



Data Fields

- unsigned short int **n**
- **snap_hval ** vals**

4.17.1 Field Documentation

4.17.1.1 unsigned short int snap_htup::n

Definition at line 19 of file snap_svc_conversion.h.

Referenced by snap_svc_convert_stack2returnstruct().

4.17.1.2 struct snap_hval** snap_htup::vals

Definition at line 20 of file snap_svc_conversion.h.

Referenced by snap_svc_convert_stack2returnstruct().

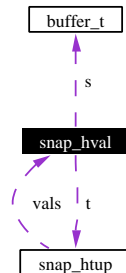
The documentation for this struct was generated from the following file:

- snap-1.1-wjdb/lib/snap_svc_conversion.h

4.18 snap_hval Struct Reference

```
#include <snap_svc_conversion.h>
```

Collaboration diagram for snap_hval:



Data Fields

- unsigned short int **typetag**
- union {
 - unsigned int **scalar**
 - buffer_t** * **s**
 - snap_htup** * **t**
- } **v**

4.18.1 Field Documentation

4.18.1.1 **buffer_t*** snap_hval::s

Definition at line 28 of file snap_svc_conversion.h.

4.18.1.2 **unsigned int** snap_hval::scalar

Definition at line 27 of file snap_svc_conversion.h.

4.18.1.3 **struct snap_htup*** snap_hval::t

Definition at line 29 of file snap_svc_conversion.h.

4.18.1.4 **unsigned short int** snap_hval::typetag

Definition at line 25 of file snap_svc_conversion.h.

4.18.1.5 union { ... } snap_hval::v

The documentation for this struct was generated from the following file:

- `snap-1.1-wjdb/lib/snap_svc_conversion.h`

4.19 snap_svc_ifip_item Struct Reference

```
#include <snap_svc_if.h>
```

Data Fields

- char * **if_name**
- unsigned int **if_index**
- uint32_t **addr**

4.19.1 Field Documentation

4.19.1.1 uint32_t snap_svc_ifip_item::addr

Definition at line 25 of file snap_svc_if.h.

Referenced by if_getallneighbours(), if_getifaceidx(), and if_getnextiface().

4.19.1.2 unsigned int snap_svc_ifip_item::if_index

Definition at line 24 of file snap_svc_if.h.

Referenced by if_getifaceidx(), and snap_svc_ifip_init().

4.19.1.3 char* snap_svc_ifip_item::if_name

Definition at line 23 of file snap_svc_if.h.

Referenced by if_getoutiface(), and snap_external_svclib_done().

The documentation for this struct was generated from the following file:

- snap_svc/**snap_svc_if.h**

4.20 snap_svc_rec Struct Reference

```
#include <snap_svc_reg_table.h>
```

Data Fields

- **snapsvc_func_proto snapsvc_func**
- **int nargs**
- **int nret**

4.20.1 Field Documentation

4.20.1.1 int snap_svc_rec::nargs

Definition at line 19 of file snap_svc_reg_table.h.

Referenced by snap_svc_table_add().

4.20.1.2 int snap_svc_rec::nret

Definition at line 20 of file snap_svc_reg_table.h.

Referenced by snap_svc_table_add().

4.20.1.3 snapsvc_func_proto snap_svc_rec::snapsvc_func

Definition at line 18 of file snap_svc_reg_table.h.

Referenced by snap_svc_table_add().

The documentation for this struct was generated from the following file:

- snap-1.1-wjdb/lib/snap_svc_reg_table.h

4.21 snaphdr Struct Reference

```
#include <snap.h>
```

Data Fields

- `u_int32_t saddr`
- `u_int32_t daddr`
- `u_int8_t version`
- `u_int8_t flags`
- `u_int16_t sport`
- `u_int16_t entry_point`
- `u_int16_t code_sizeb`
- `u_int16_t heap_sizeb`
- `u_int16_t stack_sizeb`

4.21.1 Field Documentation

4.21.1.1 `u_int16_t snaphdr::code_sizeb`

Definition at line 22 of file `snap.h`.

Referenced by `marshal_packet()`, and `unmarshal_packet()`.

4.21.1.2 `u_int32_t snaphdr::daddr`

Definition at line 17 of file `snap.h`.

Referenced by `init_request()`, and `main()`.

4.21.1.3 `u_int16_t snaphdr::entry_point`

Definition at line 21 of file `snap.h`.

Referenced by `fprintf_packet()`, `unmarshal_packet()`, and `yyparse()`.

4.21.1.4 `u_int8_t snaphdr::flags`

Definition at line 19 of file `snap.h`.

Referenced by `init_request()`, and `main()`.

4.21.1.5 `u_int16_t snaphdr::heap_sizeb`

Definition at line 23 of file `snap.h`.

Referenced by `marshal_packet()`, and `unmarshal_packet()`.

4.21.1.6 u_int32_t snaphdr::saddr

Definition at line 16 of file snap.h.

Referenced by `init_request()`, and `main()`.

4.21.1.7 u_int16_t snaphdr::sport

Definition at line 20 of file snap.h.

Referenced by `fprintf_packet()`, `init_request()`, and `main()`.

4.21.1.8 u_int16_t snaphdr::stack_sizeb

Definition at line 24 of file snap.h.

Referenced by `marshal_packet()`, and `unmarshal_packet()`.

4.21.1.9 u_int8_t snaphdr::version

Definition at line 18 of file snap.h.

Referenced by `init_request()`, and `main()`.

The documentation for this struct was generated from the following file:

- `snap-1.1-wjdb/lib/snap.h`

4.22 `svc_returnitem` Struct Reference

```
#include <snap_svc.h>
```

Data Fields

- unsigned short **type**
- void * **data**
- unsigned long **length**
- unsigned long * **oid**
- size_t **oid_length**

4.22.1 Field Documentation

4.22.1.1 void* `svc_returnitem::data`

Definition at line 29 of file `snap_svc.h`.

Referenced by `if_getallneighbours()`, `snap_external_svclib_free_returnstruct()`, `snap_external_svclib_snmp_getallotherneighboursfromip()`, `snap_external_svclib_snmp_gethop()`, `snap_external_svclib_snmp_getiface()`, `snap_external_svclib_snmp_getifnumber()`, `snap_external_svclib_snmp_getnexthopfromip()`, `snap_external_svclib_snmp_INTERNAL_execpdu_handler()`, `snap_external_svclib_snmp_isupiface()`, `snap_svc_convert_direct2stack()`, and `snap_svc_convert_returnstruct2stack()`.

4.22.1.2 unsigned long `svc_returnitem::length`

Definition at line 30 of file `snap_svc.h`.

Referenced by `snap_external_svclib_snmp_getallotherneighboursfromip()`, `snap_external_svclib_snmp_gethop()`, `snap_external_svclib_snmp_getiface()`, `snap_external_svclib_snmp_getifnumber()`, `snap_external_svclib_snmp_INTERNAL_execpdu_handler()`, `snap_external_svclib_snmp_isupiface()`, `snap_svc_convert_direct2stack()`, and `snap_svc_convert_returnstruct2stack()`.

4.22.1.3 unsigned long* `svc_returnitem::oid`

Definition at line 31 of file `snap_svc.h`.

Referenced by `snap_external_svclib_free_returnstruct()`, `snap_external_svclib_snmp_gethop()`, `snap_external_svclib_snmp_INTERNAL_execpdu_handler()`, and `snap_svc_convert_direct2stack()`.

4.22.1.4 size_t `svc_returnitem::oid_length`

Definition at line 32 of file `snap_svc.h`.

Referenced by `snap_external_svclib_snmp_gethop()`, `snap_external_svclib_snmp-INTERNAL_execpdu_handler()`, and `snap_svc_convert_direct2stack()`.

4.22.1.5 unsigned short `svc_returnitem::type`

Definition at line 28 of file `snap_svc.h`.

Referenced by `if_getallneighbours()`, `snap_external_svclib_snmp-getallotherneighboursfromip()`, `snap_external_svclib_snmp_gethop()`, `snap-external_svclib_snmp_getiface()`, `snap_external_svclib_snmp_getifnumber()`, `snap_external_svclib_snmp_getnexthopfromip()`, `snap_external_svclib_snmp-INTERNAL_execpdu_handler()`, `snap_external_svclib_snmp_isupiface()`, `snap-vc_convert_direct2stack()`, and `snap_svc_convert_returnstruct2stack()`.

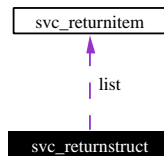
The documentation for this struct was generated from the following file:

- `snap_svc/snap_svc.h`

4.23 `svc_returnstruct` Struct Reference

```
#include <snap_svc.h>
```

Collaboration diagram for `svc_returnstruct`:



Data Fields

- `svc_returnitem * list`
- unsigned short `length`

4.23.1 Field Documentation

4.23.1.1 unsigned short `svc_returnstruct::length`

Definition at line 38 of file `snap_svc.h`.

Referenced by `if_getallneighbours()`, `snap_external_svclib_free_returnstruct()`, `snap_external_svclib_snmp_getallotherneighboursfromip()`, `snap_external_svclib_snmp_gethop()`, `snap_external_svclib_snmp_getiface()`, `snap_external_svclib_snmp_getifnumber()`, `snap_external_svclib_snmp_getnexthopfromip()`, `snap_external_svclib_snmp_INTERNAL_execpdu_handler()`, and `snap_external_svclib_snmp_isupiface()`.

4.23.1.2 `struct svc_returnitem* svc_returnstruct::list`

Definition at line 37 of file `snap_svc.h`.

Referenced by `if_getallneighbours()`, `snap_external_svclib_free_returnstruct()`, `snap_external_svclib_snmp_getallotherneighboursfromip()`, `snap_external_svclib_snmp_gethop()`, `snap_external_svclib_snmp_getiface()`, `snap_external_svclib_snmp_getifnumber()`, `snap_external_svclib_snmp_getnexthopfromip()`, `snap_external_svclib_snmp_INTERNAL_execpdu_handler()`, and `snap_external_svclib_snmp_isupiface()`.

The documentation for this struct was generated from the following file:

- `snap_svc/snap_svc.h`

4.24 user_net_device_stats Struct Reference

```
#include <interface.h>
```

Data Fields

- unsigned long **rx_packets**
- unsigned long **tx_packets**
- unsigned long **rx_bytes**
- unsigned long **tx_bytes**
- unsigned long **rx_errors**
- unsigned long **tx_errors**
- unsigned long **rx_dropped**
- unsigned long **tx_dropped**
- unsigned long **rx_multicast**
- unsigned long **rx_compressed**
- unsigned long **tx_compressed**
- unsigned long **collisions**
- unsigned long **rx_length_errors**
- unsigned long **rx_over_errors**
- unsigned long **rx_crc_errors**
- unsigned long **rx_frame_errors**
- unsigned long **rx_fifo_errors**
- unsigned long **rx_missed_errors**
- unsigned long **tx_aborted_errors**
- unsigned long **tx_carrier_errors**
- unsigned long **tx_fifo_errors**
- unsigned long **tx_heartbeat_errors**
- unsigned long **tx_window_errors**

4.24.1 Field Documentation

4.24.1.1 unsigned long user_net_device_stats::collisions

Definition at line 14 of file interface.h.

4.24.1.2 unsigned long user_net_device_stats::rx_bytes

Definition at line 5 of file interface.h.

4.24.1.3 unsigned long user_net_device_stats::rx_compressed

Definition at line 12 of file interface.h.

4.24.1.4 unsigned long user_net_device_stats::rx_crc_errors

Definition at line 19 of file interface.h.

4.24.1.5 unsigned long user_net_device_stats::rx_dropped

Definition at line 9 of file interface.h.

4.24.1.6 unsigned long user_net_device_stats::rx_errors

Definition at line 7 of file interface.h.

4.24.1.7 unsigned long user_net_device_stats::rx_fifo_errors

Definition at line 21 of file interface.h.

4.24.1.8 unsigned long user_net_device_stats::rx_frame_errors

Definition at line 20 of file interface.h.

4.24.1.9 unsigned long user_net_device_stats::rx_length_errors

Definition at line 17 of file interface.h.

4.24.1.10 unsigned long user_net_device_stats::rx_missed_errors

Definition at line 22 of file interface.h.

4.24.1.11 unsigned long user_net_device_stats::rx_multicast

Definition at line 11 of file interface.h.

4.24.1.12 unsigned long user_net_device_stats::rx_over_errors

Definition at line 18 of file interface.h.

4.24.1.13 unsigned long user_net_device_stats::rx_packets

Definition at line 3 of file interface.h.

4.24.1.14 unsigned long user_net_device_stats::tx_aborted_errors

Definition at line 24 of file interface.h.

4.24.1.15 unsigned long user_net_device_stats::tx_bytes

Definition at line 6 of file interface.h.

4.24.1.16 unsigned long user_net_device_stats::tx_carrier_errors

Definition at line 25 of file interface.h.

4.24.1.17 unsigned long user_net_device_stats::tx_compressed

Definition at line 13 of file interface.h.

4.24.1.18 unsigned long user_net_device_stats::tx_dropped

Definition at line 10 of file interface.h.

4.24.1.19 unsigned long user_net_device_stats::tx_errors

Definition at line 8 of file interface.h.

4.24.1.20 unsigned long user_net_device_stats::tx_fifo_errors

Definition at line 26 of file interface.h.

4.24.1.21 unsigned long user_net_device_stats::tx_heartbeat_errors

Definition at line 27 of file interface.h.

4.24.1.22 unsigned long user_net_device_stats::tx_packets

Definition at line 4 of file interface.h.

4.24.1.23 unsigned long user_net_device_stats::tx_window_errors

Definition at line 28 of file interface.h.

The documentation for this struct was generated from the following file:

- snap-1.1-wjdb/lib/**interface.h**

4.25 yy_buffer_state Struct Reference

Data Fields

- FILE * yy_input_file
- char * yy_ch_buf
- char * yy_buf_pos
- yy_size_t yy_buf_size
- int yy_n_chars
- int yy_is_our_buffer
- int yy_is_interactive
- int yy_at_bol
- int yy_fill_buffer
- int yy_buffer_status

4.25.1 Field Documentation

4.25.1.1 int yy_buffer_state::yy_at_bol

Definition at line 178 of file snaplex.c.

4.25.1.2 char* yy_buffer_state::yy_buf_pos

Definition at line 149 of file snaplex.c.

4.25.1.3 yy_size_t yy_buffer_state::yy_buf_size

Definition at line 154 of file snaplex.c.

4.25.1.4 int yy_buffer_state::yy_buffer_status

Definition at line 185 of file snaplex.c.

4.25.1.5 char* yy_buffer_state::yy_ch_buf

Definition at line 148 of file snaplex.c.

4.25.1.6 int yy_buffer_state::yy_fill_buffer

Definition at line 183 of file snaplex.c.

4.25.1.7 FILE* yy_buffer_state::yy_input_file

Definition at line 146 of file snaplex.c.

4.25.1.8 int yy_buffer_state::yy_is_interactive

Definition at line 172 of file snaplex.c.

4.25.1.9 int yy_buffer_state::yy_is_our_buffer

Definition at line 165 of file snaplex.c.

4.25.1.10 int yy_buffer_state::yy_n_chars

Definition at line 159 of file snaplex.c.

The documentation for this struct was generated from the following file:

- snap-1.1-wjdb/Utils/**snaplex.c**

4.26 yyalloC Union Reference

Data Fields

- short `yyss`
- `YYSTYPE` `yyvs`

4.26.1 Field Documentation

4.26.1.1 short `yyalloC::yyss`

Definition at line 511 of file `snapparse.c`.

4.26.1.2 `YYSTYPE` `yyalloC::yyvs`

Definition at line 512 of file `snapparse.c`.

The documentation for this union was generated from the following file:

- `snap-1.1-wjdb/utills/snapparse.c`

4.27 yystype Union Reference

```
#include <snapparse.tab.h>
```

Data Fields

- void * **ptr**
- int **integer**
- void * **ptr**

4.27.1 Field Documentation

4.27.1.1 int yystype::integer

Definition at line 7 of file snapparse.tab.h.

4.27.1.2 void* yystype::ptr

Definition at line 6 of file snapparse.tab.h.

4.27.1.3 void* yystype::ptr

Definition at line 162 of file snapparse.c.

The documentation for this union was generated from the following files:

- snap-1.1-wjdb/utils/**snapparse.c**
- snap-1.1-wjdb/utils/**snapparse.tab.h**

Chapter 5

Splash File Documentation

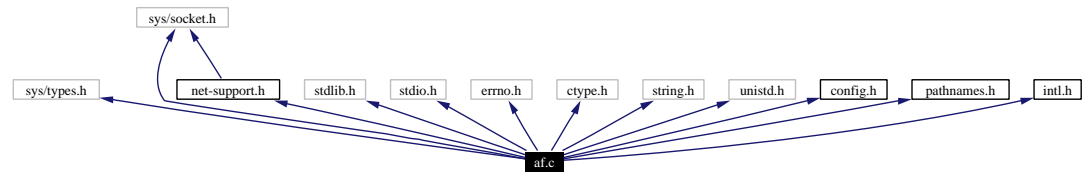
5.1 doxyintro.c File Reference



5.2 snap-1.1-wjdb/lib/af.c File Reference

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <ctype.h>
#include <string.h>
#include <unistd.h>
#include "config.h"
#include "net-support.h"
#include "pathnames.h"
#include "intl.h"
```

Include dependency graph for af.c:



Data Structures

- struct `aftrans_t`

Functions

- void `afinit` ()
- void `aftrans_def` (char *tool, char *argv0, char *dflt)
- `atype` * `get_atype` (const char *name)
- `atype` * `get_afntype` (int af)
- int `aftrans_opt` (const char *arg)

Variables

- int `flag_unx` = 0
- int `flag_ipx` = 0
- int `flag_ax25` = 0

- int **flag_ddp** = 0
- int **flag_netrom** = 0
- int **flag_inet** = 0
- int **flag_inet6** = 0
- int **flag_econet** = 0
- **aftrans_t** **aftrans** []
- char **afname** [256] = ""
- **aftype** **unspec_aftype**
- **aftype** **unix_aftype**
- **aftype** **inet_aftype**
- **aftype** **inet6_aftype**
- **aftype** **ax25_aftype**
- **aftype** **netrom_aftype**
- **aftype** **ipx_aftype**
- **aftype** **ddp_aftype**
- **aftype** **ec_aftype**

5.2.1 Function Documentation

5.2.1.1 void **afinit** ()

Definition at line 103 of file af.c.

References `_`, and `aftype::title`.

Referenced by `get_afntype()`, and `get_aftype()`.

5.2.1.2 void **aftrans_def** (char * *tool*, char * *argv0*, char * *dflt*)

Definition at line 135 of file af.c.

References `afname`, and `aftrans_opt()`.

5.2.1.3 int **aftrans_opt** (const char * *arg*)

Definition at line 211 of file af.c.

References `_`, `afname`, `aftrans`, `aftrans_t::alias`, `aftrans_t::flag`, and `aftrans_t::name`.

Referenced by `aftrans_def()`.

5.2.1.4 struct **aftype*** **get_afntype** (int *af*)

Definition at line 195 of file af.c.

References `afinit()`.

5.2.1.5 `struct aftype* get_aftype (const char * name)`

Definition at line 175 of file af.c.

References `_`, and `afinit()`.

5.2.2 Variable Documentation

5.2.2.1 `char afname[256] = ""`

Definition at line 60 of file af.c.

Referenced by `aftrans_def()`, and `aftrans_opt()`.

5.2.2.2 `struct aftrans_t aftrans[]`

Referenced by `aftrans_opt()`.

5.2.2.3 `struct aftype ax25_aftype`

Definition at line 66 of file af.c.

5.2.2.4 `struct aftype ddp_aftype`

Definition at line 69 of file af.c.

5.2.2.5 `struct aftype ec_aftype`

Definition at line 70 of file af.c.

5.2.2.6 `int flag_ax25 = 0`

Definition at line 32 of file af.c.

5.2.2.7 `int flag_ddp = 0`

Definition at line 33 of file af.c.

5.2.2.8 `int flag_econet = 0`

Definition at line 37 of file af.c.

5.2.2.9 `int flag_inet = 0`

Definition at line 35 of file af.c.

5.2.2.10 int flag_inet6 = 0

Definition at line 36 of file af.c.

5.2.2.11 int flag_ipx = 0

Definition at line 31 of file af.c.

5.2.2.12 int flag_netrom = 0

Definition at line 34 of file af.c.

5.2.2.13 int flag_unx = 0

Definition at line 30 of file af.c.

5.2.2.14 struct aftype inet6_aftype

Definition at line 65 of file af.c.

5.2.2.15 struct aftype inet_aftype

Definition at line 64 of file af.c.

5.2.2.16 struct aftype ipx_aftype

Definition at line 68 of file af.c.

5.2.2.17 struct aftype netrom_aftype

Definition at line 67 of file af.c.

5.2.2.18 struct aftype unix_aftype

Definition at line 63 of file af.c.

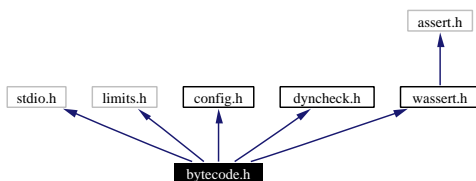
5.2.2.19 struct aftype unspec_aftype

Definition at line 62 of file af.c.

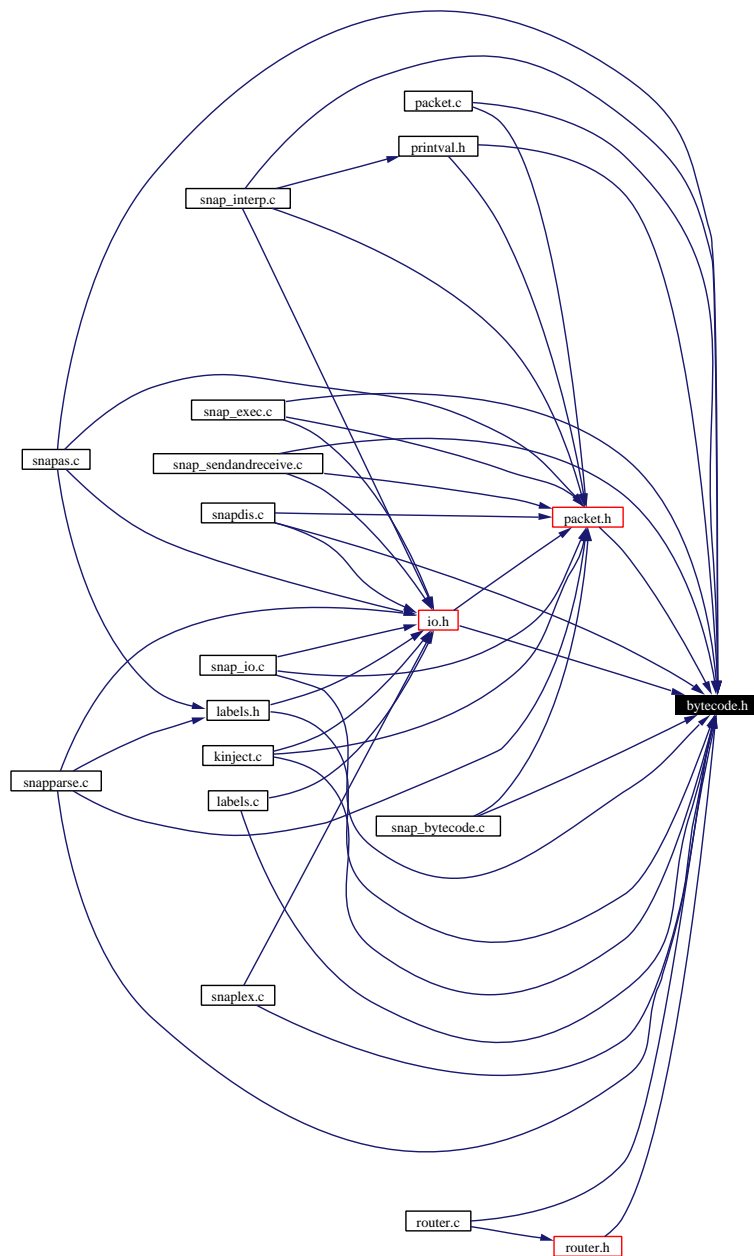
5.3 snap-1.1-wjdb/lib/bytecode.h File Reference

```
#include <stdio.h>
#include <limits.h>
#include "config.h"
#include "dyncheck.h"
#include "wassert.h"
```

Include dependency graph for bytecode.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct `heap_obj`

Defines

- #define **SMALL_INSTRS**
- #define **SMALL_VALUES**
- #define **DYNCHECK_TAG**(v, tag) ((void)0)
- #define **INTV** 0
- #define **ADDRV** 1
- #define **STRV** 2
- #define **EXCV** 3
- #define **TUPLEV** 4
- #define **FLOATV** 5
- #define **BOGUSV** 6
- #define **TAG_T** int
- #define **LENTYPE** unsigned short
- #define **MAX_HEAPOBJ_SZ** (1 << ((sizeof(unsigned short) * 8)))
- #define **ZERO_VALUE_T** 0
- #define **TAGSZ** 7
- #define **MAX_VINT** (1 << ((sizeof(unsigned int) * 8-(TAGSZ+1))))
- #define **MIN_VINT** (- MAX_VINT - 1)
- #define **GET_TAG**(v) ((v) >> (sizeof(unsigned int)*8-TAGSZ))
- #define **SET_TAG**(v, t) (((v) = (((v) << TAGSZ) >> TAGSZ) | ((t) << (sizeof(unsigned int)*8-TAGSZ))))
- #define **GET_INT**(v) (((int)((v) << TAGSZ) >> TAGSZ)
- #define **SET_INT**(v, i) (((v) = (((v) >> (sizeof(unsigned int)*8-TAGSZ) << (sizeof(unsigned int)*8-TAGSZ) | (((unsigned int)(i) << TAGSZ) >> TAGSZ))
- #define **GET_OFFS** GET_INT
- #define **SET_OFFS** SET_INT
- #define **COPY_VAL**(val1, val2) ((val1) = (val2))
- #define **GET_BOXED**(res, heap, v, t)
- #define **GET_ADDR**(res, h, v) GET_BOXED(res,h,v,uint32_t)
- #define **GET_FLOAT**(res, h, v) GET_BOXED(res,h,v,float32)
- #define **GET_ADDR_VAL**(h, v) (*((uint32_t *)(((heap_obj *)((h) + GET_OFFS(v))) → s)))
- #define **GET_FLT_VAL**(h, v) (*((float32 *)(((heap_obj *)((h) + GET_OFFS(v))) → s)))
- #define **FLTINTPAIR**(f) (int)(f),(int)(((f) - (int)(f)) * 1000000)
- #define **SET_ADDR**(val, a, p)
- #define **SET_FLOAT**(val, a, p)
- #define **IS_HEAP_VAL**(v)
- #define **EXIT** 0
- #define **PUSH** 1
- #define **POP** 2
- #define **POPI** 3
- #define **PULL** 4
- #define **STORE** 5
- #define **PAJ** 6

- #define **TPAJ** 7
- #define **JI** 8
- #define **BEZ** 9
- #define **BNE** 10
- #define **MKTUP** 11
- #define **NTH** 12
- #define **LEN** 13
- #define **ISTUP** 14
- #define **EQ** 15
- #define **EQI** 16
- #define **NEQ** 17
- #define **NEQI** 18
- #define **GT** 19
- #define **GTI** 20
- #define **GEQ** 21
- #define **GEQI** 22
- #define **LEQ** 23
- #define **LEQI** 24
- #define **LT** 25
- #define **LTI** 26
- #define **ADD** 27
- #define **ADDI** 28
- #define **SUB** 29
- #define **SUBI** 30
- #define **MULT** 31
- #define **MULTI** 32
- #define **DIV** 33
- #define **DIVI** 34
- #define **MOD** 35
- #define **MODI** 36
- #define **NEG** 37
- #define **NOT** 38
- #define **LNOT** 39
- #define **AND** 40
- #define **ANDI** 41
- #define **OR** 42
- #define **ORI** 43
- #define **LSHL** 44
- #define **LSHLI** 45
- #define **RSHL** 46
- #define **RSHLI** 47
- #define **RSHA** 48
- #define **RSHAI** 49
- #define **XOR** 50
- #define **XORI** 51
- #define **SNET** 52

- #define **SNETI** 53
- #define **BCAST** 54
- #define **BCASTI** 55
- #define **ISX** 56
- #define **GETRB** 57
- #define **GETSRC** 58
- #define **GETDST** 60
- #define **GETSPT** 61
- #define **HERE** 62
- #define **ISHERE** 63
- #define **ROUTE** 64
- #define **RTDEV** 65
- #define **SEND** 66
- #define **HOP** 67
- #define **FORW** 69
- #define **FORWTO** 70
- #define **DEMUX** 71
- #define **DEMUXI** 72
- #define **PRINT** 73
- #define **PINT** 74
- #define **PADDR** 75
- #define **PTUP** 76
- #define **PEXC** 77
- #define **PSTR** 78
- #define **PFLT** 79
- #define **EQINT** 80
- #define **EQADR** 81
- #define **EQTUP** 82
- #define **EQEXC** 83
- #define **EQSTR** 84
- #define **EQFLT** 85
- #define **NQINT** 86
- #define **NQADR** 87
- #define **NQTUP** 88
- #define **NQEXC** 89
- #define **NQSTR** 90
- #define **NQFLT** 91
- #define **SVCV** 92
- #define **CALLS** 93
- #define **FGTI** 94
- #define **FGEQI** 95
- #define **FLEQI** 96
- #define **FLTI** 97
- #define **FADDI** 98
- #define **FSUBI** 99
- #define **FMULI** 100

- #define **FDIVI** 101
- #define **GETLD** 102
- #define **SETXH** 103
- #define **RAISEX** 104
- #define **DSEND** 105
- #define **DFORW** 106
- #define **DFORWTO** 107
- #define **STACKEMPTY** 108
- #define **STACKCOUNT** 109
- #define **PULLSTACK** 110
- #define **OPCODE_T** int
- #define **GET_OP**(v) ((v) >> (sizeof(unsigned int)*8-TAGSZ))
- #define **SET_OP**(v, i) ((v) = (((v) << TAGSZ) >> TAGSZ) | ((i) << (sizeof(unsigned int)*8-TAGSZ)))
- #define **GET_LIT**(l, t, i) ((l) = (((int)((i) << TAGSZ)) >> TAGSZ))
- #define **SET_LIT**(i, t, l) ((i) = (((i) >> (sizeof(unsigned int)*8-TAGSZ)) << (sizeof(unsigned int)*8-TAGSZ)) | (((unsigned int)(l) << TAGSZ) >> TAGSZ))
- #define **GET_LIT_VAL**(i) (((int)((i) << TAGSZ)) >> TAGSZ)
- #define **GET_STR_VAL**(h, v) (((heap_obj *) (h) + GET_OFFS(v)) → s)
- #define **NUM_OPS** 87
- #define **COPY_LIT**(d, t, s)

Typedefs

- typedef unsigned int **value_t**
- typedef unsigned int **instr_t**

5.3.1 Define Documentation

5.3.1.1 #define ADD 27

Definition at line 250 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.2 #define ADDI 28

Definition at line 251 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `refine_op()`, and `yyparse()`.

5.3.1.3 #define ADDR_V 1

Definition at line 36 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_value_heap()`, `fprintf_value_tag()`, `refine_op()`, `snap_svc_convert_returnstruct2stack()`, `snap_svc_convert_stack2returnstruct()`, and `yyparse()`.

5.3.1.4 #define AND 40

Definition at line 263 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.5 #define ANDI 41

Definition at line 264 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, and `yyparse()`.

5.3.1.6 #define BCAST 54

Definition at line 277 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.7 #define BCASTI 55

Definition at line 278 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `yyparse()`.

5.3.1.8 #define BEZ 9

Definition at line 232 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `patch_jumps()`, and `yyparse()`.

5.3.1.9 #define BNE 10

Definition at line 233 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `patch_jumps()`, and `yyparse()`.

5.3.1.10 #define BOGUSV 6

Definition at line 41 of file bytecode.h.

Referenced by `marshal_packet()`.

5.3.1.11 #define CALLS 93

Definition at line 320 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `yyparse()`.

5.3.1.12 #define COPY_LIT(d, t, s)

Value:

```
{ int _lit;          \
  GET_LIT(_lit,t,(s)); \
  SET_INT((d),_lit);  \
  SET_TAG((d),t);    \
}
```

Definition at line 539 of file bytecode.h.

Referenced by `fprintf_instr()`.

5.3.1.13 #define COPY_VAL(val1, val2) ((val1) = (val2))

Definition at line 97 of file bytecode.h.

Referenced by `yyparse()`.

5.3.1.14 #define DEMUX 71

Definition at line 294 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.15 #define DEMUXI 72

Definition at line 295 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.16 #define DFORW 106

Definition at line 338 of file bytecode.h.

Referenced by `yyparse()`.

5.3.1.17 #define DFORWTO 107

Definition at line 339 of file bytecode.h.

Referenced by `yyparse()`.

5.3.1.18 #define DIV 33

Definition at line 256 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.19 #define DIVI 34

Definition at line 257 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `refine_op()`, and `yyparse()`.

5.3.1.20 #define DSEND 105

Definition at line 337 of file bytecode.h.

Referenced by `yyparse()`.

5.3.1.21 #define DYNCHECK_TAG(v, tag) ((void)0)

Definition at line 22 of file bytecode.h.

5.3.1.22 #define EQ 15

Definition at line 238 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.23 #define EQADR 81

Definition at line 306 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `patch_jumps()`, and `refine_op()`.

5.3.1.24 #define EQEXC 83

Definition at line 308 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `patch_jumps()`, and `refine_op()`.

5.3.1.25 #define EQFLT 85

Definition at line 310 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `refine_op()`.

5.3.1.26 #define EQI 16

Definition at line 239 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `refine_op()`, and `yyparse()`.

5.3.1.27 #define EQINT 80

Definition at line 305 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `patch_jumps()`, and `refine_op()`.

5.3.1.28 #define EQSTR 84

Definition at line 309 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `patch_jumps()`, and `refine_op()`.

5.3.1.29 #define EQTUP 82

Definition at line 307 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `patch_jumps()`, and `refine_op()`.

5.3.1.30 #define EXCV 3

Definition at line 38 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_value_heap()`, `fprintf_value_tag()`, `refine_op()`, `snap_svc_convert_stack2returnstruct()`, and `yyparse()`.

5.3.1.31 #define EXIT 0

Definition at line 223 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.32 #define FADDI 98

Definition at line 326 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `refine_op()`.

5.3.1.33 #define FDIVI 101

Definition at line 329 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `refine_op()`.

5.3.1.34 #define FGEQI 95

Definition at line 323 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `refine_op()`.

5.3.1.35 #define FGTI 94

Definition at line 322 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `refine_op()`.

5.3.1.36 #define FLEQI 96

Definition at line 324 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `refine_op()`.

5.3.1.37 #define FLOATV 5

Definition at line 40 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_value_heap()`, `fprintf_value_tag()`, `refine_op()`, `snap_svc_convert_stack2returnstruct()`, and `yyparse()`.

5.3.1.38 #define FLTI 97

Definition at line 325 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `refine_op()`.

5.3.1.39 #define FLTINTPAIR(f) (int)(f),(int)(((f) - (int)(f)) * 1000000)

Definition at line 115 of file bytecode.h.

5.3.1.40 #define FMULI 100

Definition at line 328 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `refine_op()`.

5.3.1.41 #define FORW 69

Definition at line 292 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.42 #define FORWTO 70

Definition at line 293 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.43 #define FSUBI 99

Definition at line 327 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `refine_op()`.

5.3.1.44 #define GEQ 21

Definition at line 244 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.45 #define GEQI 22

Definition at line 245 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `refine_op()`, and `yyparse()`.

**5.3.1.46 #define GET_ADDR(res, h, v)
GET_BOXED(res,h,v,uint32_t)**

Definition at line 107 of file bytecode.h.

**5.3.1.47 #define GET_ADDR_VAL(h, v) (((uint32_t *)(((heap_obj *)
*)((h) + GET_OFFS(v))) → s))**

Definition at line 111 of file bytecode.h.

Referenced by `fprintf_value_heap()`, and `snap_svc_convert_returnstruct2stack()`.

5.3.1.48 `#define GET_BOXED(res, heap, v, t)`

Value:

```
{
    heap_obj *ho1 = (heap_obj *)((heap) + GET_OFFS(v));
    DYNCHECK_ADDR_IN_HEAP(ho1);
    wassert(ho1->len == sizeof(t));
    (res) = *((t *)ho1->s);
}
```

Definition at line 99 of file `bytecode.h`.

5.3.1.49 `#define GET_FLOAT(res, h, v)` `GET_BOXED(res,h,v,float32)`

Definition at line 109 of file `bytecode.h`.

5.3.1.50 `#define GET_FLT_VAL(h, v) (*((float32 *)(((heap_obj *)((h) + GET_OFFS(v))) → s)))`

Definition at line 113 of file `bytecode.h`.

5.3.1.51 `#define GET_INT(v) (((int)((v) << TAGSZ)) >> TAGSZ)`

Definition at line 91 of file `bytecode.h`.

Referenced by `snap_svc_convert_returnstruct2stack()`, `snap_svc_convert_stack2returnstruct()`, and `yyparse()`.

5.3.1.52 `#define GET_LIT(l, t, i) ((l) = (((int)((i) << TAGSZ)) >> TAGSZ))`

Definition at line 353 of file `bytecode.h`.

Referenced by `marshal_packet()`.

5.3.1.53 `#define GET_LIT_VAL(i) (((int)((i) << TAGSZ)) >> TAGSZ)`

Definition at line 356 of file `bytecode.h`.

5.3.1.54 #define GET_OFFS GET_INT

Definition at line 94 of file bytecode.h.

Referenced by `snap_svc_convert_returnstruct2stack()`, and `snap_svc_convert_stack2returnstruct()`.

5.3.1.55 #define GET_OP(v) ((v) >> (sizeof(unsigned int)*8-TAGSZ))

Definition at line 350 of file bytecode.h.

Referenced by `fprintf_instr()`, `marshal_packet()`, and `patch_jumps()`.

5.3.1.56 #define GET_STR_VAL(h, v) (((heap_obj *)((h) + GET_OFFS(v))) → s)

Definition at line 358 of file bytecode.h.

5.3.1.57 #define GET_TAG(v) ((v) >> (sizeof(unsigned int)*8-TAGSZ))

Definition at line 88 of file bytecode.h.

Referenced by `fprintf_value_heap()`, `snap_svc_convert_stack2returnstruct()`, and `yyparse()`.

5.3.1.58 #define GETDST 60

Definition at line 282 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.59 #define GETLD 102

Definition at line 331 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.60 #define GETRB 57

Definition at line 280 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.61 #define GETSPT 61

Definition at line 283 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.62 `#define GETSRC 58`

Definition at line 281 of file `bytecode.h`.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.63 `#define GT 19`

Definition at line 242 of file `bytecode.h`.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.64 `#define GTI 20`

Definition at line 243 of file `bytecode.h`.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `refine_op()`, and `yyparse()`.

5.3.1.65 `#define HERE 62`

Definition at line 284 of file `bytecode.h`.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.66 `#define HOP 67`

Definition at line 289 of file `bytecode.h`.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.67 `#define INTV 0`

Definition at line 35 of file `bytecode.h`.

Referenced by `fprintf_instr()`, `fprintf_value_heap()`, `fprintf_value_tag()`, `refine_op()`, `snap_svc_convert_returnstruct2stack()`, `snap_svc_convert_stack2returnstruct()`, and `yyparse()`.

5.3.1.68 `#define IS_HEAP_VAL(v)`

Value:

```
((GET_TAG(v) == ADDR) || (GET_TAG(v) == STR) || \
 (GET_TAG(v) == TUPLE) || (GET_TAG(v) == FLOAT))
```

Definition at line 135 of file `bytecode.h`.

5.3.1.69 #define ISHERE 63

Definition at line 285 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.70 #define ISTUP 14

Definition at line 237 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.71 #define ISX 56

Definition at line 279 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.72 #define JI 8

Definition at line 231 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `patch_jumps()`, and `yyparse()`.

5.3.1.73 #define LEN 13

Definition at line 236 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.74 #define LENTYPE unsigned short

Definition at line 63 of file bytecode.h.

5.3.1.75 #define LEQ 23

Definition at line 246 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.76 #define LEQI 24

Definition at line 247 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `refine_op()`, and `yyparse()`.

5.3.1.77 #define LNOT 39

Definition at line 262 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.78 #define LSHL 44

Definition at line 267 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.79 #define LSHLI 45

Definition at line 268 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, and `yyparse()`.

5.3.1.80 #define LT 25

Definition at line 248 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.81 #define LTI 26

Definition at line 249 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `refine_op()`, and `yyparse()`.

5.3.1.82 #define MAX_HEAPOBJ_SZ (1 << ((sizeof(unsigned short) * 8)))

Definition at line 64 of file bytecode.h.

Referenced by `newwho()`, and `newtup()`.

5.3.1.83 #define MAX_VINT (1 << ((sizeof(unsigned int) * 8-(TAGSZ+1))))

Definition at line 85 of file bytecode.h.

Referenced by `yyparse()`.

5.3.1.84 #define MIN_VINT (- MAX_VINT - 1)

Definition at line 86 of file bytecode.h.

Referenced by `yyparse()`.

5.3.1.85 #define MKTUP 11

Definition at line 234 of file bytecode.h.

Referenced by fprintf_instr(), fprintf_opcode(), and yyparse().

5.3.1.86 #define MOD 35

Definition at line 258 of file bytecode.h.

Referenced by fprintf_opcode(), and yyparse().

5.3.1.87 #define MODI 36

Definition at line 259 of file bytecode.h.

Referenced by fprintf_instr(), fprintf_opcode(), and yyparse().

5.3.1.88 #define MULT 31

Definition at line 254 of file bytecode.h.

Referenced by fprintf_opcode(), and yyparse().

5.3.1.89 #define MULTI 32

Definition at line 255 of file bytecode.h.

Referenced by fprintf_instr(), fprintf_opcode(), marshal_packet(), refine_op(), and yyparse().

5.3.1.90 #define NEG 37

Definition at line 260 of file bytecode.h.

Referenced by fprintf_opcode(), and yyparse().

5.3.1.91 #define NEQ 17

Definition at line 240 of file bytecode.h.

Referenced by fprintf_opcode(), and yyparse().

5.3.1.92 #define NEQI 18

Definition at line 241 of file bytecode.h.

Referenced by fprintf_instr(), fprintf_opcode(), marshal_packet(), refine_op(), and yyparse().

5.3.1.93 #define NOT 38

Definition at line 261 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.94 #define NQADR 87

Definition at line 313 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `patch_jumps()`, and `refine_op()`.

5.3.1.95 #define NQEXC 89

Definition at line 315 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `patch_jumps()`, and `refine_op()`.

5.3.1.96 #define NQFLT 91

Definition at line 317 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `refine_op()`.

5.3.1.97 #define NQINT 86

Definition at line 312 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `patch_jumps()`, and `refine_op()`.

5.3.1.98 #define NQSTR 90

Definition at line 316 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `patch_jumps()`, and `refine_op()`.

5.3.1.99 #define NQTUP 88

Definition at line 314 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `patch_jumps()`, and `refine_op()`.

5.3.1.100 #define NTH 12

Definition at line 235 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, and `yyparse()`.

5.3.1.101 `#define NUM_OPS 87`

Definition at line 360 of file `bytecode.h`.

5.3.1.102 `#define OPCODE_T int`

Definition at line 345 of file `bytecode.h`.

Referenced by `fprintf_opcode()`, `refine_op()`, and `yyparse()`.

5.3.1.103 `#define OR 42`

Definition at line 265 of file `bytecode.h`.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.104 `#define ORI 43`

Definition at line 266 of file `bytecode.h`.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, and `yyparse()`.

5.3.1.105 `#define PADDR 75`

Definition at line 299 of file `bytecode.h`.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `patch_jumps()`, and `refine_op()`.

5.3.1.106 `#define PAJ 6`

Definition at line 229 of file `bytecode.h`.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, and `yyparse()`.

5.3.1.107 `#define PEXC 77`

Definition at line 301 of file `bytecode.h`.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `patch_jumps()`, and `refine_op()`.

5.3.1.108 `#define PFLT 79`

Definition at line 303 of file `bytecode.h`.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `refine_op()`.

5.3.1.109 #define PINT 74

Definition at line 298 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `patch_jumps()`, and `refine_op()`.

5.3.1.110 #define POP 2

Definition at line 225 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.111 #define POPI 3

Definition at line 226 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, and `yyparse()`.

5.3.1.112 #define PRINT 73

Definition at line 296 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.113 #define PSTR 78

Definition at line 302 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `patch_jumps()`, and `refine_op()`.

5.3.1.114 #define PTUP 76

Definition at line 300 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `patch_jumps()`, and `refine_op()`.

5.3.1.115 #define PULL 4

Definition at line 227 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, and `yyparse()`.

5.3.1.116 #define PULLSTACK 110

Definition at line 344 of file bytecode.h.

Referenced by `yyparse()`.

5.3.1.117 #define PUSH 1

Definition at line 224 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `patch_jumps()`, `refine_op()`, and `yyparse()`.

5.3.1.118 #define RAISEX 104

Definition at line 333 of file bytecode.h.

Referenced by `yyparse()`.

5.3.1.119 #define ROUTE 64

Definition at line 286 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.120 #define RSHA 48

Definition at line 271 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.121 #define RSHAI 49

Definition at line 272 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, and `yyparse()`.

5.3.1.122 #define RSHL 46

Definition at line 269 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.123 #define RSHLI 47

Definition at line 270 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, and `yyparse()`.

5.3.1.124 #define RTDEV 65

Definition at line 287 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.125 #define SEND 66

Definition at line 288 of file bytecode.h.

Referenced by fprintf_opcode(), and yyparse().

5.3.1.126 #define SET_ADDR(val, a, p)

Value:

```
{ int hoffset;
  heap_obj *ho;
  if (!heap_alloc((p),sizeof(uint32_t),0,&ho,&hoffset)) {
    *((uint32_t *)ho->s) = (a);
    SET_OFFS(val,hoffset);
  }
  else return -1;
}
```

Definition at line 117 of file bytecode.h.

Referenced by snap_svc_convert_returnstruct2stack(), and yyparse().

5.3.1.127 #define SET_FLOAT(val, a, p)

Value:

```
{ int hoffset;
  heap_obj *ho;
  if (!heap_alloc((p),sizeof(float32),0,&ho,&hoffset)) {\
    *((float32 *)ho->s) = (a);
    SET_OFFS(val,hoffset);
  }
  else return -1;
}
```

Definition at line 126 of file bytecode.h.

Referenced by yyparse().

5.3.1.128 #define SET_INT(v, i) (((v) = (((v) >> (sizeof(unsigned int)*8-TAGSZ)) << (sizeof(unsigned int)*8-TAGSZ)) | (((unsigned int)(i)) << TAGSZ) >> TAGSZ))

Definition at line 92 of file bytecode.h.

Referenced by snap_svc_convert_returnstruct2stack(), and yyparse().

5.3.1.129 #define SET_LIT(i, t, l) (((i) = (((i) >> (sizeof(unsigned int)*8-TAGSZ)) << (sizeof(unsigned int)*8-TAGSZ)) | (((unsigned int)(l)) << TAGSZ) >> TAGSZ))

Definition at line 354 of file bytecode.h.

Referenced by `marshal_packet()`, and `yyparse()`.

5.3.1.130 `#define SET_OFFS SET_INT`

Definition at line 95 of file `bytecode.h`.

Referenced by `snap_svc_convert_returnstruct2stack()`, and `yyparse()`.

5.3.1.131 `#define SET_OP(v, i) ((v) = (((v) << TAGSZ) >> TAGSZ) | ((i) << (sizeof(unsigned int)*8-TAGSZ)))`

Definition at line 351 of file `bytecode.h`.

Referenced by `yyparse()`.

5.3.1.132 `#define SET_TAG(v, t) ((v) = (((v) << TAGSZ) >> TAGSZ) | ((t) << (sizeof(unsigned int)*8-TAGSZ)))`

Definition at line 89 of file `bytecode.h`.

Referenced by `snap_svc_convert_returnstruct2stack()`, and `yyparse()`.

5.3.1.133 `#define SETXH 103`

Definition at line 332 of file `bytecode.h`.

Referenced by `yyparse()`.

5.3.1.134 `#define SMALL_INSTRS`

Definition at line 13 of file `bytecode.h`.

5.3.1.135 `#define SMALL_VALUES`

Definition at line 16 of file `bytecode.h`.

5.3.1.136 `#define SNET 52`

Definition at line 275 of file `bytecode.h`.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.137 `#define SNETI 53`

Definition at line 276 of file `bytecode.h`.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `yyparse()`.

5.3.1.138 #define STACKCOUNT 109

Definition at line 342 of file bytecode.h.

Referenced by `yyparse()`.

5.3.1.139 #define STACKEMPTY 108

Definition at line 341 of file bytecode.h.

Referenced by `yyparse()`.

5.3.1.140 #define STORE 5

Definition at line 228 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, and `yyparse()`.

5.3.1.141 #define STRV 2

Definition at line 37 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_value_heap()`, `fprintf_value_tag()`, `refine_op()`, `snap_svc_convert_returnstruct2stack()`, `snap_svc_convert_stack2returnstruct()`, and `yyparse()`.

5.3.1.142 #define SUB 29

Definition at line 252 of file bytecode.h.

Referenced by `fprintf_opcode()`, and `yyparse()`.

5.3.1.143 #define SUBI 30

Definition at line 253 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, `refine_op()`, and `yyparse()`.

5.3.1.144 #define SVCV 92

Definition at line 319 of file bytecode.h.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, `marshal_packet()`, and `yyparse()`.

5.3.1.145 #define TAG_T int

Definition at line 42 of file bytecode.h.

Referenced by `fprintf_value_tag()`, and `refine_op()`.

5.3.1.146 `#define TAGSZ 7`

Definition at line 80 of file `bytecode.h`.

5.3.1.147 `#define TPAJ 7`

Definition at line 230 of file `bytecode.h`.

Referenced by `fprintf_instr()`, `fprintf_opcode()`, and `yyparse()`.

5.3.1.148 `#define TUPLEV 4`

Definition at line 39 of file `bytecode.h`.

Referenced by `fprintf_instr()`, `fprintf_value_heap()`, `fprintf_value_tag()`, `refine_op()`, `snap_svc_convert_stack2returnstruct()`, and `yyparse()`.

5.3.1.149 `#define XOR 50`

Definition at line 273 of file `bytecode.h`.

Referenced by `fprintf_opcode()`.

5.3.1.150 `#define XORI 51`

Definition at line 274 of file `bytecode.h`.

Referenced by `fprintf_instr()`, and `fprintf_opcode()`.

5.3.1.151 `#define ZERO_VALUE_T 0`

Definition at line 77 of file `bytecode.h`.

5.3.2 Typedef Documentation

5.3.2.1 `typedef unsigned int instr_t`

Definition at line 362 of file `bytecode.h`.

Referenced by `fprintf_instr()`, `fprintf_packet()`, `main()`, `marshal_packet()`, `patch_jumps()`, and `unmarshal_packet()`.

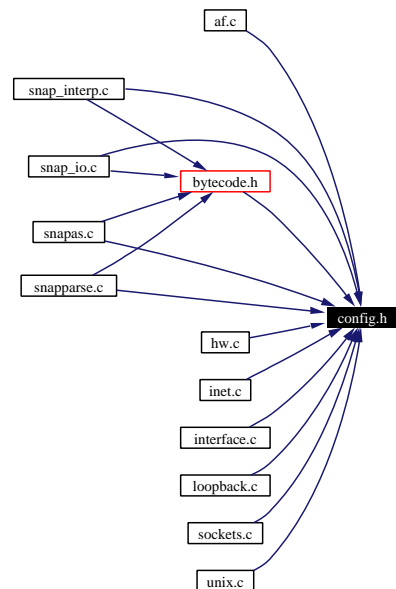
5.3.2.2 `typedef unsigned int value_t`

Definition at line 76 of file `bytecode.h`.

Referenced by `fprintf_instr()`, `fprintf_packet()`, `fprintf_value()`, `fprintf_value-heap()`, `main()`, `marshal_packet()`, `newtup()`, `patch_jumps()`, `snap_svc-convert_returnstruct2stack()`, `snap_svc_convert_stack2returnstruct()`, `unmarshal_packet()`, and `yyparse()`.

5.4 snap-1.1-wjdb/lib/config.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define CONFIG_IP_SNAP_SMALL_INSTRS`
- `#define CONFIG_IP_SNAP_SMALL_VALUES`
- `#define IS_LITTLE_ENDIAN`

Typedefs

- `typedef float float32`
- `typedef double float64`
- `typedef int int32`
- `typedef unsigned int uint32`

5.4.1 Define Documentation

5.4.1.1 `#define CONFIG_IP_SNAP_SMALL_INSTRS`

Definition at line 6 of file config.h.

5.4.1.2 `#define CONFIG_IP_SNAP_SMALL_VALUES`

Definition at line 7 of file config.h.

5.4.1.3 `#define IS_LITTLE_ENDIAN`

Definition at line 9 of file config.h.

5.4.2 Typedef Documentation

5.4.2.1 `typedef float float32`

Definition at line 11 of file config.h.

Referenced by `fprintf_value_heap()`.

5.4.2.2 `typedef double float64`

Definition at line 12 of file config.h.

5.4.2.3 `typedef int int32`

Definition at line 13 of file config.h.

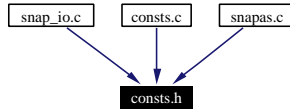
5.4.2.4 `typedef unsigned int uint32`

Definition at line 14 of file config.h.

Referenced by `unmarshal_packet()`.

5.5 snap-1.1-wjdb/lib/consts.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define DEFAULT_HEAP_SIZEB (1024 * 1024)`
- `#define DEFAULT_STACK_SIZEB (1024 * 1024)`
- `#define DEFAULT_CODE_SIZEB (1024 * 1024)`
- `#define DEFAULT_SVC_HEAP_SIZEB (50 * 1024)`

Variables

- `int heap_sizeb`
- `int stack_sizeb`
- `int code_sizeb`

5.5.1 Define Documentation

5.5.1.1 `#define DEFAULT_CODE_SIZEB (1024 * 1024)`

Definition at line 8 of file consts.h.

5.5.1.2 `#define DEFAULT_HEAP_SIZEB (1024 * 1024)`

Definition at line 6 of file consts.h.

5.5.1.3 `#define DEFAULT_STACK_SIZEB (1024 * 1024)`

Definition at line 7 of file consts.h.

5.5.1.4 `#define DEFAULT_SVC_HEAP_SIZEB (50 * 1024)`

Definition at line 10 of file consts.h.

5.5.2 Variable Documentation

5.5.2.1 `int code_sizeb`

Definition at line 14 of file `consts.h`.

5.5.2.2 `int heap_sizeb`

Definition at line 12 of file `consts.h`.

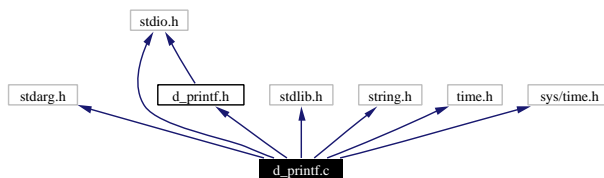
5.5.2.3 `int stack_sizeb`

Definition at line 13 of file `consts.h`.

5.6 snap-1.1-wjdb/lib/d_printf.c File Reference

```
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys/time.h>
#include "d_printf.h"
```

Include dependency graph for d_printf.c:



Functions

- void `set_debug_level_int` (int newdebuglvl)
- void `set_debug_level` (void)
- void `d_printf` (int lvl, char *fmt,...)
- void `d_printf_timed` (int lvl, char *fmt,...)

Variables

- int `debug_level` = 0
- int `debug_level_setp` = 0

5.6.1 Function Documentation

5.6.1.1 void d_printf (int *lvl*, char * *fmt*, ...)

Definition at line 49 of file d_printf.c.

5.6.1.2 void d_printf_timed (int *lvl*, char * *fmt*, ...)

Definition at line 75 of file d_printf.c.

5.6.1.3 void set_debug_level (void)

Definition at line 45 of file d_printf.c.

Referenced by d_printf(), d_printf_timed(), and snap().

5.6.1.4 void set_debug_level_int (int *newdebuglvl*)

Definition at line 30 of file d_printf.c.

5.6.2 Variable Documentation**5.6.2.1 int debug_level = 0**

Definition at line 27 of file d_printf.c.

Referenced by set_debug_level_int().

5.6.2.2 int debug_level_setp = 0

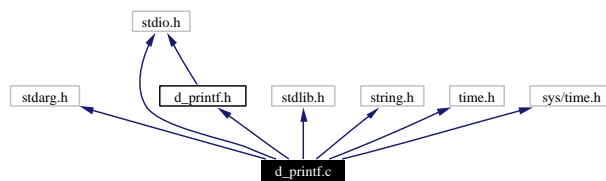
Definition at line 28 of file d_printf.c.

Referenced by d_printf(), d_printf_timed(), and set_debug_level_int().

5.7 snap_svc/d_printf.c File Reference

```
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys/time.h>
#include "d_printf.h"
```

Include dependency graph for d_printf.c:



Functions

- void **set_debug_level_int** (int newdebuglvl)
- void **set_debug_level** (void)
- void **d_printf** (int lvl, char *fmt,...)
- void **d_printf_timed** (int lvl, char *fmt,...)

Variables

- int **debug_level** = 0
- int **debug_level_setp** = 0

5.7.1 Function Documentation

5.7.1.1 void d_printf (int lvl, char * fmt, ...)

Definition at line 49 of file d_printf.c.

References `debug_level_setp`, and `set_debug_level()`.

Referenced by `fini()`, `handle_request()`, `ht_insert()`, `ht_lookup()`, `if_get_interface_count()`, `if_get_interface_name()`, `if_getallneighbours()`, `if_gethopfromiface()`, `if_getiface()`, `if_getifaceidx()`, `if_getnextiface()`, `if_getoutiface()`, `if_setiface()`, `init()`, `init_request()`, `main()`, `marshal_packet()`, `newwho()`, `newtup()`, `parse_cmdline_snap()`, `printip()`, `proc_sysnetip_getforwarding()`, `proc_sysnetip_setforwarding()`, `read_ifaces()`, `read_routes()`, `snap()`, `snap_demux_init()`,

snap_demux_receivefrom(), snap_external_svclib_done(), snap_external_svclib_init(), snap_external_svclib_snmp_addvar_null(), snap_external_svclib_snmp_addvar_withvalue(), snap_external_svclib_snmp_close(), snap_external_svclib_snmp_execeptdu(), snap_external_svclib_snmp_getallotherneighboursfromip(), snap_external_svclib_snmp_gethop(), snap_external_svclib_snmp_getiface(), snap_external_svclib_snmp_getifnumber(), snap_external_svclib_snmp_init(), snap_external_svclib_snmp_init_ip(), snap_external_svclib_snmp_initpdu(), snap_external_svclib_snmp_INTERNAL_execeptdu_handler(), snap_external_svclib_snmp_isupiface(), snap_receive(), snap_svc_bind(), snap_svc_close(), snap_svc_convert_returnstruct2stack(), snap_svc_convert_stack2arguments(), snap_svc_convert_stack2returnstruct(), snap_svc_ifip_init(), snap_svc_logerrors(), snap_svc_openmultiple(), snap_svc_openmultiple_selector_snapsvc(), snap_svc_register_fini(), snap_svc_registerall(), snap_svc_registeralllibs(), snap_svc_registerlib(), snap_svc_table_add(), snap_svc_table_init(), snap_svc_unregisteralllibs(), snap_svc_unregisterlib(), and unmarshal_packet().

5.7.1.2 void d_printf_timed (int lvl, char * fmt, ...)

Definition at line 75 of file d_printf.c.

References debug_level_setp, and set_debug_level().

Referenced by handle_snap_request(), snap_demux_receive(), snap_demux_receivefrom(), snap_demux_select(), snap_external_svclib_snmp_execeptdu(), and snap_receive().

5.7.1.3 void set_debug_level (void)

Definition at line 45 of file d_printf.c.

References set_debug_level_int().

5.7.1.4 void set_debug_level_int (int newdebuglvl)

Definition at line 30 of file d_printf.c.

References debug_level, and debug_level_setp.

Referenced by parse_cmdline_snap(), and set_debug_level().

5.7.2 Variable Documentation

5.7.2.1 int debug_level = 0

Definition at line 27 of file d_printf.c.

Referenced by set_debug_level_int().

5.7.2.2 int debug_level_setp = 0

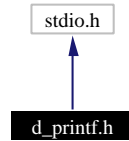
Definition at line 28 of file d_printf.c.

Referenced by d_printf(), d_printf_timed(), and set_debug_level_int().

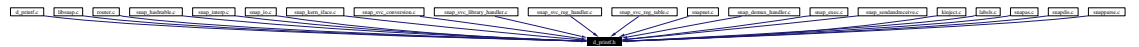
5.8 snap-1.1-wjdb/lib/d_printf.h File Reference

```
#include <stdio.h>
```

Include dependency graph for d_printf.h:



This graph shows which files directly or indirectly include this file:



Functions

- void **set_debug_level** (void)
- void **set_debug_level_int** (int)
- void **d_printf** (int, char *,...)
- void **d_printf_timed** (int, char *,...)

Variables

- int **sysctl_snap_debug_level**

5.8.1 Function Documentation

5.8.1.1 void d_printf (int, char *, ...)

Definition at line 49 of file d_printf.c.

5.8.1.2 void d_printf_timed (int, char *, ...)

Definition at line 75 of file d_printf.c.

5.8.1.3 void set_debug_level (void)

Definition at line 45 of file d_printf.c.

5.8.1.4 void set_debug_level(int (int)

Definition at line 30 of file d_printf.c.

5.8.2 Variable Documentation

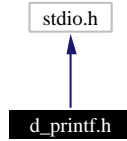
5.8.2.1 int sysctl_snap_debug_level

Definition at line 22 of file d_printf.h.

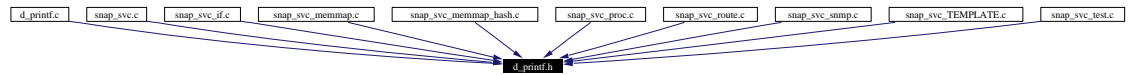
5.9 snap_svc/d_printf.h File Reference

```
#include <stdio.h>
```

Include dependency graph for d_printf.h:



This graph shows which files directly or indirectly include this file:



Functions

- void **set_debug_level** (void)
- void **set_debug_level_int** (int)
- void **d_printf** (int, char *,...)
- void **d_printf_timed** (int, char *,...)

Variables

- int **sysctl_snap_debug_level**

5.9.1 Function Documentation

5.9.1.1 void d_printf (int, char *, ...)

Definition at line 49 of file d_printf.c.

References `debug_level_setp`, and `set_debug_level()`.

Referenced by `fini()`, `handle_request()`, `ht_insert()`, `ht_lookup()`, `if_get_interface_count()`, `if_get_interface_name()`, `if_getallneighbours()`, `if_gethopfromiface()`, `if_getiface()`, `if_getifaceidx()`, `if_getnextiface()`, `if_getoutiface()`, `if_setiface()`, `init()`, `init_request()`, `main()`, `marshal_packet()`, `newwho()`, `newtup()`, `parse_cmdline_snap()`, `printip()`, `proc_sysnetip_getforwarding()`, `proc_sysnetip_setforwarding()`, `read_ifaces()`, `read_routes()`, `snap()`, `snap_demux_init()`,

snap_demux_receivefrom(), snap_external_svclib_done(), snap_external_svclib_init(), snap_external_svclib_snmp_addvar_null(), snap_external_svclib_snmp_addvar_withvalue(), snap_external_svclib_snmp_close(), snap_external_svclib_snmp_execpdu(), snap_external_svclib_snmp_getallotherneighboursfromip(), snap_external_svclib_snmp_gethop(), snap_external_svclib_snmp_getiface(), snap_external_svclib_snmp_getifnumber(), snap_external_svclib_snmp_init(), snap_external_svclib_snmp_init_ip(), snap_external_svclib_snmp_initpdu(), snap_external_svclib_snmp_INTERNAL_execpdu_handler(), snap_external_svclib_snmp_isupiface(), snap_receive(), snap_svc_bind(), snap_svc_close(), snap_svc_convert_returnstruct2stack(), snap_svc_convert_stack2arguments(), snap_svc_convert_stack2returnstruct(), snap_svc_ifip_init(), snap_svc_logerrors(), snap_svc_openmultiple(), snap_svc_openmultiple_selector_snapsvc(), snap_svc_register_fini(), snap_svc_registerall(), snap_svc_registeralllibs(), snap_svc_registerlib(), snap_svc_table_add(), snap_svc_table_init(), snap_svc_unregisteralllibs(), snap_svc_unregisterlib(), and unmarshal_packet().

5.9.1.2 void d_printf_timed (int, char *, ...)

Definition at line 75 of file d_printf.c.

References debug_level_setp, and set_debug_level().

Referenced by handle_snap_request(), snap_demux_receive(), snap_demux_receivefrom(), snap_demux_select(), snap_external_svclib_snmp_execpdu(), and snap_receive().

5.9.1.3 void set_debug_level (void)

Definition at line 45 of file d_printf.c.

References set_debug_level_int().

5.9.1.4 void set_debug_level_int (int)

Definition at line 30 of file d_printf.c.

References debug_level, and debug_level_setp.

Referenced by parse_cmdline_snap(), and set_debug_level().

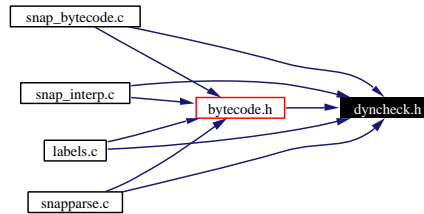
5.9.2 Variable Documentation

5.9.2.1 int sysctl_snap_debug_level

Definition at line 22 of file d_printf.h.

5.10 snap-1.1-wjdb/lib/dyncheck.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define DYNCHECK_RET(expr, ret v)`
- `#define DYNCHECK(expr) DYNCHECK_RET(expr,-1)`
- `#define DYNCHECK_IN_HEAP(ho)`
- `#define DYNCHECK_ADDR_IN_HEAP(ho)`

5.10.1 Define Documentation

5.10.1.1 `#define DYNCHECK(expr) DYNCHECK_RET(expr,-1)`

Definition at line 35 of file dyncheck.h.

5.10.1.2 `#define DYNCHECK_ADDR_IN_HEAP(ho)`

Value:

```

DYNCHECK((((void *) (ho) >= p->heap_min) &&          \
          ((void *) (ho) < p->heap_max) &&          \
          (((void *) (ho) + sizeof(uint32) +        \
            sizeof(heap_obj)) <= p->heap_max)) ||    \
          (((void *) (ho) >= (void *) p->stack_max) && \
          ((void *) (ho) < p->h_alloc_heap_max) &&   \
          (((void *) (ho) + sizeof(uint32) +        \
            sizeof(heap_obj)) <= p->h_alloc_heap_max)))
  
```

Definition at line 52 of file dyncheck.h.

5.10.1.3 `#define DYNCHECK_IN_HEAP(ho)`

Value:

```

DYNCHECK(((void *) (ho) >= p->heap_min) &&          \
          ((void *) (ho) < p->heap_max) &&          \
          (((void *) (ho) +                          \
            ((ho)->len + sizeof(heap_obj))) <= p->heap_max) || \
          ((void *) (ho) >= (void *) p->stack_max) && \
          ((void *) (ho) < p->h_alloc_heap_max) &&   \
          (((void *) (ho) +                          \
            ((ho)->len + sizeof(heap_obj))) <= p->h_alloc_heap_max)))

```

Definition at line 42 of file dyncheck.h.

5.10.1.4 #define DYNCHECK_RET(expr, retv)

Value:

```

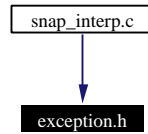
{ if (!(expr)) {                                     \
    fprintf(stderr, "%s:%d: dynamic check '%s' failed, dropping packet\n", \
              __FILE__, __LINE__, __STRING(expr)); \
    fflush(stderr);                                  \
    return (retv);                                   \
  }                                                  \
}

```

Definition at line 15 of file dyncheck.h.

5.11 snap-1.1-wjdb/lib/exception.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define RESERVED_BOUND 31`
- `#define E_NOT_ENOUGH_RB (1+RESERVED_BOUND)`
- `#define E_NON_POSITIVE_RB (2+RESERVED_BOUND)`
- `#define E_NO_ROUTE (3+RESERVED_BOUND)`
- `#define E_SERVICE_NOT_PRESENT (4+RESERVED_BOUND)`
- `#define E_SERVICE_ERROR (5+RESERVED_BOUND)`

5.11.1 Define Documentation

5.11.1.1 `#define E_NO_ROUTE (3+RESERVED_BOUND)`

Definition at line 14 of file exception.h.

5.11.1.2 `#define E_NON_POSITIVE_RB (2+RESERVED_BOUND)`

Definition at line 13 of file exception.h.

5.11.1.3 `#define E_NOT_ENOUGH_RB (1+RESERVED_BOUND)`

Definition at line 12 of file exception.h.

5.11.1.4 `#define E_SERVICE_ERROR (5+RESERVED_BOUND)`

Definition at line 16 of file exception.h.

5.11.1.5 `#define E_SERVICE_NOT_PRESENT (4+RESERVED_BOUND)`

Definition at line 15 of file exception.h.

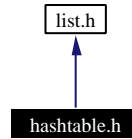
5.11.1.6 `#define RESERVED_BOUND 31`

Definition at line 9 of file exception.h.

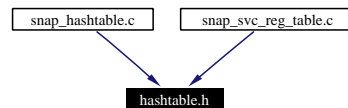
5.12 snap-1.1-wjdb/lib/hashtable.h File Reference

```
#include "list.h"
```

Include dependency graph for hashtable.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct `hash_table_t`
- struct `pair_t`

Functions

- `hash_table_t * ht_create` (int sz, int(*cmp)(const void *, const void *), int(*hash)(void *))
- void `ht_insert` (`hash_table_t *t`, void *key, void *val)
- void * `ht_lookup` (`hash_table_t *t`, void *key)
- void `ht_remove` (`hash_table_t *t`, void *key)
- int `hash_string` (char *s)

Variables

- int `ht_errno`

5.12.1 Function Documentation

5.12.1.1 int hash_string (char * s)

Definition at line 25 of file `snap_hashtable.c`.

5.12.1.2 `hash_table_t* ht_create (int sz, int(* cmp)(const void *, const void *), int(* hash)(void *))`

Definition at line 46 of file snap_hashtable.c.

5.12.1.3 `void ht_insert (hash_table_t * t, void * key, void * val)`

Definition at line 78 of file snap_hashtable.c.

5.12.1.4 `void* ht_lookup (hash_table_t * t, void * key)`

Definition at line 121 of file snap_hashtable.c.

5.12.1.5 `void ht_remove (hash_table_t * t, void * key)`

Definition at line 134 of file snap_hashtable.c.

5.12.2 Variable Documentation

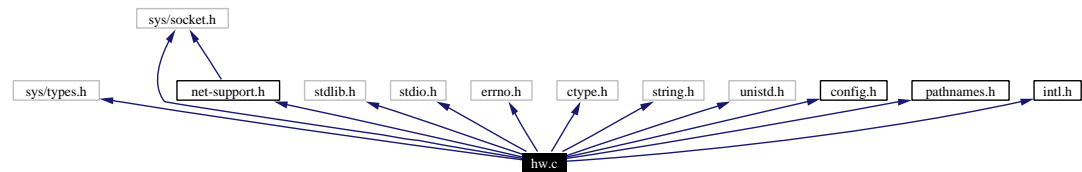
5.12.2.1 `int ht_errno`

Definition at line 24 of file hashtable.h.

5.13 snap-1.1-wjdb/lib/hw.c File Reference

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <ctype.h>
#include <string.h>
#include <unistd.h>
#include "config.h"
#include "net-support.h"
#include "pathnames.h"
#include "intl.h"
```

Include dependency graph for hw.c:



Functions

- void **hwinit** ()
- **hwtype** * **get_hwtype** (const char *name)
- **hwtype** * **get_hwntype** (int type)

Variables

- **hwtype** **unspec_hwtype**
- **hwtype** **loop_hwtype**
- **hwtype** **slip_hwtype**
- **hwtype** **cslip_hwtype**
- **hwtype** **slip6_hwtype**
- **hwtype** **cslip6_hwtype**
- **hwtype** **adaptive_hwtype**
- **hwtype** **ether_hwtype**
- **hwtype** **fddi_hwtype**

- hwtype hippi_hwtype
- hwtype tr_hwtype
- hwtype ax25_hwtype
- hwtype rose_hwtype
- hwtype netrom_hwtype
- hwtype tunnel_hwtype
- hwtype ash_hwtype
- hwtype ppp_hwtype
- hwtype arcnet_hwtype
- hwtype dlc_i_hwtype
- hwtype frad_hwtype
- hwtype hdlc_hwtype
- hwtype lapb_hwtype
- hwtype sit_hwtype

5.13.1 Function Documentation

5.13.1.1 struct hwtype* get_hwntype (int *type*)

Definition at line 202 of file hw.c.

References hwinit().

5.13.1.2 struct hwtype* get_hwtype (const char * *name*)

Definition at line 184 of file hw.c.

References hwinit().

5.13.1.3 void hwinit ()

Definition at line 127 of file hw.c.

References `_`, and hwtype::title.

Referenced by get_hwntype(), and get_hwtype().

5.13.2 Variable Documentation

5.13.2.1 struct hwtype adaptive_hwtype

Definition at line 40 of file hw.c.

5.13.2.2 struct hwtype arcnet_hwtype

Definition at line 56 of file hw.c.

5.13.2.3 struct hwtype ash_hwtype

Definition at line 52 of file hw.c.

5.13.2.4 struct hwtype ax25_hwtype

Definition at line 47 of file hw.c.

5.13.2.5 struct hwtype cslip6_hwtype

Definition at line 39 of file hw.c.

5.13.2.6 struct hwtype cslip_hwtype

Definition at line 37 of file hw.c.

5.13.2.7 struct hwtype dlc_hwtype

Definition at line 58 of file hw.c.

5.13.2.8 struct hwtype ether_hwtype

Definition at line 42 of file hw.c.

5.13.2.9 struct hwtype fddi_hwtype

Definition at line 43 of file hw.c.

5.13.2.10 struct hwtype frad_hwtype

Definition at line 59 of file hw.c.

5.13.2.11 struct hwtype hdlc_hwtype

Definition at line 61 of file hw.c.

5.13.2.12 struct hwtype hippi_hwtype

Definition at line 44 of file hw.c.

5.13.2.13 struct hwtype lapb_hwtype

Definition at line 62 of file hw.c.

5.13.2.14 struct hwtype loop_hwtype

Definition at line 34 of file hw.c.

5.13.2.15 struct hwtype netrom_hwtype

Definition at line 49 of file hw.c.

5.13.2.16 struct hwtype ppp_hwtype

Definition at line 54 of file hw.c.

5.13.2.17 struct hwtype rose_hwtype

Definition at line 48 of file hw.c.

5.13.2.18 struct hwtype sit_hwtype

Definition at line 64 of file hw.c.

5.13.2.19 struct hwtype slip6_hwtype

Definition at line 38 of file hw.c.

5.13.2.20 struct hwtype slip_hwtype

Definition at line 36 of file hw.c.

5.13.2.21 struct hwtype tr_hwtype

Definition at line 45 of file hw.c.

5.13.2.22 struct hwtype tunnel_hwtype

Definition at line 50 of file hw.c.

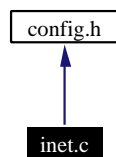
5.13.2.23 struct hwtype unspec_hwtype

Definition at line 33 of file hw.c.

5.14 snap-1.1-wjdb/lib/inet.c File Reference

```
#include "config.h"
```

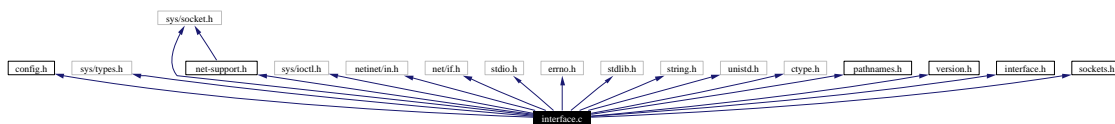
Include dependency graph for inet.c:



5.15 snap-1.1-wjdb/lib/interface.c File Reference

```
#include "config.h"
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <netinet/in.h>
#include <net/if.h>
#include <stdio.h>
#include <errno.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <ctype.h>
#include "net-support.h"
#include "pathnames.h"
#include "version.h"
#include "interface.h"
#include "sockets.h"
```

Include dependency graph for interface.c:



Functions

- `int if_fetch (char *ifname, struct interface *ife)`

Variables

- `int procnetwork_vsn = 1`

5.15.1 Function Documentation

5.15.1.1 `int if_fetch (char * ifname, struct interface * ife)`

Definition at line 145 of file `interface.c`.

References `interface::addr`, `interface::broadcast`, `ddp_sock`, `interface::ddpaddr`, `interface::dstaddr`, `ec_sock`, `interface::ecaddr`, `interface::flags`, `interface::has_ddp`, `interface::has_econet`, `interface::has_ipx_bb`, `interface::has_ipx_e2`, `interface::has_ipx_e3`, `interface::has_ipx_sn`, `interface::hwaddr`, `inet_sock`, `ipx_sock`, `interface::ipxaddr_bb`, `interface::ipxaddr_e2`, `interface::ipxaddr_e3`, `interface::ipxaddr_sn`, `interface::map`, `interface::metric`, `interface::mtu`, `interface::name`, `interface::netmask`, `interface::tx_queue_len`, and `interface::type`.

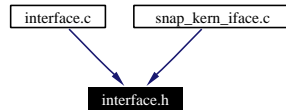
5.15.2 Variable Documentation

5.15.2.1 `int procnetsdev_vsn = 1`

Definition at line 37 of file `interface.c`.

5.16 snap-1.1-wjdb/lib/interface.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **interface**
- struct **user_net_device_stats**

Functions

- int **if_fetch** (char *ifname, struct **interface** *ife)

Variables

- int **procnetdev_vsn**

5.16.1 Function Documentation

5.16.1.1 int if_fetch (char * ifname, struct interface * ife)

Definition at line 145 of file interface.c.

References interface::addr, interface::broadaddr, ddp_sock, interface::ddpaddr, interface::dstaddr, ec_sock, interface::ecaddr, interface::flags, interface::has_ddp, interface::has_econet, interface::has_ipx_bb, interface::has_ipx_e2, interface::has_ipx_e3, interface::has_ipx_sn, interface::hwaddr, inet_sock, ipx_sock, interface::ipxaddr_bb, interface::ipxaddr_e2, interface::ipxaddr_e3, interface::ipxaddr_sn, interface::map, interface::metric, interface::mtu, interface::name, interface::netmask, interface::tx_queue_len, and interface::type.

5.16.2 Variable Documentation

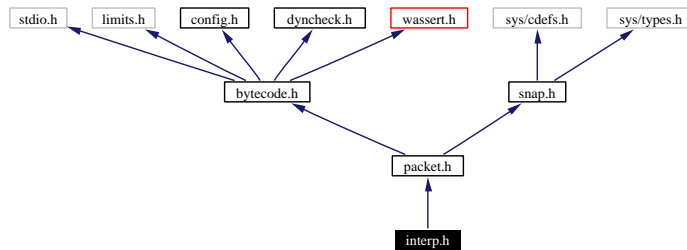
5.16.2.1 int procnetdev_vsn

Definition at line 61 of file interface.h.

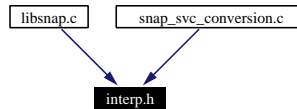
5.17 snap-1.1-wjdb/lib/interp.h File Reference

```
#include "packet.h"
```

Include dependency graph for interp.h:



This graph shows which files directly or indirectly include this file:



Functions

- int `snap_interp_packet` (`packet_t *p`)
- int `heap_alloc` (`packet_t *p`, int `lenb`, int `flag`, `heap_obj **ho`, int `*hoffset`)

5.17.1 Function Documentation

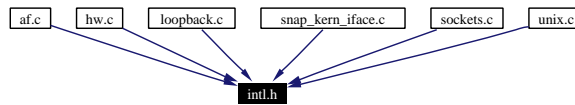
5.17.1.1 int `heap_alloc` (`packet_t * p`, int `lenb`, int `flag`, `heap_obj ** ho`, int `* hoffset`)

5.17.1.2 int `snap_interp_packet` (`packet_t * p`)

Referenced by `handle_snap_request()`, and `snap_receive()`.

5.18 snap-1.1-wjdb/lib/intl.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define _(String) (String)`
- `#define N_(String) (String)`

5.18.1 Define Documentation

5.18.1.1 `#define _(String) (String)`

Definition at line 9 of file intl.h.

Referenced by `afinit()`, `aftrans_opt()`, `get_aftype()`, `hwinit()`, and `sockets_open()`.

5.18.1.2 `#define N_(String) (String)`

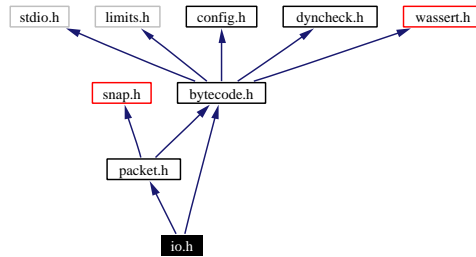
Definition at line 10 of file intl.h.

5.19 snap-1.1-wjdb/lib/io.h File Reference

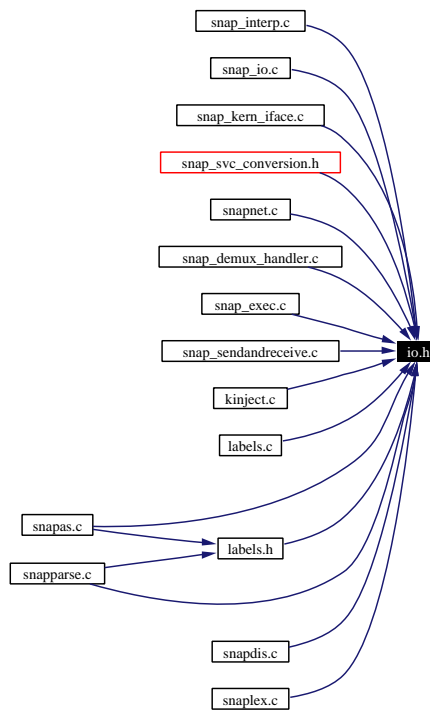
```
#include "bytecode.h"
```

```
#include "packet.h"
```

Include dependency graph for io.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct `buffer_t`

Functions

- **packet_t * unmarshal_packet** (char *buffer, int packet_lenb, int buf_lenb)
- **int marshal_packet** (packet_t *p, int stack_amt, buffer_t *bufstr)
- **int file_to_str** (int fd, buffer_t *buf)

5.19.1 Function Documentation

5.19.1.1 int file_to_str (int *fd*, buffer_t * *buf*)

Definition at line 750 of file snap_io.c.

References buffer_t::lenb, memalloc, and buffer_t::s.

Referenced by init_request(), and main().

5.19.1.2 int marshal_packet (packet_t * *p*, int *stack_amt*, buffer_t * *bufstr*)

Definition at line 162 of file snap_io.c.

References ADDI, BCASTI, BOGUSV, CALLS, packet_t::code_max, packet_t::code_min, snaphdr::code_sizeb, d_printf(), DIVI, ELSE_NOT_IN_FROM_HEAP, EQADR, EQFLT, EQI, EQSTR, EQTUP, FADDI, FDIVI, FGEQI, FGTI, heap_obj::flag, FLEQI, FLTI, FMULI, FSUBI, GEQI, GET_LIT, GET_OP, GTI, packet_t::h_alloc_heap_max, packet_t::hdr, packet_t::heap_max, packet_t::heap_min, snaphdr::heap_sizeb, IF_IN_FROM_HEAP, instr_t, packet_t::iph, packet_t::is_contiguous, heap_obj::len, buffer_t::lenb, LEQI, LTI, memalloc, MULTI, NEQI, NQADR, NQFLT, NQSTR, NQTUP, PADDR, packet_t::pc, PFLT, print_anti_timer, print_timer, PSTR, PTUP, PUSH, buffer_t::s, heap_obj::s, SET_LIT, SNETI, packet_t::sp, packet_t::stack_max, packet_t::stack_min, snaphdr::stack_sizeb, SUBI, SVCV, and value_t.

Referenced by main().

5.19.1.3 packet_t* unmarshal_packet (char * *buffer*, int *packet_lenb*, int *buf_lenb*)

Definition at line 63 of file snap_io.c.

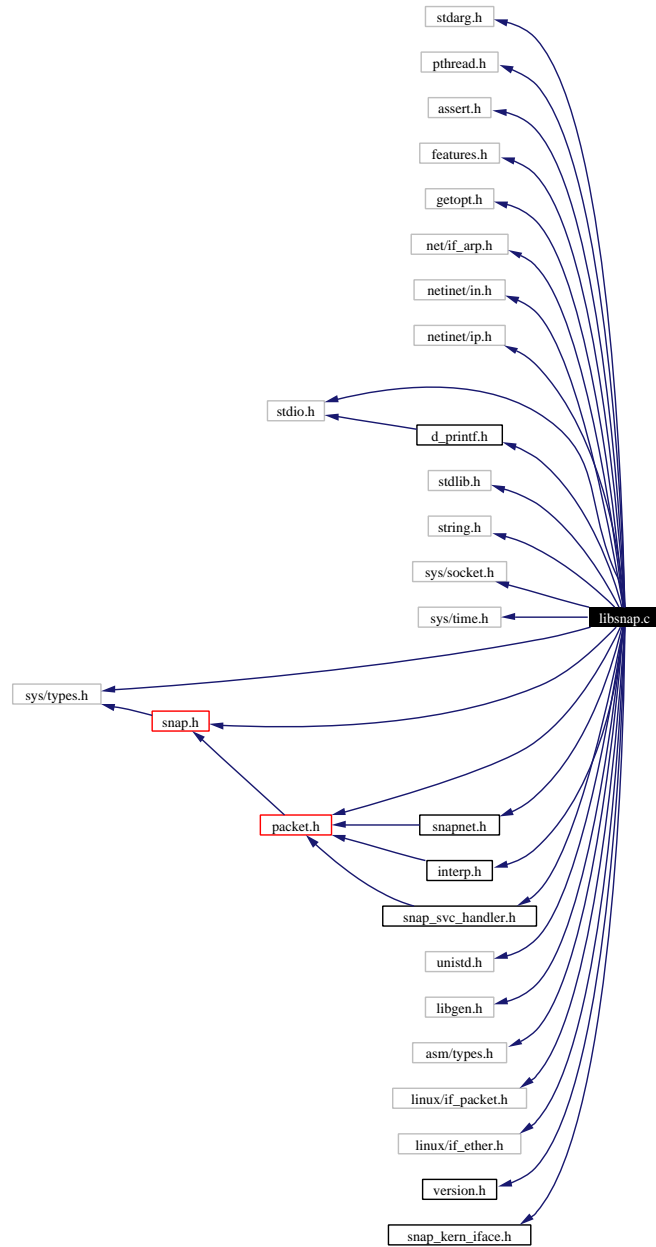
References packet_t::code_max, packet_t::code_min, snaphdr::code_sizeb, d_printf(), snaphdr::entry_point, packet_t::h_alloc_heap_max, packet_t::handler, packet_t::hdr, packet_t::heap_max, packet_t::heap_min, snaphdr::heap_sizeb, instr_t, packet_t::iph, packet_t::is_contiguous, packet_t::pc, print_anti_timer, print_timer, packet_t::sp, packet_t::stack_max, packet_t::stack_min, snaphdr::stack_sizeb, uint32, value_t, and VERIFY.

Referenced by main(), and snap_rcv_pkt().

5.20 snap-1.1-wjdb/lib/libsnap.c File Reference

```
#include <stdarg.h>
#include <pthread.h>
#include <assert.h>
#include <features.h>
#include <getopt.h>
#include <net/if_arp.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <sys/types.h>
#include <unistd.h>
#include <libgen.h>
#include <asm/types.h>
#include <linux/if_packet.h>
#include <linux/if_ether.h>
#include "d_printf.h"
#include "snapnet.h"
#include "packet.h"
#include "snap.h"
#include "version.h"
#include "interp.h"
#include "snap_kern_iface.h"
#include "snap_svc_handler.h"
```

Include dependency graph for libsnap.c:



Data Structures

- struct `cmdline_args`
- struct `glob_conf`

Defines

- `#define _GNU_SOURCE`
- `#define NIPQUAD(addr)`
- `#define UDPPORT 7777`

Functions

- `int snap_receive ()`
- `void usage (char *myname)`
- `void parse_cmdline_snap (int argc, char *argv[])`
- `int snap (struct cmdline_args *cargs)`
- `int add_snap_handler (fd_set *activeset)`
- `int isset_snap_handler (fd_set *activeset)`
- `void clear_snap_handler (fd_set *activeset)`
- `int handle_snap_request ()`
- `void parse_cmdline_snap (int argc, char **argv)`
- `int init_snap (int argc, char **argv)`

Variables

- `glob_conf gc`
- `int ethsock = -1`
- `int losock = -1`
- `int rawiprecvsock = -1`
- `int maxfd`
- `packet_t * p`
- `fd_set rfd`
- `unsigned char ra_space [4]`
- `sockaddr_in bindaddr`
- `sockaddr_in udpaddr`

5.20.1 Define Documentation

5.20.1.1 `#define _GNU_SOURCE`

Definition at line 8 of file libsnap.c.

5.20.1.2 `#define NIPQUAD(addr)`

Value:

```
((unsigned char *)&(addr))[0], \
  ((unsigned char *)&(addr))[1], \
  ((unsigned char *)&(addr))[2], \
  ((unsigned char *)&(addr))[3]
```

Definition at line 42 of file libsnap.c.

Referenced by main(), and parse_cmdline_snap().

5.20.1.3 #define UDPPORT 7777

Definition at line 49 of file libsnap.c.

5.20.2 Function Documentation

5.20.2.1 int add_snap_handler (fd_set * *activeset*)

Definition at line 181 of file libsnap.c.

References ethsock, and rawiprecvsock.

5.20.2.2 void clear_snap_handler (fd_set * *activeset*)

Definition at line 194 of file libsnap.c.

References rawiprecvsock.

5.20.2.3 int handle_snap_request ()

Definition at line 202 of file libsnap.c.

References d_printf_timed(), rawiprecvsock, snap_interp_packet(), and snap_recv_pkt().

5.20.2.4 int init_snap (int *argc*, char ** *argv*)

Definition at line 370 of file libsnap.c.

References cmdline_args::argc, cmdline_args::argv, and snap().

Referenced by main().

5.20.2.5 int isset_snap_handler (fd_set * *activeset*)

Definition at line 188 of file libsnap.c.

References rawiprecvsock.

5.20.2.6 void parse_cmdline_snap (int *argc*, char ** *argv*)

Definition at line 281 of file libsnap.c.

References basename(), d_printf(), gc, glob_conf::herehint, NIPQUAD, set_debug_level_int(), and usage().

Referenced by `snap()`.

5.20.2.7 `void parse_cmdline_snap (int argc, char * argv[])`

5.20.2.8 `int snap (struct cmdline_args * cargs)`

Definition at line 79 of file `libsnap.c`.

References `cmdline_args::argc`, `bindaddr`, `d_printf()`, `ethsock`, `gc`, `glob_conf::herehint`, `losock`, `maxfd`, `parse_cmdline_snap()`, `ra_space`, `rawiprecvsock`, `set_debug_level()`, and `snap_svc_handler_init()`.

Referenced by `init_snap()`.

5.20.2.9 `int snap_receive ()`

Definition at line 222 of file `libsnap.c`.

References `d_printf()`, `d_printf_timed()`, `maxfd`, `rawiprecvsock`, `rfd`s, `snap_interp_packet()`, and `snap_recv_pkt()`.

Referenced by `main()`.

5.20.2.10 `void usage (char * myname)`

Definition at line 272 of file `libsnap.c`.

5.20.3 Variable Documentation

5.20.3.1 `struct sockaddr_in bindaddr`

Definition at line 76 of file `libsnap.c`.

Referenced by `snap()`, `snap_demux_init_rawip()`, and `snap_demux_init_udp()`.

5.20.3.2 `int ethsock = -1`

Definition at line 67 of file `libsnap.c`.

Referenced by `add_snap_handler()`, and `snap()`.

5.20.3.3 `struct glob_conf gc`

Referenced by `parse_cmdline_snap()`, and `snap()`.

5.20.3.4 `int losock = -1`

Definition at line 68 of file `libsnap.c`.

Referenced by `snap()`.

5.20.3.5 `int maxfd`

Definition at line 71 of file `libsnap.c`.

Referenced by `snap()`, and `snap_receive()`.

5.20.3.6 `packet_t* p`

Definition at line 72 of file `libsnap.c`.

5.20.3.7 `unsigned char ra_space[4]`

Definition at line 75 of file `libsnap.c`.

Referenced by `init_request()`, `main()`, and `snap()`.

5.20.3.8 `int rawiprecvsock = -1`

Definition at line 70 of file `libsnap.c`.

Referenced by `add_snap_handler()`, `clear_snap_handler()`, `handle_snap_request()`, `isset_snap_handler()`, `snap()`, and `snap_receive()`.

5.20.3.9 `fd_set rfd`

Definition at line 74 of file `libsnap.c`.

Referenced by `snap_receive()`.

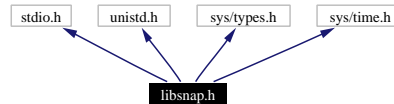
5.20.3.10 `struct sockaddr_in udpaddr`

Definition at line 77 of file `libsnap.c`.

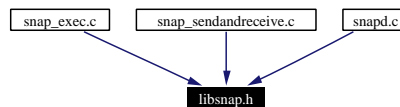
5.21 snap-1.1-wjdb/lib/libsnap.h File Reference

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/time.h>
```

Include dependency graph for libsnap.h:



This graph shows which files directly or indirectly include this file:



Defines

- `#define SNAP_LIB_H 1`

Functions

- `int init_snap (int, char **)`
- `int snap_receive ()`
- `int add_snap_handler (fd_set *)`
- `int isset_snap_handler (fd_set *)`
- `void clear_snap_handler (fd_set *)`
- `int handle_snap_request ()`

5.21.1 Define Documentation

5.21.1.1 `#define SNAP_LIB_H 1`

Definition at line 6 of file libsnap.h.

5.21.2 Function Documentation

5.21.2.1 int add_snap_handler (fd_set *)

Definition at line 181 of file libsnap.c.

References ethsock, and rawiprecvsock.

5.21.2.2 void clear_snap_handler (fd_set *)

Definition at line 194 of file libsnap.c.

References rawiprecvsock.

5.21.2.3 int handle_snap_request ()

Definition at line 202 of file libsnap.c.

References d_printf_timed(), rawiprecvsock, snap_interp_packet(), and snap_recv_pkt().

5.21.2.4 int init_snap (int, char **)

Definition at line 370 of file libsnap.c.

References cmdline_args::argc, cmdline_args::argv, and snap().

Referenced by main().

5.21.2.5 int isset_snap_handler (fd_set *)

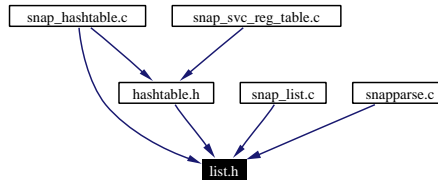
Definition at line 188 of file libsnap.c.

References rawiprecvsock.

5.21.2.6 int snap_receive ()

5.22 snap-1.1-wjdb/lib/list.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct `l`

Typedefs

- typedef `l` `list_t`

Functions

- `list_t * cons` (`void *v`, `list_t *next`)
- `void free_list` (`list_t *list`)
- `int length_list` (`list_t *list`)

5.22.1 Typedef Documentation

5.22.1.1 typedef struct `l` `list_t`

5.22.2 Function Documentation

5.22.2.1 `list_t*` `cons` (`void * v`, `list_t * next`)

Definition at line 24 of file `snap_list.c`.

5.22.2.2 `void free_list` (`list_t * list`)

Definition at line 41 of file `snap_list.c`.

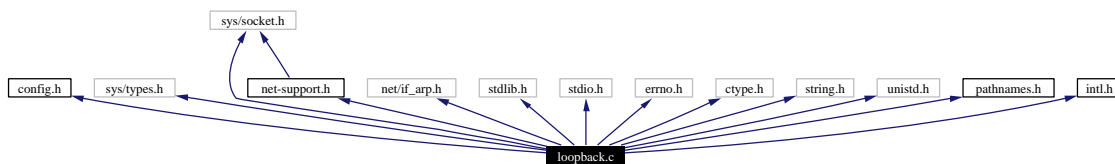
5.22.2.3 `int length_list` (`list_t * list`)

Definition at line 54 of file `snap_list.c`.

5.23 snap-1.1-wjdb/lib/loopback.c File Reference

```
#include "config.h"
#include <sys/types.h>
#include <sys/socket.h>
#include <net/if_arp.h>
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <ctype.h>
#include <string.h>
#include <unistd.h>
#include "net-support.h"
#include "pathnames.h"
#include "intl.h"
```

Include dependency graph for loopback.c:



Variables

- hwtype `unspec_hwtype`
- hwtype `loop_hwtype`

5.23.1 Variable Documentation

5.23.1.1 struct hwtype loop_hwtype

Initial value:

```
{
  "loop",  NULL,  255,  0,
  NULL,   NULL,  NULL,  NULL
}
```

Definition at line 67 of file loopback.c.

5.23.1.2 struct hwtype unspec_hwtype

Initial value:

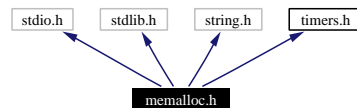
```
{
  "unspec", NULL,      -1,    0,
  pr_unspec,  pr_sunspec,  NULL,  NULL
}
```

Definition at line 62 of file loopback.c.

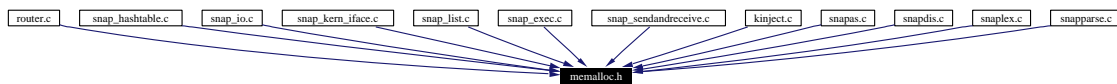
5.24 snap-1.1-wjdb/lib/memalloc.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "timers.h"
```

Include dependency graph for memalloc.h:



This graph shows which files directly or indirectly include this file:



Defines

- #define `memalloc(ptr, t, sz)`

5.24.1 Define Documentation

5.24.1.1 #define `memalloc(ptr, t, sz)`

Value:

```
{ void *_result;
    print_anti_timer(12,"memalloc");
    _result = (void *)malloc(sz);
    if (_result == NULL) {
        fprintf(stderr,"%s:%d: malloc failed\n",__FILE__,__LINE__);
        fflush(stderr);
        exit(1);
    }
    bzero(_result,sz);
    (ptr) = (t)_result;
    print_timer(12,"memalloc");
}
```

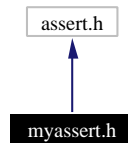
Definition at line 32 of file memalloc.h.

Referenced by `cons()`, `file_to_str()`, `ht_create()`, `ht_insert()`, `main()`, `marshal_packet()`, `parse_cmdline()`, `read_ifaces()`, `read_routes()`, and `yyparse()`.

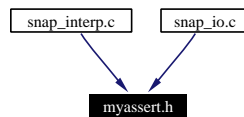
5.25 snap-1.1-wjdb/lib/myassert.h File Reference

```
#include <assert.h>
```

Include dependency graph for myassert.h:



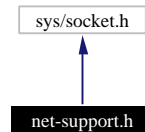
This graph shows which files directly or indirectly include this file:



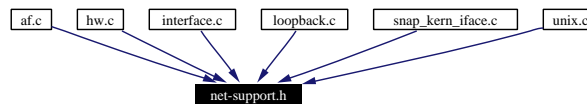
5.26 snap-1.1-wjdb/lib/net-support.h File Reference

```
#include <sys/socket.h>
```

Include dependency graph for net-support.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **aftype**
- struct **hwtype**

Defines

- #define **RTACTION_ADD** 1
- #define **RTACTION_DEL** 2
- #define **RTACTION_HELP** 3
- #define **RTACTION_FLUSH** 4
- #define **RTACTION_SHOW** 5
- #define **FLAG_EXT** 3
- #define **FLAG_NUM** 4
- #define **FLAG_SYM** 8
- #define **FLAG_CACHE** 16
- #define **FLAG_FIB** 32
- #define **FLAG_VERBOSE** 64
- #define **AFTRANS_OPTS**
- #define **AFTRANS_CNT** 9
- #define **EINTERN**(file, text)
- #define **ENOSUPP**(A, B)
- #define **ESYSNOT**(A, B) fprintf(stderr, _("%s: no support for '%s' on this system.\n"),A,B)

- #define **E_NOTFOUND** 8
- #define **E_SOCKET** 7
- #define **E_LOOKUP** 6
- #define **E_VERSION** 5
- #define **E_USAGE** 4
- #define **E_OPTERR** 3
- #define **E_INTERN** 2
- #define **E_NOSUPP** 1

Functions

- **hwtype** * **get_hwtype** (const char *name)
- **hwtype** * **get_hwntype** (int type)
- **aftype** * **get_aftype** (const char *name)
- **aftype** * **get_afntype** (int type)
- int **getargs** (char *string, char *arguments[])
- void **getroute_init** (void)
- void **setroute_init** (void)
- void **activate_init** (void)
- int **route_info** (const char *afname, int flags)
- int **route_edit** (int action, const char *afname, int flags, char **argv)
- int **activate_ld** (const char *hwname, int fd)
- int **ip_masq_info** (int numeric, int ext)
- int **INET_rprint** (int options)
- int **INET6_rprint** (int options)
- int **DDP_rprint** (int options)
- int **IPX_rprint** (int options)
- int **NETROM_rprint** (int options)
- int **AX25_rprint** (int options)
- int **INET_rinput** (int action, int flags, char **argv)
- int **INET6_rinput** (int action, int flags, char **argv)
- int **DDP_rinput** (int action, int flags, char **argv)
- int **IPX_rinput** (int action, int flags, char **argv)
- int **NETROM_rinput** (int action, int flags, char **argv)
- int **AX25_rinput** (int action, int flags, char **argv)
- int **aftrans_opt** (const char *arg)
- void **aftrans_def** (char *tool, char *argv0, char *dflt)
- char * **get_sname** (int socknumber, char *proto, int numeric)

Variables

- int **flag_unx**
- int **flag_ipx**
- int **flag_ax25**
- int **flag_ddp**
- int **flag_netrom**

- int **flag_inet**
- int **flag_inet6**
- char **afname** []

5.26.1 Define Documentation

5.26.1.1 #define AFTRANS_CNT 9

Definition at line 134 of file net-support.h.

5.26.1.2 #define AFTRANS_OPTS

Value:

```
{"ax25",    0, 0, 1}, \
 {"ip",     0, 0, 1}, \
 {"ipx",    0, 0, 1}, \
 {"appletalk", 0, 0, 1}, \
 {"netrom", 0, 0, 1}, \
 {"inet",   0, 0, 1}, \
 {"ddp",    0, 0, 1}, \
 {"unix",   0, 0, 1}, \
 {"tcpip",  0, 0, 1}
```

Definition at line 124 of file net-support.h.

5.26.1.3 #define E_INTERN 2

Definition at line 152 of file net-support.h.

5.26.1.4 #define E_LOOKUP 6

Definition at line 148 of file net-support.h.

5.26.1.5 #define E_NOSUPP 1

Definition at line 153 of file net-support.h.

5.26.1.6 #define E_NOTFOUND 8

Definition at line 146 of file net-support.h.

5.26.1.7 #define E_OPTERR 3

Definition at line 151 of file net-support.h.

5.26.1.8 #define E_SOCK 7

Definition at line 147 of file net-support.h.

5.26.1.9 #define E_USAGE 4

Definition at line 150 of file net-support.h.

5.26.1.10 #define E_VERSION 5

Definition at line 149 of file net-support.h.

5.26.1.11 #define EINTERN(file, text)

Value:

```
fprintf(stderr, \
    "%s: Internal Error '%s'.\n",file,text);
```

Definition at line 136 of file net-support.h.

5.26.1.12 #define ENOSUPP(A, B)

Value:

```
fprintf(stderr,\
    _("%s: feature '%s' not supported.\n" \
    "Please recompile 'net-tools' with "\
    "newer kernel source or full configuration.\n"),A,B)
```

Definition at line 139 of file net-support.h.

5.26.1.13 #define ESYSNOT(A, B) fprintf(stderr, _("%s: no support for '%s' on this system.\n"),A,B)

Definition at line 144 of file net-support.h.

5.26.1.14 #define FLAG_CACHE 16

Definition at line 89 of file net-support.h.

5.26.1.15 #define FLAG_EXT 3

Definition at line 86 of file net-support.h.

5.26.1.16 #define FLAG_FIB 32

Definition at line 90 of file net-support.h.

5.26.1.17 #define FLAG_NUM 4

Definition at line 87 of file net-support.h.

5.26.1.18 #define FLAG_SYM 8

Definition at line 88 of file net-support.h.

5.26.1.19 #define FLAG_VERBOSE 64

Definition at line 91 of file net-support.h.

5.26.1.20 #define RTACTION_ADD 1

Definition at line 80 of file net-support.h.

5.26.1.21 #define RTACTION_DEL 2

Definition at line 81 of file net-support.h.

5.26.1.22 #define RTACTION_FLUSH 4

Definition at line 83 of file net-support.h.

5.26.1.23 #define RTACTION_HELP 3

Definition at line 82 of file net-support.h.

5.26.1.24 #define RTACTION_SHOW 5

Definition at line 84 of file net-support.h.

5.26.2 Function Documentation

5.26.2.1 void `activate_init` (void)

5.26.2.2 int `activate_ld` (const char * *hwname*, int *fd*)

5.26.2.3 void `aftrans_def` (char * *tool*, char * *argv0*, char * *dft*)

Definition at line 135 of file `af.c`.

References `afname`, and `aftrans_opt()`.

5.26.2.4 int `aftrans_opt` (const char * *arg*)

Definition at line 211 of file `af.c`.

References `_`, `afname`, `aftrans`, `aftrans_t::alias`, `aftrans_t::flag`, and `aftrans_t::name`.

Referenced by `aftrans_def()`.

5.26.2.5 int `AX25_rinput` (int *action*, int *flags*, char ** *argv*)

5.26.2.6 int `AX25_rprint` (int *options*)

5.26.2.7 int `DDP_rinput` (int *action*, int *flags*, char ** *argv*)

5.26.2.8 int `DDP_rprint` (int *options*)

5.26.2.9 struct `aftype*` `get_afntype` (int *type*)

Definition at line 195 of file `af.c`.

References `afinit()`.

5.26.2.10 struct `aftype*` `get_aftype` (const char * *name*)

Definition at line 175 of file `af.c`.

References `_`, and `afinit()`.

5.26.2.11 struct `hwtype*` `get_hwntype` (int *type*)

Definition at line 202 of file `hw.c`.

References `hwinit()`.

5.26.2.12 struct `hwtype*` `get_hwtype` (const char * *name*)

Definition at line 184 of file `hw.c`.

References `hwinit()`.

- 5.26.2.13 `char* get_sname (int socknumber, char * proto, int numeric)`
- 5.26.2.14 `int getargs (char * string, char * arguments[])`
- 5.26.2.15 `void getroute_init (void)`
- 5.26.2.16 `int INET6_rinput (int action, int flags, char ** argv)`
- 5.26.2.17 `int INET6_rprint (int options)`
- 5.26.2.18 `int INET_rinput (int action, int flags, char ** argv)`
- 5.26.2.19 `int INET_rprint (int options)`
- 5.26.2.20 `int ip_masq_info (int numeric, int ext)`
- 5.26.2.21 `int IPX_rinput (int action, int flags, char ** argv)`
- 5.26.2.22 `int IPX_rprint (int options)`
- 5.26.2.23 `int NETROM_rinput (int action, int flags, char ** argv)`
- 5.26.2.24 `int NETROM_rprint (int options)`
- 5.26.2.25 `int route_edit (int action, const char * afname, int flags, char ** argv)`
- 5.26.2.26 `int route_info (const char * afname, int flags)`
- 5.26.2.27 `void setroute_init (void)`

5.26.3 Variable Documentation

5.26.3.1 `char afname[]`

Definition at line 122 of file `net-support.h`.

Referenced by `aftrans_def()`, and `aftrans_opt()`.

5.26.3.2 `int flag_ax25`

Definition at line 116 of file `net-support.h`.

5.26.3.3 int flag_ddp

Definition at line 117 of file net-support.h.

5.26.3.4 int flag_inet

Definition at line 119 of file net-support.h.

5.26.3.5 int flag_inet6

Definition at line 120 of file net-support.h.

5.26.3.6 int flag_ipx

Definition at line 115 of file net-support.h.

5.26.3.7 int flag_netrom

Definition at line 118 of file net-support.h.

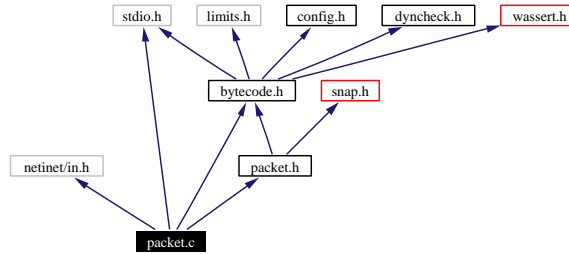
5.26.3.8 int flag_unx

Definition at line 114 of file net-support.h.

5.27 snap-1.1-wjdb/lib/packet.c File Reference

```
#include <netinet/in.h>
#include <stdio.h>
#include "bytecode.h"
#include "packet.h"
```

Include dependency graph for packet.c:



Functions

- int **fprintf_addr** (FILE *f, unsigned int addr)
- int **fprintf_instr** (packet_t *p, FILE *f, instr_t *i)
- int **fprintf_value** (packet_t *p, FILE *f, value_t *v)
- void **fprintf_packet** (FILE *outfile, packet_t *p)

5.27.1 Function Documentation

5.27.1.1 int **fprintf_addr** (FILE *f, unsigned int *addr*)

5.27.1.2 int **fprintf_instr** (packet_t *p, FILE *f, instr_t *i)

Definition at line 269 of file snap_bytecode.c.

References ADDI, ADDRv, ANDI, BCASTI, BEZ, BNE, CALLS, COPY_LIT, DIVI, EQADR, EQEXC, EQFLT, EQI, EQINT, EQSTR, EQTUP, EXCV, FADDI, FDIVI, FGEQI, FGTI, FLEQI, FLOATV, FLTI, FMULLI, fprintf_opcode(), fprintf_value(), FSUBI, GEQI, GET_OP, GTI, instr_t, INTV, JI, LEQI, LSHLI, LTI, MKTUP, MODI, MULTI, NEQI, NQADR, NQEXC, NQFLT, NQINT, NQSTR, NQTUP, NTH, ORI, PADDR, PAJ, PEXC, PFLT, PINT, POPI, PSTR, PTUP, PULL, PUSH, RSHAI, RSHLI, SNETI, STORE, STRV, SUBI, SVCV, TPAJ, TUPLEV, value_t, and XORI.

Referenced by fprintf_packet().

5.27.1.3 void fprintf_packet (FILE * *outfile*, packet_t * *p*)

Definition at line 17 of file packet.c.

References packet_t::code_max, packet_t::code_min, snaphdr::entry_point, fprintf_instr(), fprintf_value(), packet_t::hdr, packet_t::heap_min, instr_t, outfile, packet_t::sp, snaphdr::sport, packet_t::stack_min, and value_t.

Referenced by main().

5.27.1.4 int fprintf_value (packet_t * *p*, FILE * *f*, value_t * *v*)

Definition at line 149 of file snap_bytecode.c.

References fprintf_value_heap(), packet_t::heap_min, and value_t.

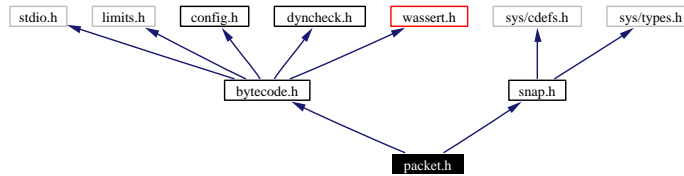
Referenced by fprintf_instr(), and fprintf_packet().

5.28 snap-1.1-wjdb/lib/packet.h File Reference

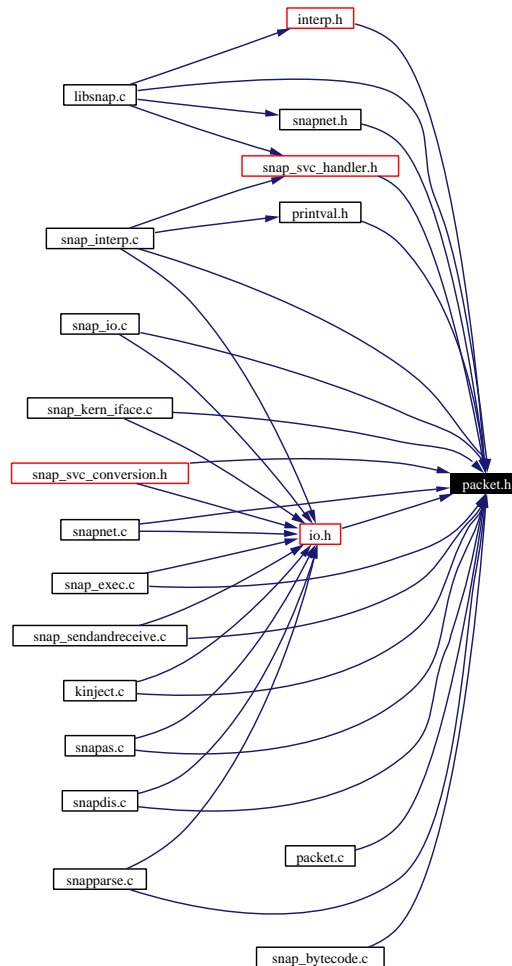
```
#include "bytecode.h"
```

```
#include "snap.h"
```

Include dependency graph for packet.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct `heap_t`
- struct `packet_t`

Typedefs

- typedef `snaphdr header_t`

Functions

- void `fprintf_packet (FILE *f, packet_t *p)`

5.28.1 Typedef Documentation

5.28.1.1 typedef struct snaphdr header_t

Definition at line 23 of file packet.h.

5.28.2 Function Documentation

5.28.2.1 void fprintf_packet (FILE * f, packet_t * p)

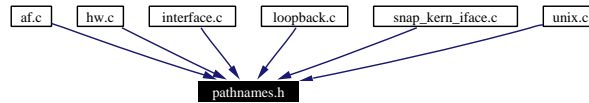
Definition at line 17 of file packet.c.

References `packet_t::code_max`, `packet_t::code_min`, `snaphdr::entry_point`, `fprintf_instr()`, `fprintf_value()`, `packet_t::hdr`, `packet_t::heap_min`, `instr_t`, `outfile`, `packet_t::sp`, `snaphdr::sport`, `packet_t::stack_min`, and `value_t`.

Referenced by `main()`.

5.29 snap-1.1-wjdb/lib/pathnames.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define `_PATH_PROCNET_TCP` `"/proc/net/tcp"`
- #define `_PATH_PROCNET_TCP6` `"/proc/net/tcp6"`
- #define `_PATH_PROCNET_UDP` `"/proc/net/udp"`
- #define `_PATH_PROCNET_UDP6` `"/proc/net/udp6"`
- #define `_PATH_PROCNET_RAW` `"/proc/net/raw"`
- #define `_PATH_PROCNET_RAW6` `"/proc/net/raw6"`
- #define `_PATH_PROCNET_UNIX` `"/proc/net/unix"`
- #define `_PATH_PROCNET_ROUTE` `"/proc/net/route"`
- #define `_PATH_PROCNET_ROUTE6` `"/proc/net/ipv6.route"`
- #define `_PATH_PROCNET_RTCACHE` `"/proc/net/rt.cache"`
- #define `_PATH_PROCNET_AX25_ROUTE` `"/proc/net/ax25.route"`
- #define `_PATH_PROCNET_NR` `"/proc/net/nr"`
- #define `_PATH_PROCNET_NR_NEIGH` `"/proc/net/nr_neigh"`
- #define `_PATH_PROCNET_NR_NODES` `"/proc/net/nr_nodes"`
- #define `_PATH_PROCNET_ARP` `"/proc/net/arp"`
- #define `_PATH_PROCNET_AX25` `"/proc/net/ax25"`
- #define `_PATH_PROCNET_IPX` `"/proc/net/ipx"`
- #define `_PATH_PROCNET_IPX_ROUTE` `"/proc/net/ipx.route"`
- #define `_PATH_PROCNET_ATALK` `"/proc/net/appletalk"`
- #define `_PATH_PROCNET_IP_BLK` `"/proc/net/ip_block"`
- #define `_PATH_PROCNET_IP_FWD` `"/proc/net/ip_forward"`
- #define `_PATH_PROCNET_IP_ACC` `"/proc/net/ip_acct"`
- #define `_PATH_PROCNET_IP_MASQ` `"/proc/net/ip_masquerade"`
- #define `_PATH_PROCNET_NDISC` `"/proc/net/ndisc"`
- #define `_PATH_PROCNET_IFINET6` `"/proc/net/if_inet6"`
- #define `_PATH_PROCNET_DEV` `"/proc/net/dev"`
- #define `_PATH_PROCNET_RARP` `"/proc/net/rarp"`
- #define `_PATH_ETHERS` `"/etc/ethers"`
- #define `_PATH_PROCNET_ROSE_ROUTE` `"/proc/net/rose-routes"`
- #define `_PATH_DEV_ROUTE` `"/dev/route"`

5.29.1 Define Documentation

5.29.1.1 `#define _PATH_DEV_ROUTE "/dev/route"`

Definition at line 44 of file pathnames.h.

5.29.1.2 `#define _PATH_ETHERS "/etc/ethers"`

Definition at line 40 of file pathnames.h.

5.29.1.3 `#define _PATH_PROCNET_ARP "/proc/net/arp"`

Definition at line 27 of file pathnames.h.

5.29.1.4 `#define _PATH_PROCNET_ATALK "/proc/net/appletalk"`

Definition at line 31 of file pathnames.h.

5.29.1.5 `#define _PATH_PROCNET_AX25 "/proc/net/ax25"`

Definition at line 28 of file pathnames.h.

5.29.1.6 `#define _PATH_PROCNET_AX25_ROUTE "/proc/net/ax25_route"`

Definition at line 23 of file pathnames.h.

5.29.1.7 `#define _PATH_PROCNET_DEV "/proc/net/dev"`

Definition at line 38 of file pathnames.h.

5.29.1.8 `#define _PATH_PROCNET_IFINET6 "/proc/net/if_inet6"`

Definition at line 37 of file pathnames.h.

5.29.1.9 `#define _PATH_PROCNET_IP_ACC "/proc/net/ip_acct"`

Definition at line 34 of file pathnames.h.

5.29.1.10 `#define _PATH_PROCNET_IP_BLK`
`"/proc/net/ip_block"`

Definition at line 32 of file pathnames.h.

5.29.1.11 `#define _PATH_PROCNET_IP_FWD`
`"/proc/net/ip_forward"`

Definition at line 33 of file pathnames.h.

5.29.1.12 `#define _PATH_PROCNET_IP_MASQ`
`"/proc/net/ip_masquerade"`

Definition at line 35 of file pathnames.h.

5.29.1.13 `#define _PATH_PROCNET_IPX` `"/proc/net/ipx"`

Definition at line 29 of file pathnames.h.

5.29.1.14 `#define _PATH_PROCNET_IPX_ROUTE`
`"/proc/net/ipx_route"`

Definition at line 30 of file pathnames.h.

5.29.1.15 `#define _PATH_PROCNET_NDISC` `"/proc/net/ndisc"`

Definition at line 36 of file pathnames.h.

5.29.1.16 `#define _PATH_PROCNET_NR` `"/proc/net/nr"`

Definition at line 24 of file pathnames.h.

5.29.1.17 `#define _PATH_PROCNET_NR_NEIGH`
`"/proc/net/nr_neigh"`

Definition at line 25 of file pathnames.h.

5.29.1.18 `#define _PATH_PROCNET_NR_NODES`
`"/proc/net/nr_nodes"`

Definition at line 26 of file pathnames.h.

5.29.1.19 `#define _PATH_PROCNET_RARP "/proc/net/rarp"`

Definition at line 39 of file pathnames.h.

5.29.1.20 `#define _PATH_PROCNET_RAW "/proc/net/raw"`

Definition at line 17 of file pathnames.h.

5.29.1.21 `#define _PATH_PROCNET_RAW6 "/proc/net/raw6"`

Definition at line 18 of file pathnames.h.

5.29.1.22 `#define _PATH_PROCNET_ROSE_ROUTE
"/proc/net/rose_routes"`

Definition at line 41 of file pathnames.h.

5.29.1.23 `#define _PATH_PROCNET_ROUTE "/proc/net/route"`

Definition at line 20 of file pathnames.h.

5.29.1.24 `#define _PATH_PROCNET_ROUTE6
"/proc/net/ipv6_route"`

Definition at line 21 of file pathnames.h.

5.29.1.25 `#define _PATH_PROCNET_RTCACHE
"/proc/net/rt_cache"`

Definition at line 22 of file pathnames.h.

5.29.1.26 `#define _PATH_PROCNET_TCP "/proc/net/tcp"`

Definition at line 13 of file pathnames.h.

5.29.1.27 `#define _PATH_PROCNET_TCP6 "/proc/net/tcp6"`

Definition at line 14 of file pathnames.h.

5.29.1.28 `#define _PATH_PROCNET_UDP "/proc/net/udp"`

Definition at line 15 of file pathnames.h.

5.29.1.29 `#define _PATH_PROCNET_UDP6 "/proc/net/udp6"`

Definition at line 16 of file pathnames.h.

5.29.1.30 `#define _PATH_PROCNET_UNIX "/proc/net/unix"`

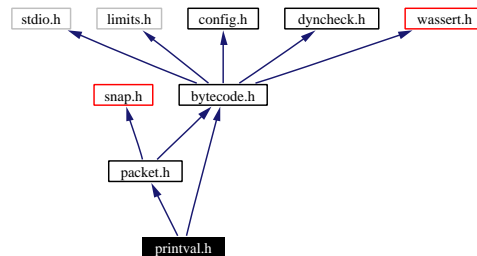
Definition at line 19 of file pathnames.h.

5.30 snap-1.1-wjdb/lib/printval.h File Reference

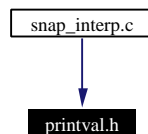
```
#include "bytecode.h"
```

```
#include "packet.h"
```

Include dependency graph for printval.h:



This graph shows which files directly or indirectly include this file:



Functions

- int **printk_addr** (unsigned int addr)
- int **printk_value_tag** (TAG_T)
- int **printk_value** (packet_t *p, value_t *)
- int **printk_opcode** (OPCODE_T)
- int **printk_instr** (packet_t *p, instr_t *)

5.30.1 Function Documentation

5.30.1.1 int printk_addr (unsigned int *addr*)

5.30.1.2 int printk_instr (packet_t * *p*, instr_t *)

5.30.1.3 int printk_opcode (OPCODE_T)

5.30.1.4 int printk_value (packet_t * *p*, value_t *)

5.30.1.5 int printk_value_tag (TAG_T)

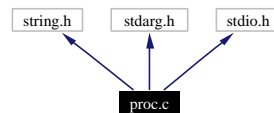
5.31 snap-1.1-wjdb/lib/proc.c File Reference

```
#include <string.h>
```

```
#include <stdarg.h>
```

```
#include <stdio.h>
```

Include dependency graph for proc.c:



Functions

- `char * proc_gen_fmt (char *name, FILE *fh,...)`

5.31.1 Function Documentation

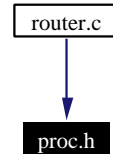
5.31.1.1 `char* proc_gen_fmt (char * name, FILE * fh, ...)`

Definition at line 9 of file `proc.c`.

Referenced by `read_routes()`.

5.32 snap-1.1-wjdb/lib/proc.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- `char * proc_gen_fmt` (`char *name`, `FILE *fh`,...)

5.32.1 Function Documentation

5.32.1.1 `char* proc_gen_fmt` (`char * name`, `FILE * fh`, ...)

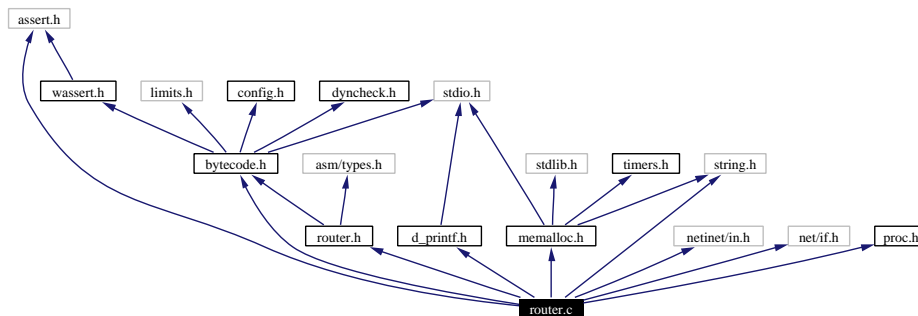
Definition at line 9 of file `proc.c`.

Referenced by `read_routes()`.

5.33 snap-1.1-wjdb/lib/router.c File Reference

```
#include <assert.h>
#include <netinet/in.h>
#include <net/if.h>
#include <string.h>
#include "bytecode.h"
#include "d_printf.h"
#include "memalloc.h"
#include "router.h"
#include "proc.h"
```

Include dependency graph for router.c:



Data Structures

- struct **rtentry**

Defines

- #define **NIPQUAD**(addr)

Functions

- int **get_iface_index** (char *iface_name)
- void **read_ifaces** (char *iface_file)
- void **read_routes** (char *route_file)
- int **nexthop** (**addr_t** dstaddr, struct **rt_lookup** *retval)

Variables

- `rtentry * routes = NULL`
- `int num_rt_entries = 0`
- `int num_routes = 0`
- `char ** ifaces = NULL`
- `int num_if_entries = 0`
- `int num_ifaces = 0`

5.33.1 Define Documentation

5.33.1.1 `#define NIPQUAD(addr)`

Value:

```
((unsigned char *)&(addr))[0], \
    ((unsigned char *)&(addr))[1], \
    ((unsigned char *)&(addr))[2], \
    ((unsigned char *)&(addr))[3]
```

Definition at line 21 of file router.c.

5.33.2 Function Documentation

5.33.2.1 `int get_iface_index (char * iface_name)`

Definition at line 71 of file router.c.

References `ifaces`, and `num_ifaces`.

5.33.2.2 `int nexthop (addr_t dstaddr, struct rt_lookup * retval)`

Definition at line 260 of file router.c.

References `addr_t`, `rt_lookup::hopaddr`, `rt_lookup::ifidx`, `num_routes`, `print_anti_timer`, `print_timer`, `rtentry::rt_dst`, `rtentry::rt_gateway`, `rtentry::rt_genmask`, and `rtentry::rt_ifidx`.

5.33.2.3 `void read_ifaces (char * iface_file)`

Definition at line 84 of file router.c.

References `d_printf()`, `ifaces`, `memalloc`, `num_if_entries`, and `num_ifaces`.

5.33.2.4 `void read_routes (char * route_file)`

Definition at line 154 of file router.c.

References `addr_t`, `d_printf()`, `memalloc`, `num_routes`, `num_rt_entries`, `proc_genfmt()`, `rtentry::rt_dst`, `rtentry::rt_flags`, `rtentry::rt_gateway`, `rtentry::rt_genmask`, `rtentry::rt_ifidx`, `rtentry::rt_irtt`, `rtentry::rt_metric`, `rtentry::rt_mtu`, `rtentry::rt_window`, and `sprintf_addr()`.

5.33.3 Variable Documentation

5.33.3.1 `char** ifaces = NULL`

Definition at line 66 of file `router.c`.

Referenced by `get_iface_index()`, and `read_ifaces()`.

5.33.3.2 `int num_if_entries = 0`

Definition at line 67 of file `router.c`.

Referenced by `read_ifaces()`.

5.33.3.3 `int num_ifaces = 0`

Definition at line 68 of file `router.c`.

Referenced by `get_iface_index()`, and `read_ifaces()`.

5.33.3.4 `int num_routes = 0`

Definition at line 62 of file `router.c`.

Referenced by `nexthop()`, and `read_routes()`.

5.33.3.5 `int num_rt_entries = 0`

Definition at line 61 of file `router.c`.

Referenced by `read_routes()`.

5.33.3.6 `struct rtentry* routes = NULL`

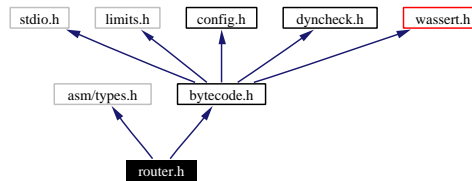
Definition at line 60 of file `router.c`.

5.34 snap-1.1-wjdb/lib/router.h File Reference

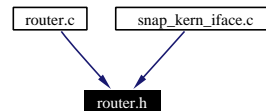
```
#include <asm/types.h>
```

```
#include "bytecode.h"
```

Include dependency graph for router.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **rt_lookup**

Defines

- #define **PROC_NET_ROUTE_PATH** "/proc/net/route"
- #define **PROC_NET_DEV_PATH** "/proc/net/dev"

Typedefs

- typedef __u32 **addr_t**

Functions

- void **read_routes** (char *iface_file)
- void **read_ifaces** (char *route_file)
- int **nexthop** (**addr_t** dst, struct **rt_lookup** *retval)

5.34.1 Define Documentation

5.34.1.1 `#define PROC_NET_DEV_PATH "/proc/net/dev"`

Definition at line 15 of file router.h.

5.34.1.2 `#define PROC_NET_ROUTE_PATH "/proc/net/route"`

Definition at line 14 of file router.h.

5.34.2 Typedef Documentation

5.34.2.1 `typedef __u32 addr_t`

Definition at line 17 of file router.h.

Referenced by `nexthop()`, and `read_routes()`.

5.34.3 Function Documentation

5.34.3.1 `int nexthop (addr_t dst, struct rt_lookup * retval)`

Definition at line 260 of file router.c.

References `addr_t`, `rt_lookup::hopaddr`, `rt_lookup::ifidx`, `num_routes`, `print_anti_timer`, `print_timer`, `rtentry::rt_dst`, `rtentry::rt_gateway`, `rtentry::rt_genmask`, and `rtentry::rt_ifidx`.

5.34.3.2 `void read_ifaces (char * route_file)`

Definition at line 84 of file router.c.

References `d_printf()`, `ifaces`, `memalloc`, `num_if_entries`, and `num_ifaces`.

5.34.3.3 `void read_routes (char * iface_file)`

Definition at line 154 of file router.c.

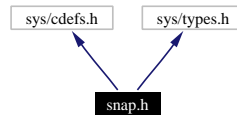
References `addr_t`, `d_printf()`, `memalloc`, `num_routes`, `num_rt_entries`, `proc_genfmt()`, `rtentry::rt_dst`, `rtentry::rt_flags`, `rtentry::rt_gateway`, `rtentry::rt_genmask`, `rtentry::rt_ifidx`, `rtentry::rt_irtt`, `rtentry::rt_metric`, `rtentry::rt_mtu`, `rtentry::rt_window`, and `sprintf_addr()`.

5.35 snap-1.1-wjdb/lib/snap.h File Reference

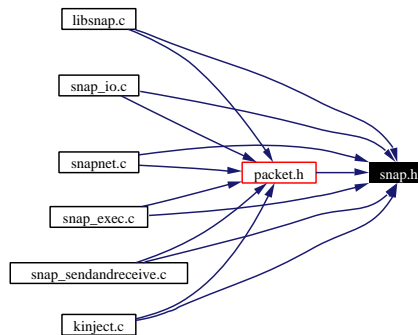
```
#include <sys/cdefs.h>
```

```
#include <sys/types.h>
```

Include dependency graph for snap.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **snaphdr**

Defines

- `#define IPPROTO_SNAP 130`

5.35.1 Define Documentation

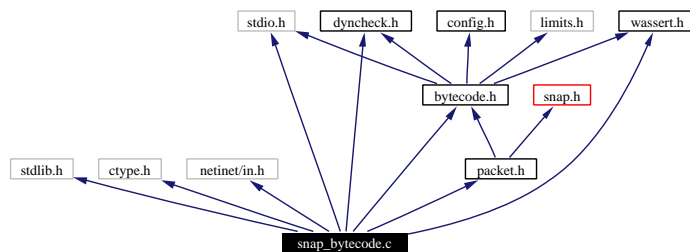
5.35.1.1 `#define IPPROTO_SNAP 130`

Definition at line 27 of file snap.h.

5.36 snap-1.1-wjdb/lib/snap_bytecode.c File Reference

```
#include <stdlib.h>
#include <ctype.h>
#include <netinet/in.h>
#include <stdio.h>
#include "bytecode.h"
#include "dyncheck.h"
#include "wassert.h"
#include "packet.h"
```

Include dependency graph for snap_bytecode.c:



Defines

- #define **GET_TAG_HEAP**(v) ((v) >> (sizeof(unsigned int)*8-TAGSZ))
- #define **SET_TAG_HEAP**(v, t) ((v) = (((v) << TAGSZ) >> TAGSZ) | ((t) << (sizeof(unsigned int)*8-TAGSZ)))
- #define **GET_INT_HEAP**(v) (((int)((v) << TAGSZ)) >> TAGSZ)
- #define **SET_INT_HEAP**(v, i) ((v) = (((v) >> (sizeof(unsigned int)*8-TAGSZ)) << (sizeof(unsigned int)*8-TAGSZ)) | (((unsigned int)(i)) << TAGSZ) >> TAGSZ)
- #define **GET_OFFS_HEAP** GET_INT_HEAP
- #define **SET_OFFS_HEAP** SET_INT_HEAP

Functions

- int **fprintf_addr** (FILE *f, uint32_t addr)
- int **sprintf_addr** (char *buf, uint32_t addr)
- int **fprintf_value_tag** (FILE *f, TAG_T tag)
- int **fprintf_value_heap** (void *heap, FILE *f, value_t *val)
- int **fprintf_value** (packet_t *p, FILE *f, value_t *val)

- int **fprintf_opcode** (FILE *f, OPCODE_T op)
- int **fprintf_instr** (packet_t *p, FILE *f, instr_t *i)

5.36.1 Define Documentation

5.36.1.1 `#define GET_INT_HEAP(v) (((int)((v) << TAGSZ)) >> TAGSZ)`

Definition at line 56 of file snap_bytecode.c.

Referenced by `fprintf_value_heap()`.

5.36.1.2 `#define GET_OFFS_HEAP GET_INT_HEAP`

Definition at line 59 of file snap_bytecode.c.

Referenced by `fprintf_value_heap()`.

5.36.1.3 `#define GET_TAG_HEAP(v) ((v) >> (sizeof(unsigned int)*8-TAGSZ))`

Definition at line 53 of file snap_bytecode.c.

5.36.1.4 `#define SET_INT_HEAP(v, i) ((v) = (((v) >> (sizeof(unsigned int)*8-TAGSZ)) << (sizeof(unsigned int)*8-TAGSZ)) | (((unsigned int)(i) << TAGSZ) >> TAGSZ))`

Definition at line 57 of file snap_bytecode.c.

5.36.1.5 `#define SET_OFFS_HEAP SET_INT_HEAP`

Definition at line 60 of file snap_bytecode.c.

5.36.1.6 `#define SET_TAG_HEAP(v, t) ((v) = (((v) << TAGSZ) >> TAGSZ) | ((t) << (sizeof(unsigned int)*8-TAGSZ)))`

Definition at line 54 of file snap_bytecode.c.

5.36.2 Function Documentation

5.36.2.1 int **fprintf_addr** (FILE *f, uint32_t *addr*)

Definition at line 22 of file snap_bytecode.c.

Referenced by `fprintf_value_heap()`.

5.36.2.2 int fprintf_instr (packet_t * p, FILE * f, instr_t * i)

Definition at line 269 of file snap_bytecode.c.

References ADDI, ADDR, ANDI, BCASTI, BEZ, BNE, CALLS, COPY_LIT, DIVI, EQADR, EQEXC, EQFLT, EQI, EQINT, EQSTR, EQTUP, EXCV, FADDI, FDIVI, FGEQI, FGTI, FLEQI, FLOATV, FLTI, FMULI, fprintf_opcode(), fprintf_value(), FSUBI, GEQI, GET_OP, GTI, instr_t, INTV, JI, LEQI, LSHLI, LTI, MKTUP, MODI, MULTI, NEQI, NQADR, NQEXC, NQFLT, NQINT, NQSTR, NQTUP, NTH, ORI, PADDR, PAJ, PEXC, PFLT, PINT, POPI, PSTR, PTUP, PULL, PUSH, RSHAI, RSHLI, SNETI, STORE, STRV, SUBI, SVCV, TPAJ, TUPLEV, value_t, and XORI.

Referenced by fprintf_packet().

5.36.2.3 int fprintf_opcode (FILE * f, OPCODE_T op)

Definition at line 154 of file snap_bytecode.c.

References ADD, ADDI, AND, ANDI, BCAST, BCASTI, BEZ, BNE, CALLS, DEMUX, DEMUXI, DIV, DIVI, EQ, EQADR, EQEXC, EQFLT, EQI, EQINT, EQSTR, EQTUP, EXIT, FADDI, FDIVI, FGEQI, FGTI, FLEQI, FLTI, FMULI, FORW, FORWTO, FSUBI, GEQ, GEQI, GETDST, GETLD, GETRB, GETSPT, GETSRC, GT, GTI, HERE, HOP, ISHERE, ISTUP, ISX, JI, LEN, LEQ, LEQI, LNOT, LSHL, LSHLI, LT, LTI, MKTUP, MOD, MODI, MULT, MULTI, NEG, NEQ, NEQI, NOT, NQADR, NQEXC, NQFLT, NQINT, NQSTR, NQTUP, NTH, OPCODE_T, OR, ORI, PADDR, PAJ, PEXC, PFLT, PINT, POP, POPI, PRINT, PSTR, PTUP, PULL, PUSH, ROUTE, RSHA, RSHAI, RSHL, RSHLI, RTDEV, SEND, SNET, SNETI, STORE, SUB, SUBI, SVCV, TPAJ, XOR, and XORI.

Referenced by fprintf_instr().

5.36.2.4 int fprintf_value (packet_t * p, FILE * f, value_t * val)

Definition at line 149 of file snap_bytecode.c.

References fprintf_value_heap(), packet_t::heap_min, and value_t.

Referenced by fprintf_instr(), and fprintf_packet().

5.36.2.5 int fprintf_value_heap (void * heap, FILE * f, value_t * val)

Definition at line 71 of file snap_bytecode.c.

References ADDR, EXCV, float32, FLOATV, fprintf_addr(), GET_ADDR_VAL, GET_INT_HEAP, GET_OFFSETS_HEAP, GET_TAG, INTV, heap_obj::len, heap_obj::s, STRV, TUPLEV, and value_t.

Referenced by fprintf_value().

5.36.2.6 `int fprintf_value_tag (FILE * f, TAG_T tag)`

Definition at line 32 of file `snap_bytecode.c`.

References `ADDRV`, `EXCV`, `FLOATV`, `INTV`, `STRV`, `TAG_T`, and `TUPLEV`.

5.36.2.7 `int sprintf_addr (char * buf, uint32_t addr)`

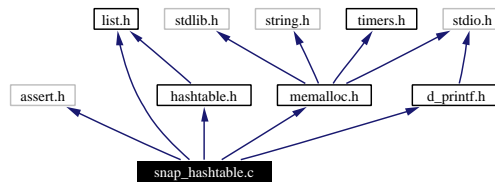
Definition at line 27 of file `snap_bytecode.c`.

Referenced by `read_routes()`.

5.37 snap-1.1-wjdb/lib/snap_hashtable.c File Reference

```
#include <assert.h>
#include "list.h"
#include "memalloc.h"
#include "hashtable.h"
#include "d_printf.h"
```

Include dependency graph for snap_hashtable.c:



Functions

- `int hash_string (char *s)`
- `hash_table_t * ht_create (int sz, int(*cmp)(const void *, const void *), int(*hash)(void *))`
- `void ht_insert (hash_table_t *t, void *key, void *val)`
- `void * ht_lookup (hash_table_t *t, void *key)`
- `void ht_remove (hash_table_t *t, void *key)`

Variables

- `int ht_errno = 0`

5.37.1 Function Documentation

5.37.1.1 int hash_string (char * s)

Definition at line 25 of file snap_hashtable.c.

5.37.1.2 hash_table_t* ht_create (int sz, int(* cmp)(const void *, const void *), int(* hash)(void *))

Definition at line 46 of file snap_hashtable.c.

5.37.1.3 void ht_insert (hash_table_t * t, void * key, void * val)

Definition at line 78 of file snap_hashtable.c.

5.37.1.4 void* ht_lookup (hash_table_t * t, void * key)

Definition at line 121 of file snap_hashtable.c.

5.37.1.5 void ht_remove (hash_table_t * t, void * key)

Definition at line 134 of file snap_hashtable.c.

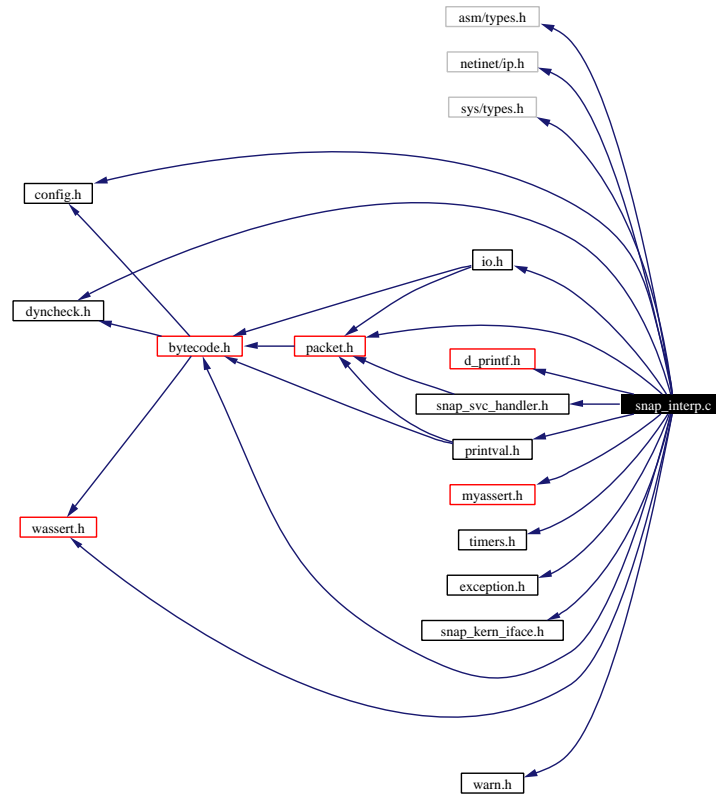
5.37.2 Variable Documentation**5.37.2.1 int ht_errno = 0**

Definition at line 43 of file snap_hashtable.c.

5.38 snap-1.1-wjdb/lib/snap_interp.c File Reference

```
#include <asm/types.h>
#include <netinet/ip.h>
#include <sys/types.h>
#include "bytecode.h"
#include "config.h"
#include "d_printf.h"
#include "dyncheck.h"
#include "io.h"
#include "myassert.h"
#include "packet.h"
#include "timers.h"
#include "exception.h"
#include "wassert.h"
#include "snap_kern_iface.h"
#include "snap_svc_handler.h"
#include "printval.h"
#include "warn.h"
```

Include dependency graph for snap_interp.c:



Defines

- `#define NIPQUAD(addr) addr << 24 >> 24, addr << 16 >> 24, addr << 8 >> 24, addr >> 24`

5.38.1 Define Documentation

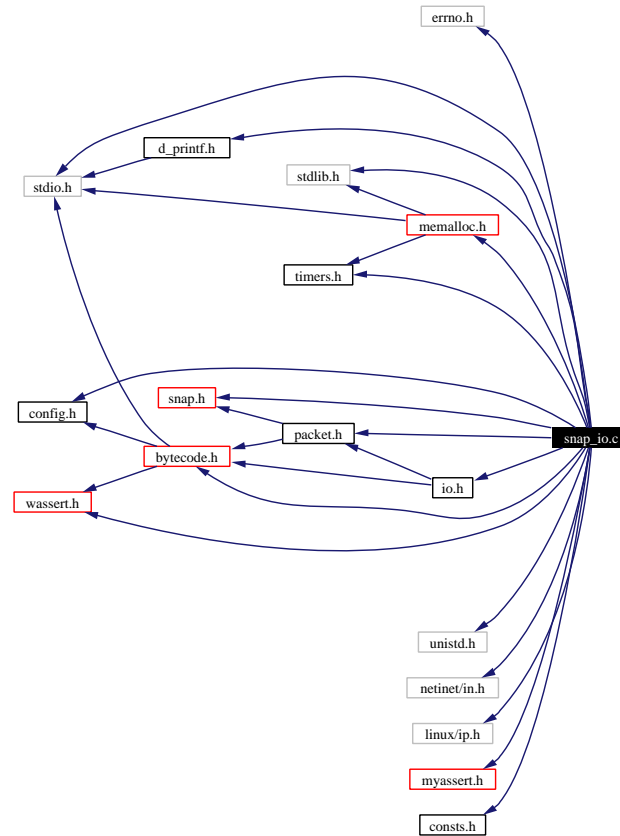
5.38.1.1 `#define NIPQUAD(addr) addr << 24 >> 24, addr << 16 >> 24, addr << 8 >> 24, addr >> 24`

Definition at line 21 of file `snap_interp.c`.

5.39 snap-1.1-wjdb/lib/snap_io.c File Reference

```
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <netinet/in.h>
#include <linux/ip.h>
#include "snap.h"
#include "config.h"
#include "myassert.h"
#include "consts.h"
#include "bytecode.h"
#include "io.h"
#include "memalloc.h"
#include "packet.h"
#include "timers.h"
#include "d_printf.h"
#include "wassert.h"
```

Include dependency graph for snap_io.c:



Defines

- `#define VERIFY(e)`
- `#define IF_IN_FROM_HEAP(ho, sizeb, hmin, hmax, hmin2, hmax2)`
- `#define ELSE_NOT_IN_FROM_HEAP } else`
- `#define IN_TOSPACE_HEAP(hdst, sizeb, toh, tohmax) (((void *) (hdst) >= (toh)) && (((void *) (hdst) + (sizeb)) <= (tohmax)))`

Functions

- `packet_t * unmarshal_packet (char *buffer, int packet_lenb, int buf_lenb)`
- `int marshal_packet (packet_t *p, int stack_amt, buffer_t *bufstr)`
- `int file_to_str (int fd, buffer_t *buf)`

5.39.1 Define Documentation

5.39.1.1 `#define ELSE_NOT_IN_FROM_HEAP } else`

Definition at line 53 of file `snap_io.c`.

Referenced by `marshal_packet()`.

5.39.1.2 `#define IF_IN_FROM_HEAP(ho, sizeb, hmin, hmax, hmin2, hmax2)`

Value:

```
if (((void *) (ho) >= (void *) (hmin)) && \
    ((void *) (ho) < (void *) (hmax)) && \
    ((void *) (ho) + \
     ((sizeb) = (ho)->len + sizeof(heap_obj))) <= (void *) (hmax)) || \
    ((void *) (ho) >= (void *) (hmin2)) && \
    ((void *) (ho) < (void *) (hmax2)) && \
    ((void *) (ho) + \
     ((sizeb) = (ho)->len + sizeof(heap_obj))) <= (void *) (hmax2))) { \
    if (((sizeb) & 0x3) != 0) { \
        (sizeb) = ((sizeb) & ~0x3) + 4; \
    } \
}
```

Definition at line 40 of file `snap_io.c`.

Referenced by `marshal_packet()`.

5.39.1.3 `#define IN_TOSPACE_HEAP(hdst, sizeb, toh, tohmax)` `(((void *) (hdst) >= (toh)) && (((void *) (hdst) + (sizeb)) <= (tohmax)))`

Definition at line 55 of file `snap_io.c`.

5.39.1.4 `#define VERIFY(e)`

Value:

```
if (e); \
else { \
    fprintf(stderr, "%s:%d: packet check failed\n", __FILE__, __LINE__); \
    return NULL; \
}
```

Definition at line 31 of file `snap_io.c`.

Referenced by `unmarshal_packet()`.

5.39.2 Function Documentation

5.39.2.1 `int file_to_str (int fd, buffer_t * buf)`

Definition at line 750 of file `snap_io.c`.

References `buffer_t::lenb`, `memalloc`, and `buffer_t::s`.

Referenced by `init_request()`, and `main()`.

5.39.2.2 `int marshal_packet (packet_t * p, int stack_amt, buffer_t * bufstr)`

Definition at line 162 of file `snap_io.c`.

References `ADDI`, `BCASTI`, `BOGUSV`, `CALLS`, `packet_t::code_max`, `packet_t::code_min`, `snaphdr::code_sizeb`, `d_printf()`, `DIVI`, `ELSE_NOT_IN_FROM_HEAP`, `EQADR`, `EQFLT`, `EQI`, `EQSTR`, `EQTUP`, `FADDI`, `FDIVI`, `FGEQI`, `FGTI`, `heap_obj::flag`, `FLEQI`, `FLTI`, `FMULI`, `FSUBI`, `GEQI`, `GET_LIT`, `GET_OP`, `GTI`, `packet_t::h_alloc_heap_max`, `packet_t::hdr`, `packet_t::heap_max`, `packet_t::heap_min`, `snaphdr::heap_sizeb`, `IF_IN_FROM_HEAP`, `instr_t`, `packet_t::iph`, `packet_t::is_contiguous`, `heap_obj::len`, `buffer_t::lenb`, `LEQI`, `LTI`, `memalloc`, `MULTI`, `NEQI`, `NQADR`, `NQFLT`, `NQSTR`, `NQTUP`, `PADDR`, `packet_t::pc`, `PFLT`, `print_anti_timer`, `print_timer`, `PSTR`, `PTUP`, `PUSH`, `heap_obj::s`, `buffer_t::s`, `SET_LIT`, `SNETI`, `packet_t::sp`, `packet_t::stack_max`, `packet_t::stack_min`, `snaphdr::stack_sizeb`, `SUBI`, `SVCV`, and `value_t`.

Referenced by `main()`.

5.39.2.3 `packet_t* unmarshal_packet (char * buffer, int packet_lenb, int buf_lenb)`

Definition at line 63 of file `snap_io.c`.

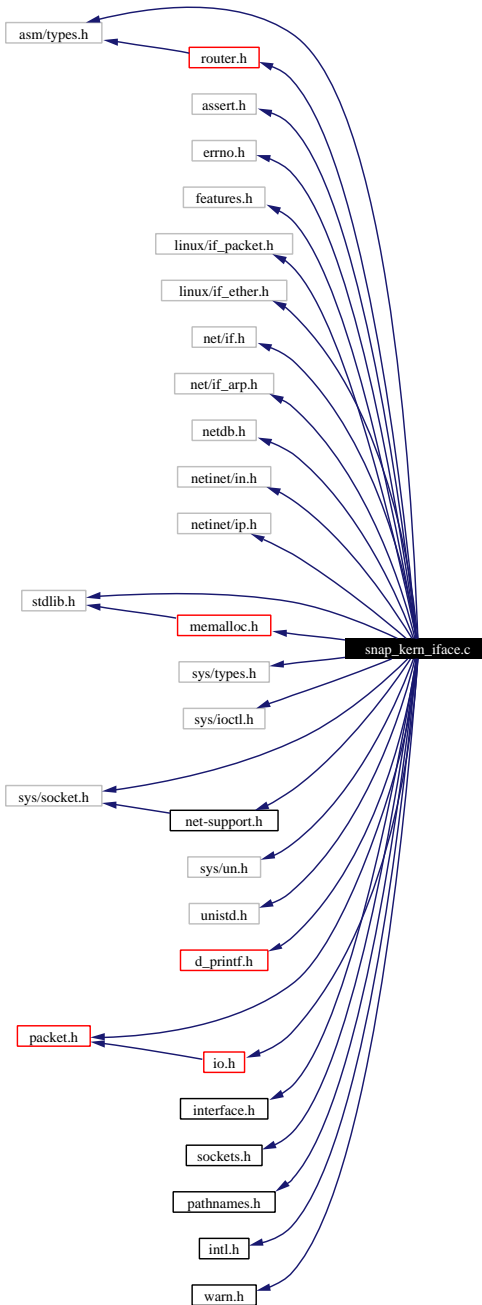
References `packet_t::code_max`, `packet_t::code_min`, `snaphdr::code_sizeb`, `d_printf()`, `snaphdr::entry_point`, `packet_t::h_alloc_heap_max`, `packet_t::handler`, `packet_t::hdr`, `packet_t::heap_max`, `packet_t::heap_min`, `snaphdr::heap_sizeb`, `instr_t`, `packet_t::iph`, `packet_t::is_contiguous`, `packet_t::pc`, `print_anti_timer`, `print_timer`, `packet_t::sp`, `packet_t::stack_max`, `packet_t::stack_min`, `snaphdr::stack_sizeb`, `uint32`, `value_t`, and `VERIFY`.

Referenced by `main()`, and `snap_recv_pkt()`.

5.40 snap-1.1-wjdb/lib/snap_kern_iface.c File Reference

```
#include <asm/types.h>
#include <assert.h>
#include <errno.h>
#include <features.h>
#include <linux/if_packet.h>
#include <linux/if_ether.h>
#include <net/if.h>
#include <net/if_arp.h>
#include <netdb.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ioctl.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <unistd.h>
#include "d_printf.h"
#include "packet.h"
#include "memalloc.h"
#include "interface.h"
#include "sockets.h"
#include "pathnames.h"
#include "intl.h"
#include "net-support.h"
#include "router.h"
#include "io.h"
#include "warn.h"
```

Include dependency graph for snap_kern_iface.c:



Defines

- `#define NIPQUAD(addr)`

5.40.1 Define Documentation

5.40.1.1 #define NIPQUAD(addr)

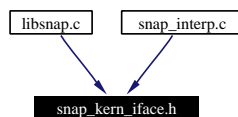
Value:

```
((unsigned char *)&(addr))[0], \  
    ((unsigned char *)&(addr))[1], \  
    ((unsigned char *)&(addr))[2], \  
    ((unsigned char *)&(addr))[3]
```

Definition at line 41 of file snap_kern_iface.c.

5.41 snap-1.1-wjdb/lib/snap_kern_iface.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define _SNAP_KERN_IFACE`

5.41.1 Define Documentation

5.41.1.1 `#define _SNAP_KERN_IFACE`

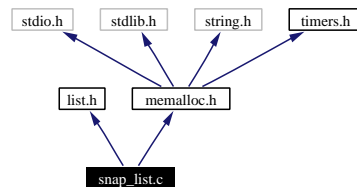
Definition at line 4 of file `snap_kern_iface.h`.

5.42 snap-1.1-wjdb/lib/snap_list.c File Reference

```
#include "list.h"
```

```
#include "memalloc.h"
```

Include dependency graph for snap_list.c:



Functions

- `list_t * cons` (`void *v`, `list_t *next`)
- `void free_list` (`list_t *list`)
- `int length_list` (`list_t *list`)

5.42.1 Function Documentation

5.42.1.1 `list_t* cons` (`void * v`, `list_t * next`)

Definition at line 24 of file snap_list.c.

5.42.1.2 `void free_list` (`list_t * list`)

Definition at line 41 of file snap_list.c.

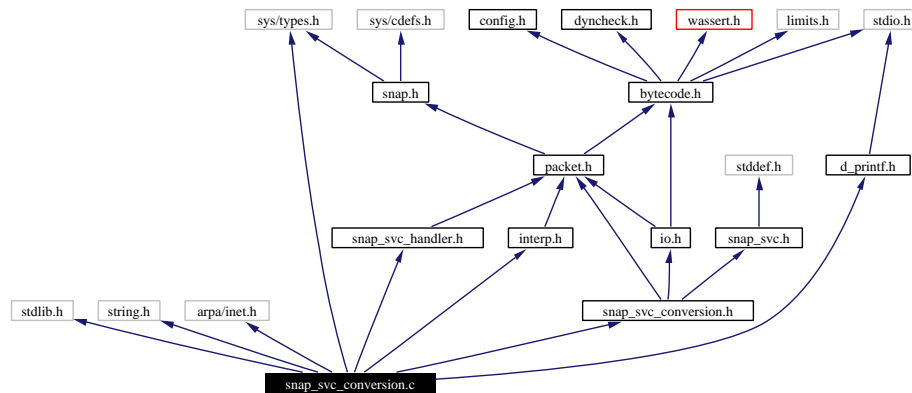
5.42.1.3 `int length_list` (`list_t * list`)

Definition at line 54 of file snap_list.c.

5.43 snap-1.1-wjdb/lib/snap_svc_conversion.c File Reference

```
#include <stdlib.h>
#include <string.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include "d_printf.h"
#include "interp.h"
#include "snap_svc_handler.h"
#include "snap_svc_conversion.h"
```

Include dependency graph for snap_svc_conversion.c:



Functions

- void * **snap_svc_convert_stack2returnstruct** (packet_t *p, value_t *src)
- int **snap_svc_convert_returnstruct2stack** (packet_t *p, value_t *dst, struct svc_returnitem *item)
- int **snap_svc_convert_direct2stack** (packet_t *p, void *value, int d_SvcType)
- void ** **snap_svc_convert_stack2arguments** (packet_t *p, int dArgCount)

5.43.1 Function Documentation

5.43.1.1 `int snap_svc_convert_direct2stack (packet_t * p, void * value, int dSvcType)`

Definition at line 188 of file snap_svc_conversion.c.

References `svc_returnitem::data`, `svc_returnitem::length`, `svc_returnitem::oid`, `svc_returnitem::oid_length`, `snap_svc_convert_returnstruct2stack()`, `packet_t::sp`, `SVC_SNMP_TYPE_LONG`, `SVC_SNMP_TYPE_STRING`, and `svc_returnitem::type`.

5.43.1.2 `int snap_svc_convert_returnstruct2stack (packet_t * p, value_t * dst, struct svc_returnitem * item)`

Definition at line 120 of file snap_svc_conversion.c.

References `ADDRV`, `d_printf()`, `svc_returnitem::data`, `heap_obj::flag`, `GET_ADDR_VAL`, `GET_INT`, `GET_OFFS`, `packet_t::heap_min`, `INTV`, `heap_obj::len`, `svc_returnitem::length`, `heap_obj::s`, `SET_ADDR`, `SET_INT`, `SET_OFFS`, `SET_TAG`, `packet_t::sp`, `packet_t::stack_max`, `STRV`, `SVC_SNMP_TYPE_ADDR`, `SVC_SNMP_TYPE_INT`, `SVC_SNMP_TYPE_LONG`, `SVC_SNMP_TYPE_STRING`, `svc_returnitem::type`, and `value_t`.

Referenced by `snap_svc_convert_direct2stack()`.

5.43.1.3 `void** snap_svc_convert_stack2arguments (packet_t * p, int dArgCount)`

Definition at line 220 of file snap_svc_conversion.c.

References `d_printf()`, `snap_svc_convert_stack2returnstruct()`, `packet_t::sp`, and `packet_t::stack_min`.

5.43.1.4 `void* snap_svc_convert_stack2returnstruct (packet_t * p, value_t * src)`

Definition at line 19 of file snap_svc_conversion.c.

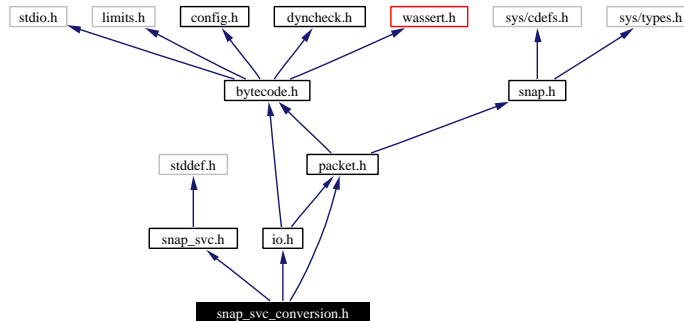
References `ADDRV`, `d_printf()`, `EXCV`, `FLOATV`, `GET_INT`, `GET_OFFS`, `GET_TAG`, `packet_t::heap_min`, `INTV`, `snap_htup::n`, `STRV`, `TUPLEV`, `snap_htup::vals`, and `value_t`.

Referenced by `snap_svc_convert_stack2arguments()`.

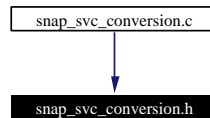
5.44 snap-1.1-wjdb/lib/snap_svc_conversion.h File Reference

```
#include <snap_svc.h>
#include "packet.h"
#include "io.h"
```

Include dependency graph for snap_svc_conversion.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct `snap_htup`
- struct `snap_hval`

Functions

- void * `snap_svc_convert_stack2returnstruct` (`packet_t *`, `value_t *`)
- int `snap_svc_convert_returnstruct2stack` (`packet_t *`, `value_t *`, `struct svc_returnitem *`)
- int `snap_svc_convert_direct2stack` (`packet_t *`, void *, int)
- void ** `snap_svc_convert_stack2arguments` (`packet_t *`, int)

5.44.1 Function Documentation

5.44.1.1 `int snap_svc_convert_direct2stack (packet_t *, void *, int)`

Definition at line 188 of file `snap_svc_conversion.c`.

References `svc_returnitem::data`, `svc_returnitem::length`, `svc_returnitem::oid`, `svc_returnitem::oid_length`, `snap_svc_convert_returnstruct2stack()`, `packet_t::sp`, `SVC_SNMP_TYPE_LONG`, `SVC_SNMP_TYPE_STRING`, and `svc_returnitem::type`.

5.44.1.2 `int snap_svc_convert_returnstruct2stack (packet_t *, value_t *, struct svc_returnitem *)`

Definition at line 120 of file `snap_svc_conversion.c`.

References `ADDRV`, `d_printf()`, `svc_returnitem::data`, `heap_obj::flag`, `GET_ADDR_VAL`, `GET_INT`, `GET_OFFS`, `packet_t::heap_min`, `INTV`, `heap_obj::len`, `svc_returnitem::length`, `heap_obj::s`, `SET_ADDR`, `SET_INT`, `SET_OFFS`, `SET_TAG`, `packet_t::sp`, `packet_t::stack_max`, `STRV`, `SVC_SNMP_TYPE_ADDR`, `SVC_SNMP_TYPE_INT`, `SVC_SNMP_TYPE_LONG`, `SVC_SNMP_TYPE_STRING`, `svc_returnitem::type`, and `value_t`.

Referenced by `snap_svc_convert_direct2stack()`.

5.44.1.3 `void** snap_svc_convert_stack2arguments (packet_t *, int)`

Definition at line 220 of file `snap_svc_conversion.c`.

References `d_printf()`, `snap_svc_convert_stack2returnstruct()`, `packet_t::sp`, and `packet_t::stack_min`.

5.44.1.4 `void* snap_svc_convert_stack2returnstruct (packet_t *, value_t *)`

Definition at line 19 of file `snap_svc_conversion.c`.

References `ADDRV`, `d_printf()`, `EXCV`, `FLOATV`, `GET_INT`, `GET_OFFS`, `GET_TAG`, `packet_t::heap_min`, `INTV`, `snap_htup::n`, `STRV`, `TUPLEV`, `snap_htup::vals`, and `value_t`.

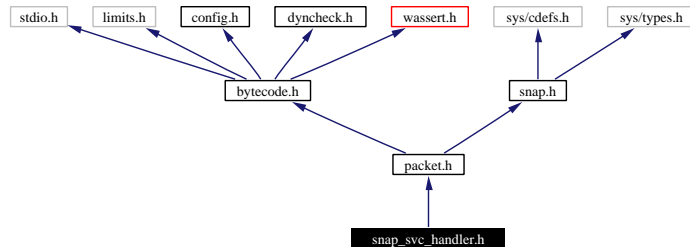
Referenced by `snap_svc_convert_stack2arguments()`.

5.45 snap-1.1-wjdb/lib/snap_svc_handler.c File Reference

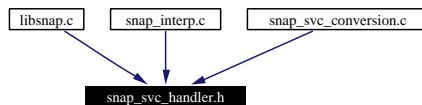
5.46 snap-1.1-wjdb/lib/snap_svc_handler.h File Reference

```
#include "packet.h"
```

Include dependency graph for snap_svc_handler.h:



This graph shows which files directly or indirectly include this file:



Functions

- void `snap_svc_handler_init` ()
- void `snap_svc_handler_close` ()
- void `snap_svc_handler_reinit` ()
- int `snap_svc_call_service` (`packet_t *p`, `char *name`)

5.46.1 Function Documentation

5.46.1.1 int `snap_svc_call_service` (`packet_t * p`, `char * name`)

5.46.1.2 void `snap_svc_handler_close` ()

5.46.1.3 void `snap_svc_handler_init` ()

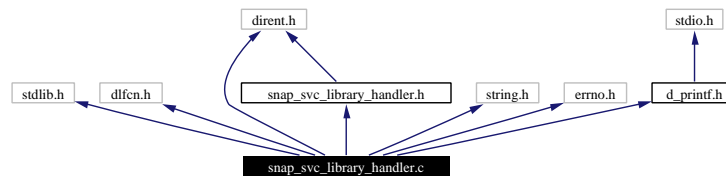
Referenced by `snap()`.

5.46.1.4 void `snap_svc_handler_reinit` ()

5.47 snap-1.1-wjdb/lib/snap_svc_library_handler.c File Reference

```
#include <stdlib.h>
#include <dlfcn.h>
#include <dirent.h>
#include <string.h>
#include <errno.h>
#include "d_printf.h"
#include "snap_svc_library_handler.h"
```

Include dependency graph for snap_svc_library_handler.c:



Functions

- int **snap_svc_logerrors** (int dLineNo)
- int **snap_svc_open** (void **hDll, char *strLib)
- int **snap_svc_bind** (void *hDll, char *strFunc, void **hFunc)
- int **snap_svc_close** (void **hDll)
- int **snap_svc_openmultiple_selector_empty** (const struct dirent *cureentry)
- int **snap_svc_openmultiple_selector_snapsvc** (const struct dirent *cureentry)
- int **snap_svc_openmultiple** (void ***hDllList, char *strDir, **snap_svc_fileselector** thisSelector)
- int **snap_svc_closemultiple** (void ***hDllList)

5.47.1 Function Documentation

5.47.1.1 int snap_svc_bind (void * hDll, char * strFunc, void ** hFunc)

Definition at line 61 of file snap_svc_library_handler.c.

References `d_printf()`, and `snap_svc_logerrors()`.

Referenced by `snap_svc_registerlib()`, and `snap_svc_unregisterlib()`.

5.47.1.2 int snap_svc_close (void ** *hDll*)

Definition at line 82 of file snap_svc_library_handler.c.

References `d_printf()`, and `snap_svc_logerrors()`.

Referenced by `snap_svc_closemultiple()`, and `snap_svc_unregisterlib()`.

5.47.1.3 int snap_svc_closemultiple (void * *hDllList*)**

Definition at line 158 of file snap_svc_library_handler.c.

References `hDllList`, and `snap_svc_close()`.

Referenced by `snap_svc_unregisteralllibs()`.

5.47.1.4 int snap_svc_logerrors (int *dLineNo*)

Definition at line 35 of file snap_svc_library_handler.c.

References `d_printf()`.

Referenced by `snap_svc_bind()`, `snap_svc_close()`, and `snap_svc_open()`.

5.47.1.5 int snap_svc_open (void ** *hDll*, char * *strLib*)

Definition at line 51 of file snap_svc_library_handler.c.

References `snap_svc_logerrors()`.

Referenced by `snap_svc_openmultiple()`.

5.47.1.6 int snap_svc_openmultiple (void * *hDllList*, char * *strDir*, snap_svc_fileselector *thisSelector*)**

Definition at line 124 of file snap_svc_library_handler.c.

References `d_printf()`, `hDllList`, `snap_svc_fileselector`, and `snap_svc_open()`.

Referenced by `snap_svc_registeralllibs()`.

5.47.1.7 int snap_svc_openmultiple_selector_empty (const struct dirent * *curentry*)

Definition at line 101 of file snap_svc_library_handler.c.

5.47.1.8 int snap_svc_openmultiple_selector_snapsvc (const struct dirent * *curentry*)

Definition at line 105 of file snap_svc_library_handler.c.

References `d_printf()`.

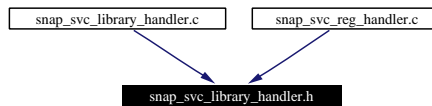
5.48 snap-1.1-wjdb/lib/snap_svc_library_handler.h File Reference

```
#include <dirent.h>
```

Include dependency graph for snap_svc_library_handler.h:



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef int(* **snap_svc_fileselector**)(const struct dirent *)

Functions

- int **snap_svc_logerrors** (int dLineNo)
- int **snap_svc_open** (void **hDll, char *strLib)
- int **snap_svc_bind** (void *hDll, char *strFunc, void **hFunc)
- int **snap_svc_close** (void **hDll)
- int **snap_svc_openmultiple_selector_empty** (const struct dirent *)
- int **snap_svc_openmultiple_selector_snapsvc** (const struct dirent *)
- int **snap_svc_openmultiple** (void ***hDllList, char *strDir, **snap_svc_fileselector** thisSelector)
- int **snap_svc_closemultiple** (void ***hDllList)

5.48.1 Typedef Documentation

5.48.1.1 typedef int(* **snap_svc_fileselector**)(const struct dirent *)

Definition at line 17 of file `snap_svc_library_handler.h`.

Referenced by `snap_svc_openmultiple()`.

5.48.2 Function Documentation

5.48.2.1 `int snap_svc_bind (void * hDll, char * strFunc, void ** hFunc)`

Definition at line 61 of file `snap_svc_library_handler.c`.

References `d_printf()`, and `snap_svc_logerrors()`.

Referenced by `snap_svc_registerlib()`, and `snap_svc_unregisterlib()`.

5.48.2.2 `int snap_svc_close (void ** hDll)`

Definition at line 82 of file `snap_svc_library_handler.c`.

References `d_printf()`, and `snap_svc_logerrors()`.

Referenced by `snap_svc_closemultiple()`, and `snap_svc_unregisterlib()`.

5.48.2.3 `int snap_svc_closemultiple (void *** hDllList)`

Definition at line 158 of file `snap_svc_library_handler.c`.

References `hDllList`, and `snap_svc_close()`.

Referenced by `snap_svc_unregisteralllibs()`.

5.48.2.4 `int snap_svc_logerrors (int dLineNo)`

Definition at line 35 of file `snap_svc_library_handler.c`.

References `d_printf()`.

Referenced by `snap_svc_bind()`, `snap_svc_close()`, and `snap_svc_open()`.

5.48.2.5 `int snap_svc_open (void ** hDll, char * strLib)`

Definition at line 51 of file `snap_svc_library_handler.c`.

References `snap_svc_logerrors()`.

Referenced by `snap_svc_openmultiple()`.

5.48.2.6 `int snap_svc_openmultiple (void *** hDllList, char * strDir, snap_svc_fileselector thisSelector)`

Definition at line 124 of file `snap_svc_library_handler.c`.

References `d_printf()`, `hDllList`, `snap_svc_fileselector`, and `snap_svc_open()`.

Referenced by `snap_svc_registeralllibs()`.

5.48.2.7 `int snap_svc_openmultiple_selector_empty (const struct dirent *)`

Definition at line 101 of file `snap_svc.library_handler.c`.

5.48.2.8 `int snap_svc_openmultiple_selector_snapsvc (const struct dirent *)`

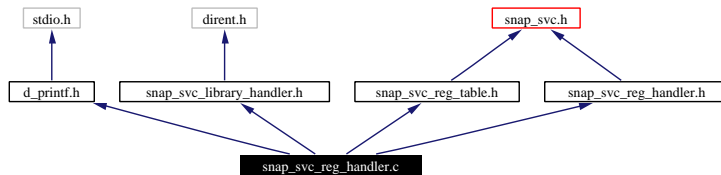
Definition at line 105 of file `snap_svc.library_handler.c`.

References `d_printf()`.

5.49 snap-1.1-wjdb/lib/snap_svc_reg_handler.c File Reference

```
#include "d_printf.h"
#include "snap_svc_library_handler.h"
#include "snap_svc_reg_table.h"
#include "snap_svc_reg_handler.h"
```

Include dependency graph for snap_svc_reg_handler.c:



Functions

- void **snap_svc_register_init** ()
- void **snap_svc_register_fini** ()
- void * **snap_svc_register_returnlaststruct** ()
- void * **snap_svc_register_freelaststruct** ()
- int **snap_svc_registerlib** (void *hDll)
- int **snap_svc_unregisterlib** (void *hDll)
- int **snap_svc_registeralllibs** (char *strDirectory)
- int **snap_svc_unregisteralllibs** ()
- void **snap_svc_registerall** ()
- void **snap_svc_unregisterall** ()
- void **snap_svc_reregisterall** ()

Variables

- int **snap_svc_register_initialized** = 0
- **tDllList hDllList** = NULL
- **snap_svc_getlastresult pReturnLast**
- **snap_svc_free_local_returnstruct pReturnFree**

5.49.1 Function Documentation

5.49.1.1 void snap_svc_register_fini ()

Definition at line 23 of file snap_svc_reg_handler.c.

References `d_printf()`, `snap_svc_register_initialized`, and `snap_svc_unregisterall()`.

5.49.1.2 `void* snap_svc_register_freelaststruct ()`

Definition at line 37 of file `snap_svc_reg_handler.c`.

References `pReturnFree`.

5.49.1.3 `void snap_svc_register_init ()`

Definition at line 18 of file `snap_svc_reg_handler.c`.

References `hDllList`, and `snap_svc_register_initialized`.

5.49.1.4 `void* snap_svc_register_returnlaststruct ()`

Definition at line 33 of file `snap_svc_reg_handler.c`.

References `pReturnLast`.

5.49.1.5 `void snap_svc_registerall ()`

Definition at line 127 of file `snap_svc_reg_handler.c`.

References `d_printf()`, and `snap_svc_registeralllibs()`.

Referenced by `snap_svc_reregisterall()`.

5.49.1.6 `int snap_svc_registeralllibs (char * strDirectory)`

Definition at line 88 of file `snap_svc_reg_handler.c`.

References `d_printf()`, `hDllList`, `snap_svc_openmultiple()`, and `snap_svc_registerlib()`.

Referenced by `snap_svc_registerall()`.

5.49.1.7 `int snap_svc_registerlib (void * hDll)`

Definition at line 41 of file `snap_svc_reg_handler.c`.

References `d_printf()`, `pReturnFree`, `pReturnLast`, `snap_svc_bind()`, `snap_svc_init`, `snap_svc_register`, `snap_svc_table_add()`, and `snapsvc_func_proto`.

Referenced by `snap_svc_registeralllibs()`.

5.49.1.8 `void snap_svc_reregisterall ()`

Definition at line 141 of file `snap_svc_reg_handler.c`.

References `snap_svc_registerall()`, and `snap_svc_unregisterall()`.

5.49.1.9 void snap_svc_unregisterall ()

Definition at line 137 of file `snap_svc_reg_handler.c`.

References `snap_svc_unregisteralllibs()`.

Referenced by `snap_svc_register_fini()`, and `snap_svc_reregisterall()`.

5.49.1.10 int snap_svc_unregisteralllibs ()

Definition at line 110 of file `snap_svc_reg_handler.c`.

References `d_printf()`, `hDllList`, `snap_svc_closemultiple()`, and `snap_svc_unregisterlib()`.

Referenced by `snap_svc_unregisterall()`.

5.49.1.11 int snap_svc_unregisterlib (void * hDll)

Definition at line 74 of file `snap_svc_reg_handler.c`.

References `d_printf()`, `snap_svc_bind()`, `snap_svc_close()`, and `snap_svc_init`.

Referenced by `snap_svc_unregisteralllibs()`.

5.49.2 Variable Documentation

5.49.2.1 tDllList hDllList = NULL

Definition at line 14 of file `snap_svc_reg_handler.c`.

Referenced by `snap_svc_closemultiple()`, `snap_svc_openmultiple()`, `snap_svc_register_init()`, `snap_svc_registeralllibs()`, and `snap_svc_unregisteralllibs()`.

5.49.2.2 snap_svc_free_local_returnstruct pReturnFree

Definition at line 16 of file `snap_svc_reg_handler.c`.

Referenced by `snap_svc_register_freelaststruct()`, and `snap_svc_registerlib()`.

5.49.2.3 snap_svc_getlastresult pReturnLast

Definition at line 15 of file `snap_svc_reg_handler.c`.

Referenced by `snap_svc_register_returnlaststruct()`, and `snap_svc_registerlib()`.

5.49.2.4 `int snap_svc_register_initialized = 0`

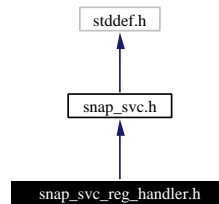
Definition at line 13 of file `snap_svc_reg_handler.c`.

Referenced by `snap_svc_register_fini()`, and `snap_svc_register_init()`.

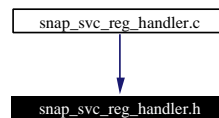
5.50 snap-1.1-wjdb/lib/snap_svc_reg_handler.h File Reference

```
#include <snap_svc.h>
```

Include dependency graph for snap_svc_reg_handler.h:



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef void * **tDll**
- typedef **tDll** * **tDllList**

Functions

- void **snap_svc_register_init** ()
- void **snap_svc_register_fini** ()
- void * **snap_svc_register_returnlaststruct** ()
- void * **snap_svc_register_freelaststruct** ()
- int **snap_svc_registerlib** (void *hDll)
- int **snap_svc_unregisterlib** (void *hDll)
- int **snap_svc_registeralllibs** (char *strDirectory)
- int **snap_svc_unregisteralllibs** ()
- void **snap_svc_registerall** ()
- void **snap_svc_unregisterall** ()
- void **snap_svc_reregisterall** ()

5.50.1 Typedef Documentation

5.50.1.1 typedef void* tDll

Definition at line 12 of file snap_svc_reg_handler.h.

5.50.1.2 typedef tDll* tDllList

Definition at line 13 of file snap_svc_reg_handler.h.

5.50.2 Function Documentation

5.50.2.1 void snap_svc_register_fini ()

Definition at line 23 of file snap_svc_reg_handler.c.

References d_printf(), snap_svc_register_initialized, and snap_svc_unregisterall().

5.50.2.2 void* snap_svc_register_freelaststruct ()

Definition at line 37 of file snap_svc_reg_handler.c.

References pReturnFree.

5.50.2.3 void snap_svc_register_init ()

Definition at line 18 of file snap_svc_reg_handler.c.

References hDllList, and snap_svc_register_initialized.

5.50.2.4 void* snap_svc_register_returnlaststruct ()

Definition at line 33 of file snap_svc_reg_handler.c.

References pReturnLast.

5.50.2.5 void snap_svc_registerall ()

Definition at line 127 of file snap_svc_reg_handler.c.

References d_printf(), and snap_svc_registeralllibs().

Referenced by snap_svc_reregisterall().

5.50.2.6 int snap_svc_registeralllibs (char * *strDirectory*)

Definition at line 88 of file snap_svc_reg_handler.c.

References `d_printf()`, `hDllList`, `snap_svc_openmultiple()`, and `snap_svc_registerlib()`.

Referenced by `snap_svc_registerall()`.

5.50.2.7 `int snap_svc_registerlib (void * hDll)`

Definition at line 41 of file `snap_svc_reg_handler.c`.

References `d_printf()`, `pReturnFree`, `pReturnLast`, `snap_svc_bind()`, `snap_svc_init`, `snap_svc_register`, `snap_svc_table_add()`, and `snapsvc_func_proto`.

Referenced by `snap_svc_registeralllibs()`.

5.50.2.8 `void snap_svc_reregisterall ()`

Definition at line 141 of file `snap_svc_reg_handler.c`.

References `snap_svc_registerall()`, and `snap_svc_unregisterall()`.

5.50.2.9 `void snap_svc_unregisterall ()`

Definition at line 137 of file `snap_svc_reg_handler.c`.

References `snap_svc_unregisteralllibs()`.

Referenced by `snap_svc_register_fini()`, and `snap_svc_reregisterall()`.

5.50.2.10 `int snap_svc_unregisteralllibs ()`

Definition at line 110 of file `snap_svc_reg_handler.c`.

References `d_printf()`, `hDllList`, `snap_svc_closemultiple()`, and `snap_svc_unregisterlib()`.

Referenced by `snap_svc_unregisterall()`.

5.50.2.11 `int snap_svc_unregisterlib (void * hDll)`

Definition at line 74 of file `snap_svc_reg_handler.c`.

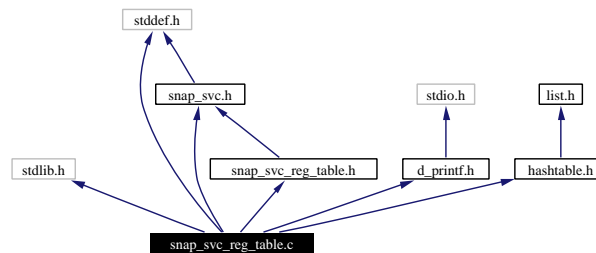
References `d_printf()`, `snap_svc_bind()`, `snap_svc_close()`, and `snap_svc_init`.

Referenced by `snap_svc_unregisteralllibs()`.

5.51 snap-1.1-wjdb/lib/snap_svc_reg_table.c File Reference

```
#include <stdlib.h>
#include <stddef.h>
#include <snap_svc.h>
#include "d_printf.h"
#include "hashtable.h"
#include "snap_svc_reg_table.h"
```

Include dependency graph for snap_svc_reg_table.c:



Functions

- int **mystrcmp** (char *s1, char *s2)
- void * **snap_svc_table_find** (char *strName)
- int **snap_svc_table_add** (char *strName, **snapsvc_func_proto** pFunc, int nargs, int nret)
- int **snap_svc_table_init** ()
- int **snap_svc_table_fini** ()

Variables

- int **snap_svc_table_initialized** = 0

5.51.1 Function Documentation

5.51.1.1 int mystrcmp (char * s1, char * s2)

Definition at line 21 of file snap_svc_reg_table.c.

Referenced by snap_svc_table_init().

**5.51.1.2 int snap_svc_table_add (char * *strName*,
snapsvc_func_proto *pFunc*, int *nargs*, int *nret*)**

Definition at line 42 of file snap_svc_reg_table.c.

References `d_printf()`, `ht_insert()`, `snap_svc_rec::nargs`, `snap_svc_rec::nret`, `snap_svc_table_init()`, `snap_svc_table_initialized`, `snap_svc_rec::snapsvc_func`, and `snapsvc_func_proto`.

Referenced by `snap_svc_registerlib()`.

5.51.1.3 void* snap_svc_table_find (char * *strName*)

Definition at line 37 of file snap_svc_reg_table.c.

References `ht_lookup()`.

5.51.1.4 int snap_svc_table_fini ()

Definition at line 92 of file snap_svc_reg_table.c.

References `snap_svc_table_initialized`.

5.51.1.5 int snap_svc_table_init ()

Definition at line 75 of file snap_svc_reg_table.c.

References `d_printf()`, `DEF_SVC_TAB_SZ`, `ht_create()`, `mystrcmp()`, and `snap_svc_table_initialized`.

Referenced by `snap_svc_table_add()`.

5.51.2 Variable Documentation**5.51.2.1 int snap_svc_table_initialized = 0**

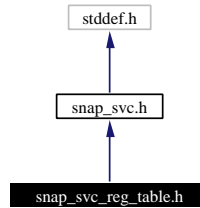
Definition at line 18 of file snap_svc_reg_table.c.

Referenced by `snap_svc_table_add()`, `snap_svc_table_fini()`, and `snap_svc_table_init()`.

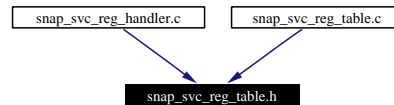
5.52 snap-1.1-wjdb/lib/snap_svc_reg_table.h File Reference

```
#include <snap_svc.h>
```

Include dependency graph for snap_svc_reg_table.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **snap_svc_rec**

Defines

- #define **DEF_SVC_TAB_SZ** 100

Functions

- int **snap_svc_table_init** ()
- int **snap_svc_table_fini** ()
- int **snap_svc_table_add** (char *, **snapsvc_func_proto**, int, int)
- void * **snap_svc_table_find** (char *)

5.52.1 Define Documentation

5.52.1.1 #define DEF_SVC_TAB_SZ 100

Definition at line 12 of file snap_svc_reg_table.h.

Referenced by snap_svc_table_init().

5.52.2 Function Documentation

5.52.2.1 `int snap_svc_table_add (char *, snapsvc_func_proto, int, int)`

Definition at line 42 of file `snap_svc_reg_table.c`.

References `d_printf()`, `ht_insert()`, `snap_svc_rec::nargs`, `snap_svc_rec::nret`, `snap_svc_table_init()`, `snap_svc_table_initialized`, `snap_svc_rec::snapsvc_func`, and `snapsvc_func_proto`.

Referenced by `snap_svc_registerlib()`.

5.52.2.2 `void* snap_svc_table_find (char *)`

Definition at line 37 of file `snap_svc_reg_table.c`.

References `ht_lookup()`.

5.52.2.3 `int snap_svc_table_fini ()`

Definition at line 92 of file `snap_svc_reg_table.c`.

References `snap_svc_table_initialized`.

5.52.2.4 `int snap_svc_table_init ()`

Definition at line 75 of file `snap_svc_reg_table.c`.

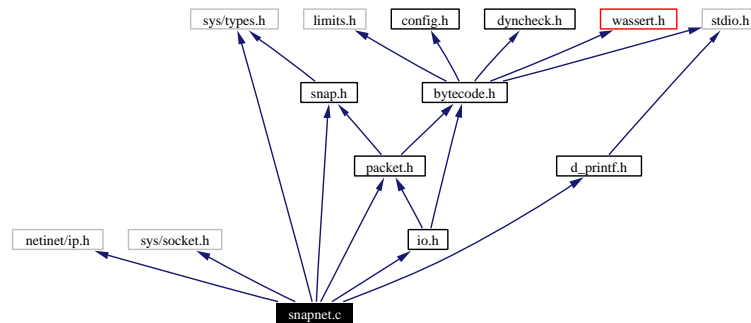
References `d_printf()`, `DEF_SVC_TAB_SZ`, `ht_create()`, `mystrcmp()`, and `snap_svc_table_initialized`.

Referenced by `snap_svc_table_add()`.

5.53 snap-1.1-wjdb/lib/snapnet.c File Reference

```
#include <netinet/ip.h>
#include <sys/socket.h>
#include <sys/types.h>
#include "io.h"
#include "packet.h"
#include "snap.h"
#include "d_printf.h"
```

Include dependency graph for snapnet.c:



Defines

- `#define MAX_MTU 3924`

Functions

- `int snap_recv_pkt (int sock, packet_t **p)`

Variables

- `char pbuf [3 *MAX_MTU]`

5.53.1 Define Documentation

5.53.1.1 `#define MAX_MTU 3924`

Definition at line 11 of file snapnet.c.

5.53.2 Function Documentation

5.53.2.1 int snap_rcv_pkt (int *sock*, packet_t ** *p*)

Definition at line 15 of file snapnet.c.

References len, pbuf, and unmarshal_packet().

Referenced by handle_snap_request(), and snap_receive().

5.53.3 Variable Documentation

5.53.3.1 char pbuf[3 * MAX_MTU]

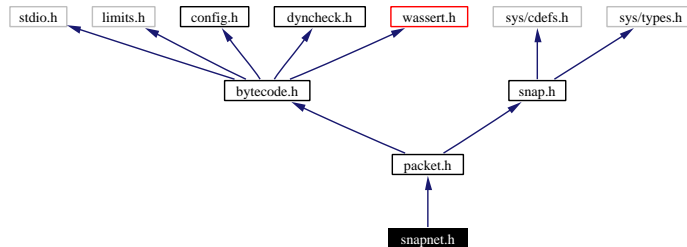
Definition at line 13 of file snapnet.c.

Referenced by snap_rcv_pkt().

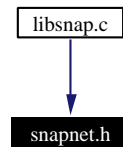
5.54 snap-1.1-wjdb/lib/snapnet.h File Reference

```
#include "packet.h"
```

Include dependency graph for snapnet.h:



This graph shows which files directly or indirectly include this file:



Functions

- int **snap_rcv_pkt** (int sock, **packet_t** **p)

5.54.1 Function Documentation

5.54.1.1 int **snap_rcv_pkt** (int *sock*, **packet_t** ** *p*)

Definition at line 15 of file snapnet.c.

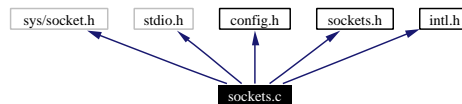
References len, pbuf, and unmarshal_packet().

Referenced by handle_snap_request(), and snap_receive().

5.55 snap-1.1-wjdb/lib/sockets.c File Reference

```
#include <sys/socket.h>
#include <stdio.h>
#include "config.h"
#include "sockets.h"
#include "intl.h"
```

Include dependency graph for sockets.c:



Functions

- int `sockets_open` (void)

Variables

- int `skfd` = -1

5.55.1 Function Documentation

5.55.1.1 int `sockets_open` (void)

Definition at line 33 of file sockets.c.

References `_`, `ax25_sock`, `ddp_sock`, `ec_sock`, `inet6_sock`, `inet_sock`, `ipx_sock`, and `rose_sock`.

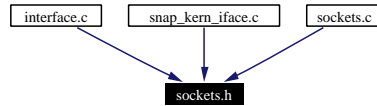
5.55.2 Variable Documentation

5.55.2.1 int `skfd` = -1

Definition at line 10 of file sockets.c.

5.56 snap-1.1-wjdb/lib/sockets.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int `sockets_open` (void)

Variables

- int `skfd`
- int `ipx_sock`
- int `ax25_sock`
- int `rose_sock`
- int `inet_sock`
- int `inet6_sock`
- int `ddp_sock`
- int `ec_sock`

5.56.1 Function Documentation

5.56.1.1 int `sockets_open` (void)

Definition at line 33 of file `sockets.c`.

References `_`, `ax25_sock`, `ddp_sock`, `ec_sock`, `inet6_sock`, `inet_sock`, `ipx_sock`, and `rose_sock`.

5.56.2 Variable Documentation

5.56.2.1 int `ax25_sock`

Definition at line 1 of file `sockets.h`.

Referenced by `sockets_open()`.

5.56.2.2 int `ddp_sock`

Definition at line 1 of file `sockets.h`.

Referenced by `if_fetch()`, and `sockets_open()`.

5.56.2.3 int ec_sock

Definition at line 1 of file sockets.h.

Referenced by `if_fetch()`, and `sockets_open()`.

5.56.2.4 int inet6_sock

Definition at line 1 of file sockets.h.

Referenced by `sockets_open()`.

5.56.2.5 int inet_sock

Definition at line 1 of file sockets.h.

Referenced by `if_fetch()`, and `sockets_open()`.

5.56.2.6 int ipx_sock

Definition at line 1 of file sockets.h.

Referenced by `if_fetch()`, and `sockets_open()`.

5.56.2.7 int rose_sock

Definition at line 1 of file sockets.h.

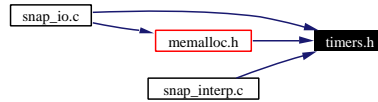
Referenced by `sockets_open()`.

5.56.2.8 int skfd

Definition at line 1 of file sockets.h.

5.57 snap-1.1-wjdb/lib/timers.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define **print_timer**(index, string)
- #define **print_anti_timer**(index, string)
- #define **print_mtimer**(index, string)
- #define **print_anti_mtimer**(index, string)

Functions

- void **init_all_timers** (void)
- void **dump_all_timers** (void)
- void **internal_print_time** (int index, char *label)
- void **internal_print_anti_time** (int index, char *label)

Variables

- char * **print_flags**
- int **print_flag_count**
- int **do_print_individual_timers**
- int **do_print_item_messages**
- int **do_print_antitimers**

5.57.1 Define Documentation

5.57.1.1 #define print_anti_mtimer(index, string)

Value:

```

if ((print_flags[index] == '1') && (index < print_flag_count)) { \
    internal_print_anti_time(index, string); \
}
  
```

Definition at line 95 of file timers.h.

5.57.1.2 #define print_anti_timer(index, string)

Value:

```
if ((print_flags[index] == '1') && (index < print_flag_count)) { \
    internal_print_anti_time(index, string); \
}
```

Definition at line 83 of file timers.h.

Referenced by marshal_packet(), nexthop(), and unmarshal_packet().

5.57.1.3 #define print_mtimer(index, string)

Value:

```
if ((print_flags[index] == '1') && (index < print_flag_count)) { \
    internal_print_time(index, string); \
}
```

Definition at line 90 of file timers.h.

5.57.1.4 #define print_timer(index, string)

Value:

```
if ((print_flags[index] == '1') && (index < print_flag_count)) { \
    internal_print_time(index, string); \
}
```

Definition at line 78 of file timers.h.

Referenced by marshal_packet(), nexthop(), and unmarshal_packet().

5.57.2 Function Documentation

5.57.2.1 void dump_all_timers (void)

5.57.2.2 void init_all_timers (void)

5.57.2.3 void internal_print_anti_time (int *index*, char * *label*)

5.57.2.4 void internal_print_time (int *index*, char * *label*)

5.57.3 Variable Documentation

5.57.3.1 int do_print_antitimers

Definition at line 60 of file timers.h.

5.57.3.2 int do_print_individual_timers

Definition at line 58 of file timers.h.

5.57.3.3 int do_print_item_messages

Definition at line 59 of file timers.h.

5.57.3.4 int print_flag_count

Definition at line 57 of file timers.h.

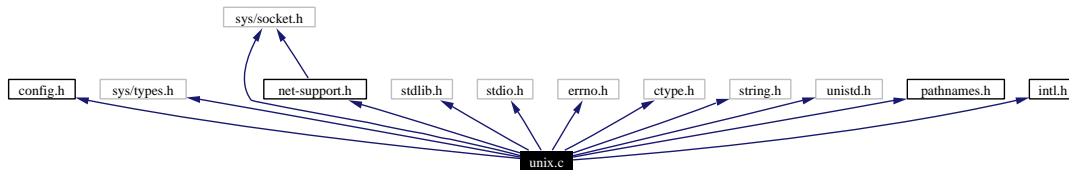
5.57.3.5 char* print_flags

Definition at line 56 of file timers.h.

5.58 snap-1.1-wjdb/lib/unix.c File Reference

```
#include "config.h"
#include <sys/types.h>
#include <sys/socket.h>
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <ctype.h>
#include <string.h>
#include <unistd.h>
#include "net-support.h"
#include "pathnames.h"
#include "intl.h"
```

Include dependency graph for unix.c:



Variables

- `aftype unspec_aftype`

5.58.1 Variable Documentation

5.58.1.1 struct `aftype unspec_aftype`

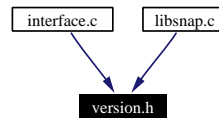
Initial value:

```
{
  "unspec", NULL,      AF_UNSPEC,  0,
  UNSPEC_print, UNSPEC_sprint,  NULL,    NULL,
  NULL
}
```

Definition at line 92 of file `unix.c`.

5.59 snap-1.1-wjdb/lib/version.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define SNAP_VERSION "1.1"`

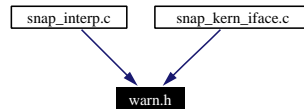
5.59.1 Define Documentation

5.59.1.1 `#define SNAP_VERSION "1.1"`

Definition at line 6 of file version.h.

5.60 snap-1.1-wjdb/lib/warn.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define warn(fmt, arg...) fprintf(stderr,fmt,##arg)`

5.60.1 Define Documentation

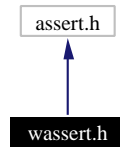
5.60.1.1 `#define warn(fmt, arg...) fprintf(stderr,fmt,##arg)`

Definition at line 9 of file warn.h.

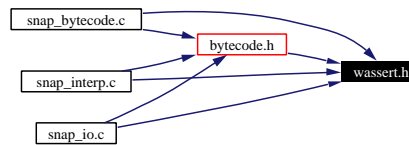
5.61 snap-1.1-wjdb/lib/wassert.h File Reference

```
#include <assert.h>
```

Include dependency graph for wassert.h:



This graph shows which files directly or indirectly include this file:



Defines

- `#define wassert(e)`

5.61.1 Define Documentation

5.61.1.1 `#define wassert(e)`

Value:

```

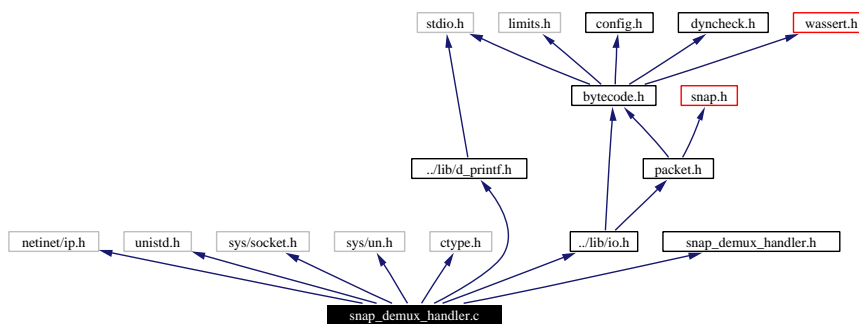
if (e);
else {
    fprintf(stderr,"%s:%d: soft-assertion failed in %s\n",
        __FILE__,__LINE__,__STRING(e));
}
  
```

Definition at line 15 of file wassert.h.

5.62 snap-1.1-wjdb/src/snap_demux_handler.c File Reference

```
#include <netinet/ip.h>
#include <unistd.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <ctype.h>
#include "../lib/d_printf.h"
#include "../lib/io.h"
#include "snap_demux_handler.h"
```

Include dependency graph for snap_demux_handler.c:



Functions

- void **snap_demux_close_unix** ()
- int **snap_demux_init_unix** ()
- void **snap_demux_close_rawip** ()
- int **snap_demux_init_rawip** ()
- void **snap_demux_close_udp** ()
- int **snap_demux_init_udp** ()
- void **snap_demux_buffer_noop** (char *pbuf)
- void **snap_demux_buffer_print_unsafe** (char *pbuf)
- void **snap_demux_buffer_print** (char *pbuf)
- void **snap_demux_close** ()
- int **snap_demux_init** (int protocols)
- int **snap_demux_receivefrom** (int socket_waiting, **buffer_handler** active_handler)
- int **snap_demux_receive** (**buffer_handler** active_handler)
- int **snap_demux_select** ()
- int **snap_demux_handler** (**buffer_handler** active_handler)

Variables

- int `socket_unix` = -1
- int `socket_rawip` = -1
- int `socket_udp` = -1
- int `protocols_internal` = 0
- int `max_filedes` = -1
- fd_set `fdset`

5.62.1 Function Documentation

5.62.1.1 void `snap_demux_buffer_noop` (char * *pbuf*)

Definition at line 132 of file `snap_demux_handler.c`.

5.62.1.2 void `snap_demux_buffer_print` (char * *pbuf*)

Definition at line 139 of file `snap_demux_handler.c`.

5.62.1.3 void `snap_demux_buffer_print_unsafe` (char * *pbuf*)

Definition at line 135 of file `snap_demux_handler.c`.

5.62.1.4 void `snap_demux_close` ()

Definition at line 170 of file `snap_demux_handler.c`.

References `max_filedes`, `protocols_internal`, `snap_demux_close_rawip()`, `snap_demux_close_udp()`, `snap_demux_close_unix()`, `SNAP_RAWIP`, `SNAP_UDP`, and `SNAP_UNIX`.

Referenced by `main()`, and `snap_demux_init()`.

5.62.1.5 void `snap_demux_close_rawip` ()

Definition at line 65 of file `snap_demux_handler.c`.

References `socket_rawip`.

Referenced by `snap_demux_close()`, and `snap_demux_init_rawip()`.

5.62.1.6 void `snap_demux_close_udp` ()

Definition at line 98 of file `snap_demux_handler.c`.

References `socket_udp`.

Referenced by `snap_demux_close()`, and `snap_demux_init_udp()`.

5.62.1.7 void snap_demux_close_unix ()

Definition at line 27 of file snap_demux_handler.c.

References socket_unix.

Referenced by snap_demux_close(), and snap_demux_init_unix().

5.62.1.8 int snap_demux_handler (buffer_handler active_handler)

Definition at line 297 of file snap_demux_handler.c.

References buffer_handler, snap_demux_receive(), and snap_demux_select().

Referenced by main().

5.62.1.9 int snap_demux_init (int protocols)

Definition at line 188 of file snap_demux_handler.c.

References d_printf(), fdset, max_filedes, protocols_internal, snap_demux_close(), snap_demux_init_rawip(), snap_demux_init_udp(), snap_demux_init_unix(), SNAP_RAWIP, SNAP_UDP, SNAP_UNIX, socket_rawip, socket_udp, and socket_unix.

Referenced by main().

5.62.1.10 int snap_demux_init_rawip ()

Definition at line 73 of file snap_demux_handler.c.

References bindaddr, snap_demux_close_rawip(), and socket_rawip.

Referenced by snap_demux_init().

5.62.1.11 int snap_demux_init_udp ()

Definition at line 106 of file snap_demux_handler.c.

References bindaddr, snap_demux_close_udp(), and socket_udp.

Referenced by snap_demux_init().

5.62.1.12 int snap_demux_init_unix ()

Definition at line 40 of file snap_demux_handler.c.

References snap_demux_close_unix(), and socket_unix.

Referenced by snap_demux_init().

5.62.1.13 int snap_demux_receive (buffer_handler active_handler)

Definition at line 251 of file snap_demux_handler.c.

References `buffer_handler`, `d_printf_timed()`, `fdset`, `protocols_internal`, `snap_demux_receivefrom()`, `SNAP_RAWIP`, `SNAP_UDP`, `SNAP_UNIX`, `socket_rawip`, `socket_udp`, and `socket_unix`.

Referenced by `snap_demux_handler()`.

5.62.1.14 int snap_demux_receivefrom (int socket_waiting, buffer_handler active_handler)

Definition at line 222 of file snap_demux_handler.c.

References `buffer_handler`, `d_printf()`, `d_printf_timed()`, and `SNAP_BUFLLEN`.

Referenced by `snap_demux_receive()`.

5.62.1.15 int snap_demux_select ()

Definition at line 269 of file snap_demux_handler.c.

References `d_printf_timed()`, `fdset`, `max_filedes`, `protocols_internal`, `SNAP_RAWIP`, `SNAP_UDP`, `SNAP_UNIX`, `socket_rawip`, `socket_udp`, and `socket_unix`.

Referenced by `snap_demux_handler()`.

5.62.2 Variable Documentation**5.62.2.1 fd_set fdset**

Definition at line 21 of file snap_demux_handler.c.

Referenced by `snap_demux_init()`, `snap_demux_receive()`, `snap_demux_select()`, and `snap_external_svclib_snmp_execpdu()`.

5.62.2.2 int max_filedes = -1

Definition at line 20 of file snap_demux_handler.c.

Referenced by `snap_demux_close()`, `snap_demux_init()`, and `snap_demux_select()`.

5.62.2.3 int protocols_internal = 0

Definition at line 19 of file snap_demux_handler.c.

Referenced by `snap_demux_close()`, `snap_demux_init()`, `snap_demux_receive()`, and `snap_demux_select()`.

5.62.2.4 int socket_rawip = -1

Definition at line 16 of file snap_demux_handler.c.

Referenced by snap_demux_close_rawip(), snap_demux_init(), snap_demux_init_rawip(), snap_demux_receive(), and snap_demux_select().

5.62.2.5 int socket_udp = -1

Definition at line 17 of file snap_demux_handler.c.

Referenced by snap_demux_close_udp(), snap_demux_init(), snap_demux_init_udp(), snap_demux_receive(), and snap_demux_select().

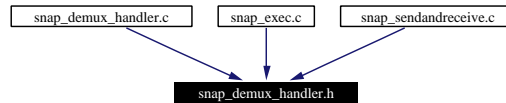
5.62.2.6 int socket_unix = -1

Definition at line 15 of file snap_demux_handler.c.

Referenced by snap_demux_close_unix(), snap_demux_init(), snap_demux_init_unix(), snap_demux_receive(), and snap_demux_select().

5.63 snap-1.1-wjdb/src/snap_demux_handler.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define SNAP_BUFLEN 3924`
- `#define SNAP_UNIX 0x1`
- `#define SNAP_RAWIP 0x2`
- `#define SNAP_UDP 0x4`

Typedefs

- `typedef void(* buffer_handler)(char *)`

Functions

- `int snap_demux_init (int)`
- `int snap_demux_receive (buffer_handler)`
- `int snap_demux_select ()`
- `void snap_demux_close ()`
- `int snap_demux_handler (buffer_handler)`
- `void snap_demux_buffer_noop (char *)`
- `void snap_demux_buffer_print (char *)`
- `void snap_demux_buffer_print_unsafe (char *)`

Variables

- `short int receiveport`

5.63.1 Define Documentation

5.63.1.1 `#define SNAP_BUFLEN 3924`

Definition at line 6 of file `snap_demux_handler.h`.

Referenced by `snap_demux_receivefrom()`.

5.63.1.2 `#define SNAP_RAWIP 0x2`

Definition at line 10 of file snap_demux_handler.h.

Referenced by snap_demux_close(), snap_demux_init(), snap_demux_receive(), and snap_demux_select().

5.63.1.3 `#define SNAP_UDP 0x4`

Definition at line 11 of file snap_demux_handler.h.

Referenced by main(), snap_demux_close(), snap_demux_init(), snap_demux_receive(), and snap_demux_select().

5.63.1.4 `#define SNAP_UNIX 0x1`

Definition at line 9 of file snap_demux_handler.h.

Referenced by main(), snap_demux_close(), snap_demux_init(), snap_demux_receive(), and snap_demux_select().

5.63.2 Typedef Documentation

5.63.2.1 `typedef void(* buffer_handler)(char*)`

Definition at line 14 of file snap_demux_handler.h.

Referenced by main(), snap_demux_handler(), snap_demux_receive(), and snap_demux_receivefrom().

5.63.3 Function Documentation

5.63.3.1 `void snap_demux_buffer_noop (char *)`

Definition at line 132 of file snap_demux_handler.c.

5.63.3.2 `void snap_demux_buffer_print (char *)`

Definition at line 139 of file snap_demux_handler.c.

5.63.3.3 `void snap_demux_buffer_print_unsafe (char *)`

Definition at line 135 of file snap_demux_handler.c.

5.63.3.4 `void snap_demux_close ()`

Definition at line 170 of file snap_demux_handler.c.

References `max_filedes`, `protocols_internal`, `snap_demux_close_rawip()`, `snap_demux_close_udp()`, `snap_demux_close_unix()`, `SNAP_RAWIP`, `SNAP_UDP`, and `SNAP_UNIX`.

Referenced by `main()`, and `snap_demux_init()`.

5.63.3.5 int snap_demux_handler (buffer_handler)

Definition at line 297 of file `snap_demux_handler.c`.

References `buffer_handler`, `snap_demux_receive()`, and `snap_demux_select()`.

Referenced by `main()`.

5.63.3.6 int snap_demux_init (int)

Definition at line 188 of file `snap_demux_handler.c`.

References `d_printf()`, `fdset`, `max_filedes`, `protocols_internal`, `snap_demux_close()`, `snap_demux_init_rawip()`, `snap_demux_init_udp()`, `snap_demux_init_unix()`, `SNAP_RAWIP`, `SNAP_UDP`, `SNAP_UNIX`, `socket_rawip`, `socket_udp`, and `socket_unix`.

Referenced by `main()`.

5.63.3.7 int snap_demux_receive (buffer_handler)

Definition at line 251 of file `snap_demux_handler.c`.

References `buffer_handler`, `d_printf_timed()`, `fdset`, `protocols_internal`, `snap_demux_receivefrom()`, `SNAP_RAWIP`, `SNAP_UDP`, `SNAP_UNIX`, `socket_rawip`, `socket_udp`, and `socket_unix`.

Referenced by `snap_demux_handler()`.

5.63.3.8 int snap_demux_select ()

Definition at line 269 of file `snap_demux_handler.c`.

References `d_printf_timed()`, `fdset`, `max_filedes`, `protocols_internal`, `SNAP_RAWIP`, `SNAP_UDP`, `SNAP_UNIX`, `socket_rawip`, `socket_udp`, and `socket_unix`.

Referenced by `snap_demux_handler()`.

5.63.4 Variable Documentation

5.63.4.1 short int receiveport

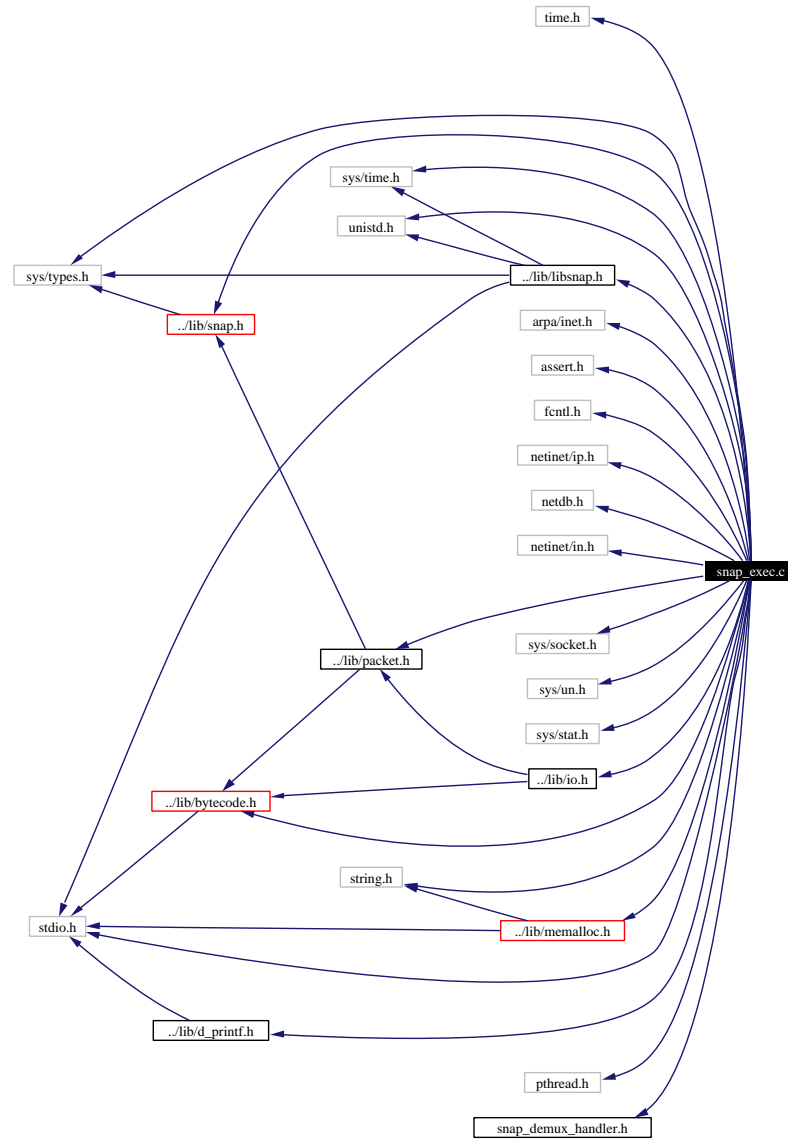
Definition at line 17 of file `snap_demux_handler.h`.

Referenced by `init_request()`, and `parse_cmdline()`.

5.64 snap-1.1-wjdb/src/snap_exec.c File Reference

```
#include <time.h>
#include <sys/time.h>
#include "../lib/libsnap.h"
#include <arpa/inet.h>
#include <assert.h>
#include <fcntl.h>
#include <netinet/ip.h>
#include <netdb.h>
#include <netinet/in.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <sys/stat.h>
#include <sys/types.h>
#include "../lib/snap.h"
#include "../lib/bytecode.h"
#include "../lib/packet.h"
#include "../lib/d_printf.h"
#include "../lib/memalloc.h"
#include "../lib/io.h"
#include <pthread.h>
#include "snap_demux_handler.h"
```

Include dependency graph for snap_exec.c:



Defines

- #define **NIPQUAD**(addr)
- #define **IPPROTO_SNAP** 130

Functions

- char * **basename** (const char *)
- void **parse_cmdline** (int argc, char **argv)
- int **compare_longints** (const void *a, const void *b)
- int **init_request** (int argc, char **argv)

- void **sendpkt** ()
- void **usage** (int argc, char **argv)
- int **main** (int argc, char **argv)

Variables

- unsigned char **out_ttl** = 32
- short int **receiveport** = 7777
- sockaddr_in **destaddr**
- sockaddr_in **srcaddr**
- sockaddr_in **localaddr**
- int **infd**
- **buffer_t** **inbuf**
- int **sd**

5.64.1 Define Documentation

5.64.1.1 #define IPPROTO_SNAP 130

Definition at line 41 of file snap_exec.c.

Referenced by `init_request()`.

5.64.1.2 #define NIPQUAD(addr)

Value:

```
((unsigned char *)&addr)[0], \  
    ((unsigned char *)&addr)[1], \  
    ((unsigned char *)&addr)[2], \  
    ((unsigned char *)&addr)[3]
```

Definition at line 35 of file snap_exec.c.

5.64.2 Function Documentation

5.64.2.1 char* basename (const char *)

5.64.2.2 int compare_longints (const void * a, const void * b)

Definition at line 57 of file snap_exec.c.

5.64.2.3 int init_request (int argc, char ** argv)

Definition at line 64 of file snap_exec.c.

References `d_printf()`, `snaphdr::daddr`, `destaddr`, `file_to_str()`, `snaphdr::flags`, `infd`, `IPPROTO_SNAP`, `buffer_t::lenb`, `out_ttl`, `parse_cmdline()`, `ra_space`, `receiveport`, `snaphdr::saddr`, `sd`, `snaphdr::sport`, `srcaddr`, and `snaphdr::version`.

5.64.2.4 int main (int argc, char ** argv)

Definition at line 242 of file snap_exec.c.

References `buffer_handler`, `init_request()`, `sendpkt()`, `snap_demux_close()`, `snap_demux_handler()`, `snap_demux_init()`, and `SNAP_UDP`.

5.64.2.5 void parse_cmdline (int argc, char ** argv)

Definition at line 130 of file snap_exec.c.

References `basename()`, `destaddr`, `infd`, `infilename`, `out_ttl`, `receiveport`, `srcaddr`, and `usage()`.

5.64.2.6 void sendpkt ()

Definition at line 111 of file snap_exec.c.

References `buffer_t::lenb`, `localaddr`, `buffer_t::s`, and `sd`.

Referenced by `main()`.

5.64.2.7 void usage (int argc, char ** argv)

Definition at line 123 of file snap_exec.c.

References `basename()`.

5.64.3 Variable Documentation**5.64.3.1 struct sockaddr_in destaddr**

Definition at line 46 of file snap_exec.c.

Referenced by `init_request()`, and `parse_cmdline()`.

5.64.3.2 buffer_t inbuf

Definition at line 54 of file snap_exec.c.

5.64.3.3 int infd

Definition at line 53 of file snap_exec.c.

Referenced by `init_request()`, and `parse_cmdline()`.

5.64.3.4 struct sockaddr_in localaddr

Definition at line 48 of file snap_exec.c.

Referenced by `sendpkt()`.

5.64.3.5 unsigned char out_ttl = 32

Definition at line 44 of file snap_exec.c.

Referenced by `init_request()`, and `parse_cmdline()`.

5.64.3.6 short int receiveport = 7777

Definition at line 45 of file snap_exec.c.

Referenced by `init_request()`, and `parse_cmdline()`.

5.64.3.7 int sd

Definition at line 55 of file snap_exec.c.

Referenced by `init_request()`, and `sendpkt()`.

5.64.3.8 struct sockaddr_in srcaddr

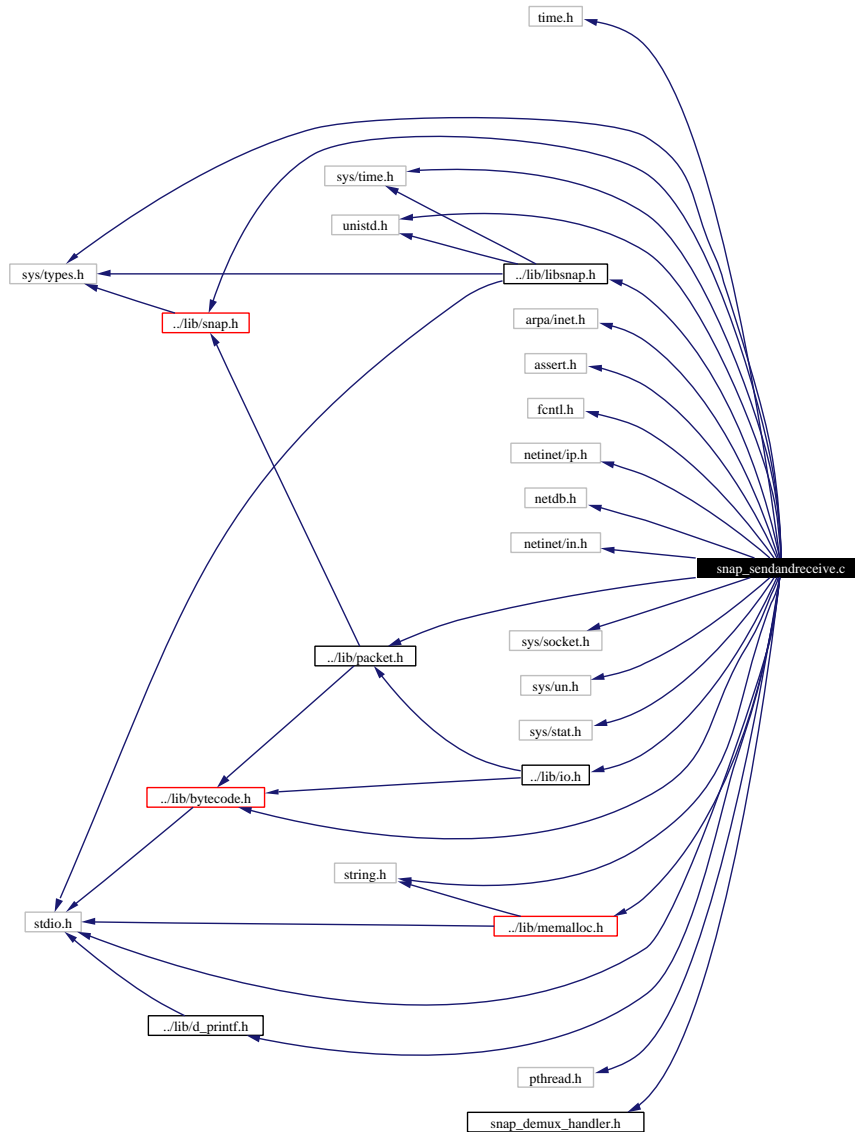
Definition at line 47 of file snap_exec.c.

Referenced by `init_request()`, and `parse_cmdline()`.

5.65 snap-1.1-wjdb/src/snap_sendandreceive.c File Reference

```
#include <time.h>
#include <sys/time.h>
#include "../lib/libsnap.h"
#include <arpa/inet.h>
#include <assert.h>
#include <fcntl.h>
#include <netinet/ip.h>
#include <netdb.h>
#include <netinet/in.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <sys/stat.h>
#include <sys/types.h>
#include "../lib/snap.h"
#include "../lib/bytecode.h"
#include "../lib/packet.h"
#include "../lib/d_printf.h"
#include "../lib/memalloc.h"
#include "../lib/io.h"
#include <pthread.h>
#include "snap_demux_handler.h"
```

Include dependency graph for snap_sendandreceive.c:



Defines

- #define **NIPQUAD**(addr)
- #define **IPPROTO_SNAP** 130
- #define **NO_RUNS** 101

Functions

- char * **basename** (const char *)
- void **parse_cmdline** (int argc, char **argv)
- int **compare_longints** (const void *a, const void *b)

- int **init_request** (int argc, char **argv)
- void **sendpkt** ()
- void **usage** (int argc, char **argv)
- int **main** (int argc, char **argv)

Variables

- unsigned char **out_ttl** = 32
- short int **receiveport** = 7777
- sockaddr_in **destaddr**
- sockaddr_in **srcaddr**
- sockaddr_in **localaddr**
- int **infd**
- **buffer_t** inbuf
- int **sd**

5.65.1 Define Documentation

5.65.1.1 #define IPPROTO_SNAP 130

Definition at line 41 of file snap_sendandreceive.c.

Referenced by `init_request()`.

5.65.1.2 #define NIPQUAD(addr)

Value:

```
((unsigned char *)&addr)[0], \
    ((unsigned char *)&addr)[1], \
    ((unsigned char *)&addr)[2], \
    ((unsigned char *)&addr)[3]
```

Definition at line 35 of file snap_sendandreceive.c.

5.65.1.3 #define NO_RUNS 101

Definition at line 42 of file snap_sendandreceive.c.

Referenced by `main()`.

5.65.2 Function Documentation

5.65.2.1 char* basename (const char *)

5.65.2.2 int compare_longints (const void * a, const void * b)

Definition at line 58 of file snap_sendandreceive.c.

Referenced by `main()`.

5.65.2.3 `int init_request (int argc, char ** argv)`

Definition at line 65 of file `snap_sendandreceive.c`.

References `d_printf()`, `snaphdr::daddr`, `destaddr`, `file_to_str()`, `snaphdr::flags`, `infd`, `IPPROTO_SNAP`, `buffer_t::lenb`, `out_ttl`, `parse_cmdline()`, `ra_space`, `receiveport`, `snaphdr::saddr`, `sd`, `snaphdr::sport`, `srcaddr`, and `snaphdr::version`.

Referenced by `main()`.

5.65.2.4 `int main (int argc, char ** argv)`

Definition at line 243 of file `snap_sendandreceive.c`.

References `buffer_handler`, `compare_longints()`, `init_request()`, `NO_RUNS`, `sendpkt()`, `snap_demux_close()`, `snap_demux_handler()`, `snap_demux_init()`, and `SNAP_UNIX`.

5.65.2.5 `void parse_cmdline (int argc, char ** argv)`

Definition at line 131 of file `snap_sendandreceive.c`.

References `basename()`, `destaddr`, `infd`, `infilename`, `out_ttl`, `receiveport`, `srcaddr`, and `usage()`.

5.65.2.6 `void sendpkt ()`

Definition at line 112 of file `snap_sendandreceive.c`.

References `buffer_t::lenb`, `localaddr`, `buffer_t::s`, and `sd`.

5.65.2.7 `void usage (int argc, char ** argv)`

Definition at line 124 of file `snap_sendandreceive.c`.

References `basename()`.

5.65.3 Variable Documentation

5.65.3.1 `struct sockaddr_in destaddr`

Definition at line 47 of file `snap_sendandreceive.c`.

Referenced by `init_request()`, and `parse_cmdline()`.

5.65.3.2 buffer_t inbuf

Definition at line 55 of file snap_sendandreceive.c.

5.65.3.3 int infd

Definition at line 54 of file snap_sendandreceive.c.

Referenced by `init_request()`, and `parse_cmdline()`.

5.65.3.4 struct sockaddr_in localaddr

Definition at line 49 of file snap_sendandreceive.c.

Referenced by `sendpkt()`.

5.65.3.5 unsigned char out_ttl = 32

Definition at line 45 of file snap_sendandreceive.c.

Referenced by `init_request()`, and `parse_cmdline()`.

5.65.3.6 short int receiveport = 7777

Definition at line 46 of file snap_sendandreceive.c.

Referenced by `init_request()`, and `parse_cmdline()`.

5.65.3.7 int sd

Definition at line 56 of file snap_sendandreceive.c.

Referenced by `init_request()`, and `sendpkt()`.

5.65.3.8 struct sockaddr_in srcaddr

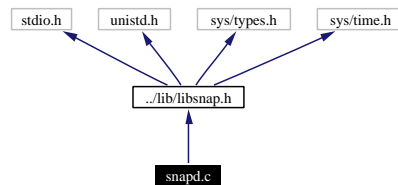
Definition at line 48 of file snap_sendandreceive.c.

Referenced by `init_request()`, and `parse_cmdline()`.

5.66 snap-1.1-wjdb/src/snapd.c File Reference

```
#include "../lib/libsnap.h"
```

Include dependency graph for snapd.c:



Functions

- `int main (int argc, char **argv)`

5.66.1 Function Documentation

5.66.1.1 `int main (int argc, char ** argv)`

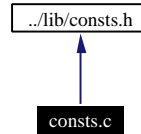
Definition at line 11 of file snapd.c.

References `init_snap()`, and `snap_receive()`.

5.67 snap-1.1-wjdb/utils/consts.c File Reference

```
#include "../lib/consts.h"
```

Include dependency graph for consts.c:



Variables

- `int heap_sizeb = DEFAULT_HEAP_SIZEB`
- `int stack_sizeb = DEFAULT_STACK_SIZEB`
- `int code_sizeb = DEFAULT_CODE_SIZEB`

5.67.1 Variable Documentation

5.67.1.1 `int code_sizeb = DEFAULT_CODE_SIZEB`

Definition at line 12 of file `consts.c`.

5.67.1.2 `int heap_sizeb = DEFAULT_HEAP_SIZEB`

Definition at line 10 of file `consts.c`.

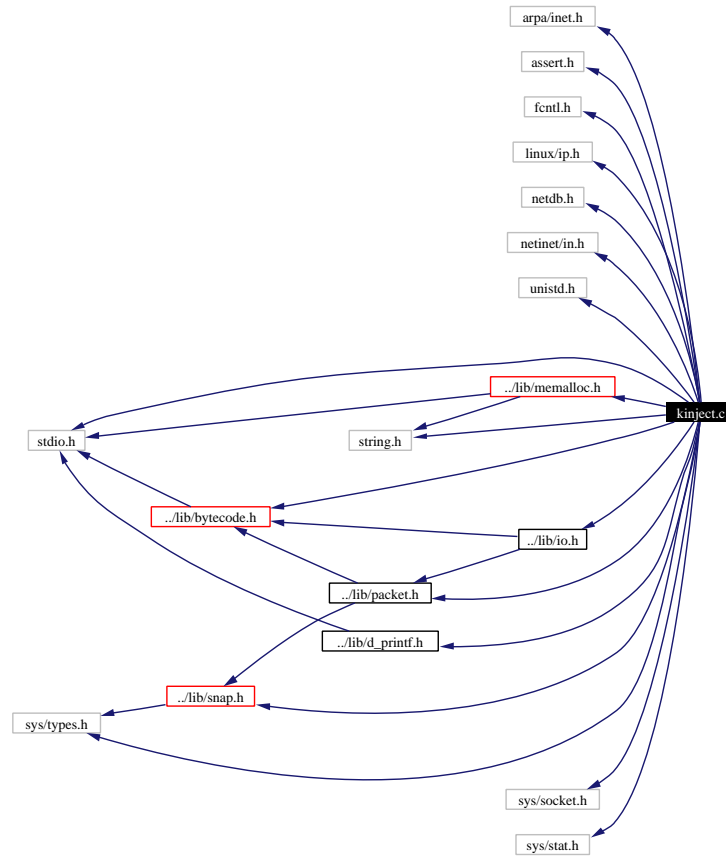
5.67.1.3 `int stack_sizeb = DEFAULT_STACK_SIZEB`

Definition at line 11 of file `consts.c`.

5.68 snap-1.1-wjdb/utils/kinject.c File Reference

```
#include <arpa/inet.h>
#include <assert.h>
#include <fcntl.h>
#include <linux/ip.h>
#include <netdb.h>
#include <netinet/in.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/stat.h>
#include <sys/types.h>
#include "../lib/snap.h"
#include "../lib/bytecode.h"
#include "../lib/packet.h"
#include "../lib/d_printf.h"
#include "../lib/memalloc.h"
#include "../lib/io.h"
```

Include dependency graph for kinject.c:



Defines

- `#define NIPQUAD(addr)`
- `#define IPPROTO_SNAP 130`

Functions

- `char * basename (const char *)`
- `void parse_cmdline (int argc, char **argv)`
- `int main (int argc, char **argv)`
- `void usage (int argc, char **argv)`

Variables

- unsigned char `out_ttl = 16`
- short int `udpport = 7777`
- sockaddr_in `destaddr`
- int `infd`

5.68.1 Define Documentation

5.68.1.1 `#define IPPROTO_SNAP 130`

Definition at line 28 of file kinject.c.

Referenced by main().

5.68.1.2 `#define NIPQUAD(addr)`

Value:

```
((unsigned char *)&addr)[0], \  
    ((unsigned char *)&addr)[1], \  
    ((unsigned char *)&addr)[2], \  
    ((unsigned char *)&addr)[3]
```

Definition at line 22 of file kinject.c.

5.68.2 Function Documentation

5.68.2.1 `char* basename (const char *)`

5.68.2.2 `int main (int argc, char ** argv)`

Definition at line 43 of file kinject.c.

References `snaphdr::daddr`, `destaddr`, `file_to_str()`, `snaphdr::flags`, `infd`, `IPPROTO_SNAP`, `buffer_t::lenb`, `NIPQUAD`, `out_ttl`, `parse_cmdline()`, `ra_space`, `snaphdr::saddr`, `snaphdr::sport`, `udpport`, and `snaphdr::version`.

5.68.2.3 `void parse_cmdline (int argc, char ** argv)`

Definition at line 115 of file kinject.c.

References `basename()`, `destaddr`, `infd`, `infilename`, `out_ttl`, `udpport`, and `usage()`.

5.68.2.4 `void usage (int argc, char ** argv)`

Definition at line 101 of file kinject.c.

References `basename()`.

5.68.3 Variable Documentation

5.68.3.1 `struct sockaddr_in destaddr`

Definition at line 33 of file kinject.c.

Referenced by `main()`, and `parse_cmdline()`.

5.68.3.2 int infd

Definition at line 41 of file `kinject.c`.

Referenced by `main()`, and `parse_cmdline()`.

5.68.3.3 unsigned char out_ttl = 16

Definition at line 31 of file `kinject.c`.

Referenced by `main()`, and `parse_cmdline()`.

5.68.3.4 short int udpport = 7777

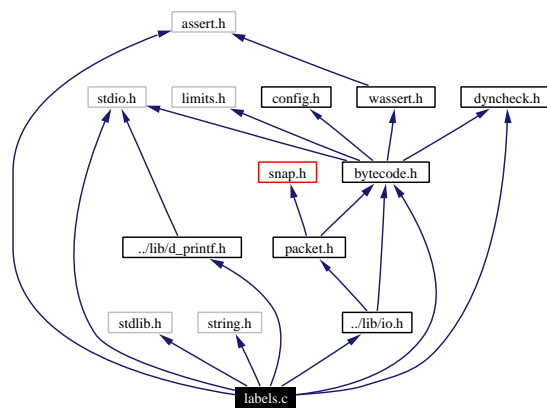
Definition at line 32 of file `kinject.c`.

Referenced by `main()`, and `parse_cmdline()`.

5.69 snap-1.1-wjdb/utils/labels.c File Reference

```
#include <assert.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../lib/io.h"
#include "../lib/bytecode.h"
#include "../lib/d_printf.h"
#include "../lib/dyncheck.h"
```

Include dependency graph for labels.c:



Data Structures

- struct `label_mapping_t`

Defines

- `#define MAX_NUM_LABELS 100`

Functions

- void `register_label_def` (char *labname, void *where)
- void `register_label_use` (char *labname, void *where)
- void `patch_jumps` (packet_t *p)

5.69.1 Define Documentation

5.69.1.1 `#define MAX_NUM_LABELS 100`

Definition at line 25 of file labels.c.

5.69.2 Function Documentation

5.69.2.1 `void patch_jumps (packet_t * p)`

Definition at line 131 of file labels.c.

References BEZ, BNE, `packet_t::code_min`, EQADR, EQEXC, EQINT, EQSTR, EQTUP, GET_OP, `instr_t`, JI, NQADR, NQEXC, NQINT, NQSTR, NQTUP, PADDR, `packet_t::pc`, PEXC, PINT, PSTR, PTUP, PUSH, `packet_t::sp`, `packet_t::stack_min`, and `value_t`.

Referenced by `main()`.

5.69.2.2 `void register_label_def (char * labname, void * where)`

Definition at line 72 of file labels.c.

Referenced by `yyparse()`.

5.69.2.3 `void register_label_use (char * labname, void * where)`

Definition at line 78 of file labels.c.

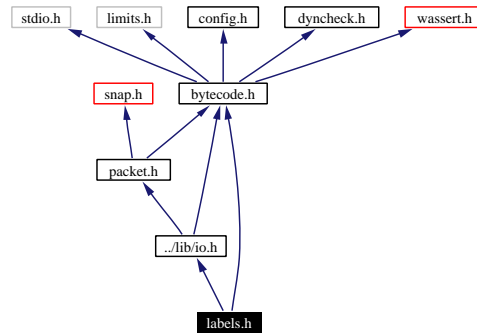
Referenced by `yyparse()`.

5.70 snap-1.1-wjdb/utils/labels.h File Reference

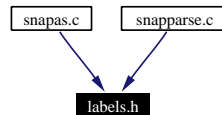
```
#include "../lib/io.h"
```

```
#include "../lib/bytecode.h"
```

Include dependency graph for labels.h:



This graph shows which files directly or indirectly include this file:



Functions

- void **register_label_def** (char *labname, void *where)
- void **register_label_use** (char *labname, void *where)
- void **patch_jumps** (packet_t *p)

5.70.1 Function Documentation

5.70.1.1 void patch_jumps (packet_t * p)

Definition at line 131 of file labels.c.

References BEZ, BNE, packet_t::code_min, EQADR, EQEXC, EQINT, EQSTR, EQTUP, GET_OP, instr_t, JI, NQADR, NQEXC, NQINT, NQSTR, NQTUP, PADDR, packet_t::pc, PEXC, PINT, PTR, PTUP, PUSH, packet_t::sp, packet_t::stack_min, and value_t.

Referenced by main().

5.70.1.2 void register_label_def (char * *labname*, void * *where*)

Definition at line 72 of file labels.c.

Referenced by yyparse().

5.70.1.3 void register_label_use (char * *labname*, void * *where*)

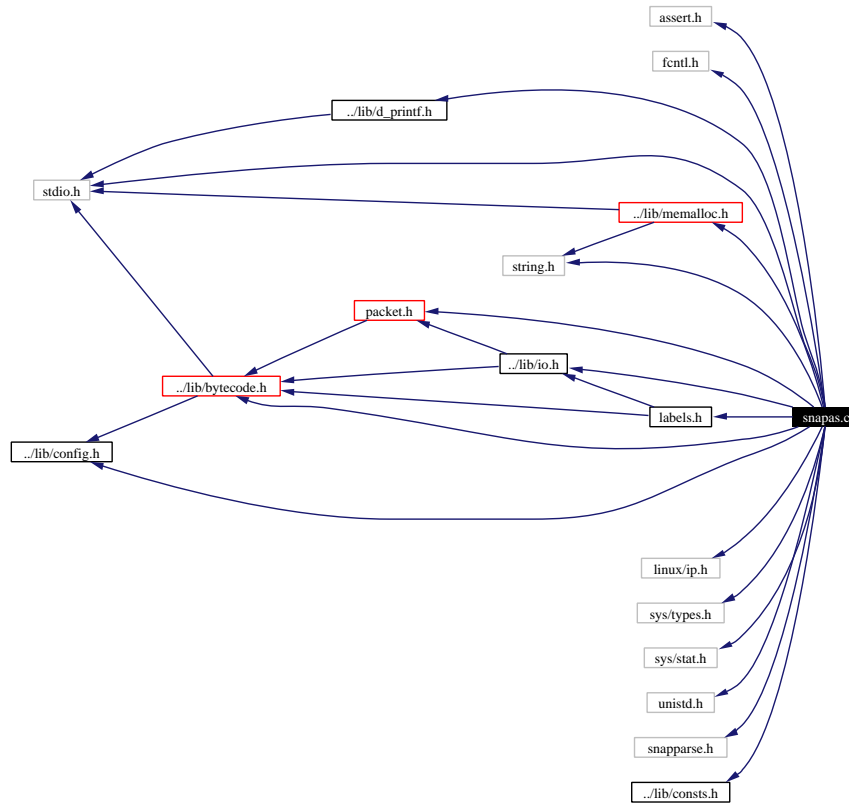
Definition at line 78 of file labels.c.

Referenced by yyparse().

5.71 snap-1.1-wjdb/utils/snapas.c File Reference

```
#include <assert.h>
#include <fcntl.h>
#include <stdio.h>
#include <string.h>
#include <linux/ip.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include "../lib/config.h"
#include "../lib/d_printf.h"
#include "../lib/bytecode.h"
#include "../lib/io.h"
#include "../lib/memalloc.h"
#include "../lib/packet.h"
#include "snapparse.h"
#include "labels.h"
#include "../lib/consts.h"
```

Include dependency graph for snapas.c:



Functions

- char * **basename** (const char *)
- void **parse_cmdline** (int argc, char **argv)
- int **yyparse** (void)
- int **main** (int argc, char **argv)
- void **usage** (int argc, char **argv)

Variables

- int **yydebug**
- FILE * **yyin** = (FILE *) 0
- int **outfd**
- char * **outfile** = NULL
- char * **infile**
- **packet_t** * p
- **instr_t** * cbuf
- **value_t** * sbuf
- void * **hbuf**
- char * **pbuf**
- int **noop**

5.71.1 Function Documentation

5.71.1.1 `char* basename (const char *)`

5.71.1.2 `int main (int argc, char ** argv)`

Definition at line 55 of file snapas.c.

References `cbuf`, `packet_t::code_max`, `packet_t::code_min`, `d_printf()`, `packet_t::h_alloc_heap_max`, `packet_t::h_alloc_ptr`, `hbuf`, `packet_t::hdr`, `packet_t::heap_max`, `packet_t::heap_min`, `instr_t`, `packet_t::is_contiguous`, `buffer_t::lenb`, `marshal_packet()`, `memalloc`, `noop`, `outfd`, `parse_cmdline()`, `patch_jumps()`, `packet_t::pc`, `buffer_t::s`, `sbuf`, `packet_t::sp`, `packet_t::stack_max`, `packet_t::stack_min`, `value_t`, `yydebug`, and `yyparse()`.

5.71.1.3 `void parse_cmdline (int argc, char ** argv)`

Definition at line 119 of file snapas.c.

References `basename()`, `infilename`, `memalloc`, `outfd`, `outfilename`, and `usage()`.

5.71.1.4 `void usage (int argc, char ** argv)`

Definition at line 113 of file snapas.c.

References `basename()`.

5.71.1.5 `int yyparse (void)`

Referenced by `main()`.

5.71.2 Variable Documentation

5.71.2.1 `instr_t* cbuf`

Definition at line 44 of file snapas.c.

Referenced by `main()`.

5.71.2.2 `void* hbuf`

Definition at line 46 of file snapas.c.

Referenced by `main()`.

5.71.2.3 char* infilename

Definition at line 41 of file snapas.c.

Referenced by parse_cmdline().

5.71.2.4 int noop

Definition at line 49 of file snapas.c.

Referenced by main(), and yyparse().

5.71.2.5 int outfd

Definition at line 39 of file snapas.c.

Referenced by main(), and parse_cmdline().

5.71.2.6 char* outfilename = NULL

Definition at line 40 of file snapas.c.

Referenced by parse_cmdline().

5.71.2.7 packet_t* p

Definition at line 43 of file snapas.c.

5.71.2.8 char* pbuf

Definition at line 47 of file snapas.c.

5.71.2.9 value_t* sbuf

Definition at line 45 of file snapas.c.

Referenced by main().

5.71.2.10 int yydebug

Definition at line 36 of file snapas.c.

Referenced by main().

5.71.2.11 FILE * yyin = (FILE *) 0

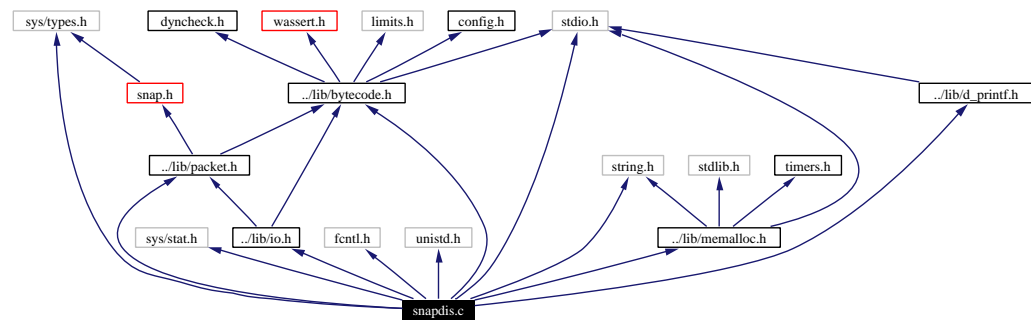
Definition at line 265 of file snaplex.c.

Referenced by snap_yy_input().

5.72 snap-1.1-wjdb/utils/snapdis.c File Reference

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <string.h>
#include <fcntl.h>
#include <unistd.h>
#include "../lib/bytecode.h"
#include "../lib/packet.h"
#include "../lib/io.h"
#include "../lib/memalloc.h"
#include "../lib/d_printf.h"
```

Include dependency graph for snapdis.c:



Functions

- char * **basename** (const char *)
- void **parse_cmdline** (int argc, char **argv)
- int **main** (int argc, char **argv)
- void **usage** (int argc, char **argv)

Variables

- int **infd**
- FILE * **outfile**

5.72.1 Function Documentation

5.72.1.1 `char* basename (const char *)`

Referenced by `parse_cmdline()`, `parse_cmdline_snap()`, and `usage()`.

5.72.1.2 `int main (int argc, char ** argv)`

Definition at line 28 of file `snapdis.c`.

References `d_printf()`, `file_to_str()`, `fprintf_packet()`, `infd`, `buffer_t::lenb`, `outfile`, `parse_cmdline()`, `buffer_t::s`, and `unmarshal_packet()`.

5.72.1.3 `void parse_cmdline (int argc, char ** argv)`

Definition at line 55 of file `snapdis.c`.

References `basename()`, `infd`, `infilename`, `outfile`, `outfilename`, and `usage()`.

Referenced by `init_request()`, and `main()`.

5.72.1.4 `void usage (int argc, char ** argv)`

Definition at line 49 of file `snapdis.c`.

References `basename()`.

Referenced by `parse_cmdline()`, and `parse_cmdline_snap()`.

5.72.2 Variable Documentation

5.72.2.1 `int infd`

Definition at line 21 of file `snapdis.c`.

Referenced by `main()`, and `parse_cmdline()`.

5.72.2.2 `FILE* outfile`

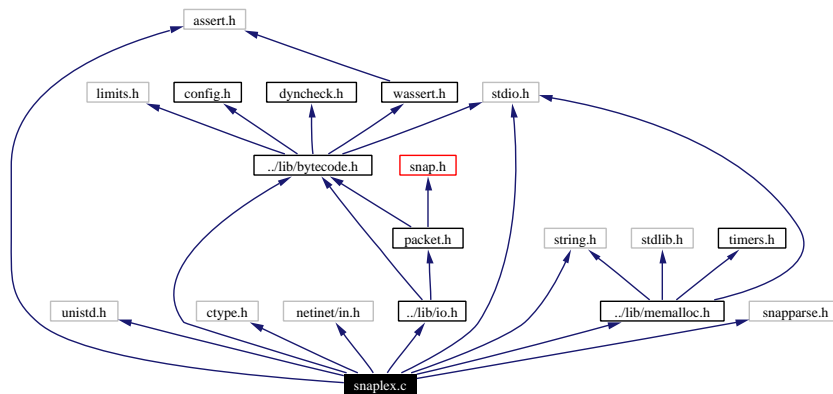
Definition at line 22 of file `snapdis.c`.

Referenced by `fprintf_packet()`, `main()`, and `parse_cmdline()`.

5.73 snap-1.1-wjdb/utils/snaplex.c File Reference

```
#include <stdio.h>
#include <unistd.h>
#include <assert.h>
#include <ctype.h>
#include <netinet/in.h>
#include <string.h>
#include "../lib/bytecode.h"
#include "../lib/io.h"
#include "../lib/memalloc.h"
#include "snapparse.h"
```

Include dependency graph for snaplex.c:



Data Structures

- struct yy_buffer_state

Defines

- #define FLEX_SCANNER
- #define YY_FLEX_MAJOR_VERSION 2
- #define YY_FLEX_MINOR_VERSION 5
- #define yyconst
- #define YY_PROTO(proto) ()
- #define YY_NULL 0

- #define **YY_SC_TO_UI**(c) ((unsigned int) (unsigned char) c)
- #define **BEGIN** yy_start = 1 + 2 *
- #define **YY_START** ((yy_start - 1) / 2)
- #define **YYSTATE** YY_START
- #define **YY_STATE_EOF**(state) (YY_END_OF_BUFFER + state + 1)
- #define **YY_NEW_FILE** yyrestart(yyin)
- #define **YY_END_OF_BUFFER_CHAR** 0
- #define **YY_BUF_SIZE** 16384
- #define **EOB_ACT_CONTINUE_SCAN** 0
- #define **EOB_ACT_END_OF_FILE** 1
- #define **EOB_ACT_LAST_MATCH** 2
- #define **yyless**(n)
- #define **unput**(c) yyunput(c, yytext_ptr)
- #define **YY_BUFFER_NEW** 0
- #define **YY_BUFFER_NORMAL** 1
- #define **YY_BUFFER_EOF_PENDING** 2
- #define **YY_CURRENT_BUFFER** yy_current_buffer
- #define **YY_FLUSH_BUFFER** yy_flush_buffer(yy_current_buffer)
- #define **yy_new_buffer** yy_create_buffer
- #define **yy_set_interactive**(is_interactive)
- #define **yy_set_bol**(at_bol)
- #define **YY_AT_BOL**() (yy_current_buffer → yy_at_bol)
- #define **yytext_ptr** yytext
- #define **YY_DO_BEFORE_ACTION**
- #define **YY_NUM_RULES** 102
- #define **YY_END_OF_BUFFER** 103
- #define **REJECT** reject_used_but_not_detected
- #define **yymore**() yymore_used_but_not_detected
- #define **YY_MORE_ADJ** 0
- #define **YY_RESTORE_YY_MORE_OFFSET**
- #define **INITIAL** 0
- #define **YY_INPUT**(buf, result, max_size) { (result) = snap_yy_input((buf),(max_size)); }
- #define **YY_NO_PUSH_STATE** 1
- #define **YY_NO_POP_STATE** 1
- #define **YY_NO_TOP_STATE** 1
- #define **YY_READ_BUF_SIZE** 8192
- #define **ECHO** (void) fwrite(yytext, yyleng, 1, yyout)
- #define **yyterminate**() return YY_NULL
- #define **YY_START_STACK_INCR** 25
- #define **YY_FATAL_ERROR**(msg) yy_fatal_error(msg)
- #define **YY_DECL** int yylex YY_PROTO((void))
- #define **YY_BREAK** break;
- #define **YY_RULE_SETUP** YY_USER_ACTION
- #define **YY_EXIT_FAILURE** 2
- #define **yyless**(n)

Typedefs

- typedef `yy_buffer_state` * `YY_BUFFER_STATE`
- typedef unsigned int `yy_size_t`
- typedef unsigned char `YY_CHAR`
- typedef int `yy_state_type`

Functions

- void `yyrestart` `YY_PROTO` ((FILE *input_file))
- void `yy_switch_to_buffer` `YY_PROTO` ((`YY_BUFFER_STATE` new_buffer))
- void `yy_load_buffer_state` `YY_PROTO` ((void))
- `YY_BUFFER_STATE` `yy_create_buffer` `YY_PROTO` ((FILE *file, int size))
- void `yy_delete_buffer` `YY_PROTO` ((`YY_BUFFER_STATE` b))
- void `yy_init_buffer` `YY_PROTO` ((`YY_BUFFER_STATE` b, FILE *file))
- `YY_BUFFER_STATE` `yy_scan_buffer` `YY_PROTO` ((char *base, `yy_size_t` size))
- `YY_BUFFER_STATE` `yy_scan_string` `YY_PROTO` ((yyconst char *yy_str))
- `YY_BUFFER_STATE` `yy_scan_bytes` `YY_PROTO` ((yyconst char *bytes, int len))
- void `conv_string` (char *s, `buffer_t` *buf)
- int `snap_yy_input` (char *buf, int max_size)

Variables

- int `yleng`
- FILE * `yyin` = (FILE *) 0
- FILE * `yyout` = (FILE *) 0
- char * `yytext`
- int `value_int`
- `uint32` `value_addr`
- `buffer_t` `value_str`
- int `value_exc`
- `float32` `value_float`
- int `read_from_file` = 1
- char * `lexbuf` = NULL
- int `lexbuf_len` = 0
- int `lexbuf_pos` = 0
- register char * `yy_bp`
- int `size`
- FILE * `file`
- int `len`

5.73.1 Define Documentation

5.73.1.1 `#define BEGIN yy_start = 1 + 2 *`

Definition at line 79 of file snaplex.c.

5.73.1.2 `#define ECHO (void) fwrite(yytext, yyleng, 1, yyout)`

Definition at line 918 of file snaplex.c.

5.73.1.3 `#define EOB_ACT_CONTINUE_SCAN 0`

Definition at line 104 of file snaplex.c.

5.73.1.4 `#define EOB_ACT_END_OF_FILE 1`

Definition at line 105 of file snaplex.c.

5.73.1.5 `#define EOB_ACT_LAST_MATCH 2`

Definition at line 106 of file snaplex.c.

5.73.1.6 `#define FLEX_SCANNER`

Definition at line 7 of file snaplex.c.

5.73.1.7 `#define INITIAL 0`

Definition at line 773 of file snaplex.c.

5.73.1.8 `#define REJECT reject_used_but_not_detected`

Definition at line 767 of file snaplex.c.

5.73.1.9 `#define unput(c) yyunput(c, yytext_ptr)`

Definition at line 135 of file snaplex.c.

5.73.1.10 `#define YY_AT_BOL() (yy_current_buffer → yy_at_bol)`

Definition at line 262 of file snaplex.c.

5.73.1.11 `#define YY_BREAK break;`

Definition at line 977 of file snaplex.c.

5.73.1.12 `#define YY_BUF_SIZE 16384`

Definition at line 97 of file snaplex.c.

5.73.1.13 `#define YY_BUFFER_EOF_PENDING 2`

Definition at line 198 of file snaplex.c.

5.73.1.14 `#define YY_BUFFER_NEW 0`

Definition at line 186 of file snaplex.c.

5.73.1.15 `#define YY_BUFFER_NORMAL 1`

Definition at line 187 of file snaplex.c.

5.73.1.16 `#define YY_CURRENT_BUFFER yy_current_buffer`

Definition at line 207 of file snaplex.c.

5.73.1.17 `#define YY_DECL int yylex YY_PROTO((void))`

Definition at line 965 of file snaplex.c.

5.73.1.18 `#define YY_DO_BEFORE_ACTION`

Value:

```
yytext_ptr = yy_bp; \  
  yyleng = (int) (yy_cp - yy_bp); \  
  yy_hold_char = *yy_cp; \  
  *yy_cp = '\0'; \  
  yy_c_buf_p = yy_cp;
```

Definition at line 278 of file snaplex.c.

5.73.1.19 `#define YY_END_OF_BUFFER 103`

Definition at line 286 of file snaplex.c.

5.73.1.20 `#define YY_END_OF_BUFFER_CHAR 0`

Definition at line 94 of file snaplex.c.

5.73.1.21 `#define YY_EXIT_FAILURE 2`

5.73.1.22 `#define YY_FATAL_ERROR(msg) yy_fatal_error(msg)`

Definition at line 958 of file snaplex.c.

5.73.1.23 `#define YY_FLEX_MAJOR_VERSION 2`

Definition at line 8 of file snaplex.c.

5.73.1.24 `#define YY_FLEX_MINOR_VERSION 5`

Definition at line 9 of file snaplex.c.

5.73.1.25 `#define YY_FLUSH_BUFFER yy_flush_buffer(
yy_current_buffer)`

Definition at line 236 of file snaplex.c.

5.73.1.26 `#define YY_INPUT(buf, result, max_size) { (result) =
snap_yy_input((buf),(max_size)); }`

Definition at line 835 of file snaplex.c.

5.73.1.27 `#define YY_MORE_ADJ 0`

Definition at line 769 of file snaplex.c.

5.73.1.28 `#define yy_new_buffer yy_create_buffer`

Definition at line 246 of file snaplex.c.

5.73.1.29 `#define YY_NEW_FILE yyrestart(yyin)`

Definition at line 92 of file snaplex.c.

5.73.1.30 `#define YY_NO_POP_STATE 1`

Definition at line 888 of file snaplex.c.

5.73.1.31 #define YY_NO_PUSH_STATE 1

Definition at line 887 of file snaplex.c.

5.73.1.32 #define YY_NO_TOP_STATE 1

Definition at line 889 of file snaplex.c.

5.73.1.33 #define YY_NULL 0

Definition at line 66 of file snaplex.c.

5.73.1.34 #define YY_NUM_RULES 102

Definition at line 285 of file snaplex.c.

5.73.1.35 #define YY_PROTO(proto) ()

Definition at line 62 of file snaplex.c.

5.73.1.36 #define YY_READ_BUF_SIZE 8192

Definition at line 909 of file snaplex.c.

5.73.1.37 #define YY_RESTORE_YY_MORE_OFFSET

Definition at line 770 of file snaplex.c.

5.73.1.38 #define YY_RULE_SETUP YY_USER_ACTION

Definition at line 980 of file snaplex.c.

5.73.1.39 #define YY_SC_TO_UI(c) ((unsigned int) (unsigned char) c)

Definition at line 73 of file snaplex.c.

5.73.1.40 #define yy_set_bol(at_bol)

Value:

```
{ \
  if ( ! yy_current_buffer ) \
    yy_current_buffer = yy_create_buffer( yyin, YY_BUF_SIZE ); \
```

```
yy_current_buffer->yy_at_bol = at_bol; \
}
```

Definition at line 255 of file snaplex.c.

5.73.1.41 #define yy_set_interactive(is_interactive)

Value:

```
{ \
  if ( ! yy_current_buffer ) \
    yy_current_buffer = yy_create_buffer( yyin, YY_BUF_SIZE ); \
  yy_current_buffer->yy_is_interactive = is_interactive; \
}
```

Definition at line 248 of file snaplex.c.

5.73.1.42 #define YY_START ((yy_start - 1) / 2)

Definition at line 85 of file snaplex.c.

5.73.1.43 #define YY_START_STACK_INCR 25

Definition at line 953 of file snaplex.c.

5.73.1.44 #define YY_STATE_EOF(state) (YY_END_OF_BUFFER + state + 1)

Definition at line 89 of file snaplex.c.

5.73.1.45 #define yyconst

Definition at line 55 of file snaplex.c.

5.73.1.46 #define yyless(n)

Value:

```
do \
  { \
    \
    yytext[yyless] = yy_hold_char; \
    yy_c_buf_p = yytext + n; \
    yy_hold_char = *yy_c_buf_p; \
    *yy_c_buf_p = '\0'; \
    yyless = n; \
  } \
while ( 0 )
```

Definition at line 124 of file snaplex.c.

5.73.1.47 `#define yyles(n)`

Value:

```
do \
    { \
        \
        *yy_cp = yy_hold_char; \
        YY_RESTORE_YY_MORE_OFFSET \
        yy_c_buf_p = yy_cp = yy_bp + n - YY_MORE_ADJ; \
        YY_DO_BEFORE_ACTION; \
    } \
    while ( 0 )
```

Definition at line 124 of file snaplex.c.

5.73.1.48 `#define yymore() yymore_used_but_not_detected`

Definition at line 768 of file snaplex.c.

5.73.1.49 `#define YYSTATE YY_START`

Definition at line 86 of file snaplex.c.

5.73.1.50 `#define yyterminate() return YY_NULL`

Definition at line 948 of file snaplex.c.

5.73.1.51 `#define yytext_ptr yytext`

Definition at line 268 of file snaplex.c.

5.73.2 Typedef Documentation

5.73.2.1 `typedef struct yy_buffer_state* YY_BUFFER_STATE`

Definition at line 99 of file snaplex.c.

5.73.2.2 `typedef unsigned char YY_CHAR`

Definition at line 264 of file snaplex.c.

5.73.2.3 typedef unsigned int yy_size_t

Definition at line 141 of file snaplex.c.

5.73.2.4 typedef int yy_state_type

Definition at line 266 of file snaplex.c.

5.73.3 Function Documentation

5.73.3.1 void conv_string (char * s, buffer_t * buf)

5.73.3.2 int snap_yy_input (char * buf, int max_size)

Definition at line 805 of file snaplex.c.

References lexbuf, lexbuf_len, lexbuf_pos, and yyin.

5.73.3.3 YY_BUFFER_STATE yy_scan_bytes YY_PROTO ((yyconst char *bytes, int len))

5.73.3.4 YY_BUFFER_STATE yy_scan_string YY_PROTO ((yyconst char *yy_str))

5.73.3.5 YY_BUFFER_STATE yy_scan_buffer YY_PROTO ((char *base, yy_size_t size))

5.73.3.6 void yy_init_buffer YY_PROTO ((YY_BUFFER_STATE b, FILE *file))

5.73.3.7 void yy_flush_buffer YY_PROTO ((YY_BUFFER_STATE b))

5.73.3.8 YY_BUFFER_STATE yy_create_buffer YY_PROTO ((FILE *file, int size))

5.73.3.9 int input YY_PROTO ((void))

5.73.3.10 void yy_switch_to_buffer YY_PROTO ((YY_BUFFER_STATE new_buffer))

5.73.3.11 void yyrestart YY_PROTO ((FILE *input_file))

5.73.4 Variable Documentation

5.73.4.1 FILE* file

Definition at line 2236 of file snaplex.c.

Referenced by `newho()`, and `newtup()`.

5.73.4.2 int len

Definition at line 2349 of file `snaplex.c`.

Referenced by `newtup()`, and `snap_rcv_pkt()`.

5.73.4.3 char* lexbuf = NULL

Definition at line 799 of file `snaplex.c`.

Referenced by `snap_yy_input()`.

5.73.4.4 int lexbuf_len = 0

Definition at line 800 of file `snaplex.c`.

Referenced by `snap_yy_input()`.

5.73.4.5 int lexbuf_pos = 0

Definition at line 801 of file `snaplex.c`.

Referenced by `snap_yy_input()`.

5.73.4.6 int read_from_file = 1

Definition at line 797 of file `snaplex.c`.

5.73.4.7 yy_size_t size

Definition at line 2517 of file `snaplex.c`.

5.73.4.8 uint32 value_addr

Definition at line 792 of file `snaplex.c`.

Referenced by `yyvsparse()`.

5.73.4.9 int value_exc

Definition at line 794 of file `snaplex.c`.

Referenced by `yyvsparse()`.

5.73.4.10 float32 value_float

Definition at line 795 of file snaplex.c.

Referenced by yyparse().

5.73.4.11 int value_int

Definition at line 791 of file snaplex.c.

Referenced by yyparse().

5.73.4.12 buffer_t value_str

Definition at line 793 of file snaplex.c.

5.73.4.13 register char* yy_bp

Definition at line 2009 of file snaplex.c.

5.73.4.14 FILE* yyin = (FILE *) 0

Definition at line 265 of file snaplex.c.

5.73.4.15 int yyleng

Definition at line 216 of file snaplex.c.

5.73.4.16 FILE * yyout = (FILE *) 0

Definition at line 265 of file snaplex.c.

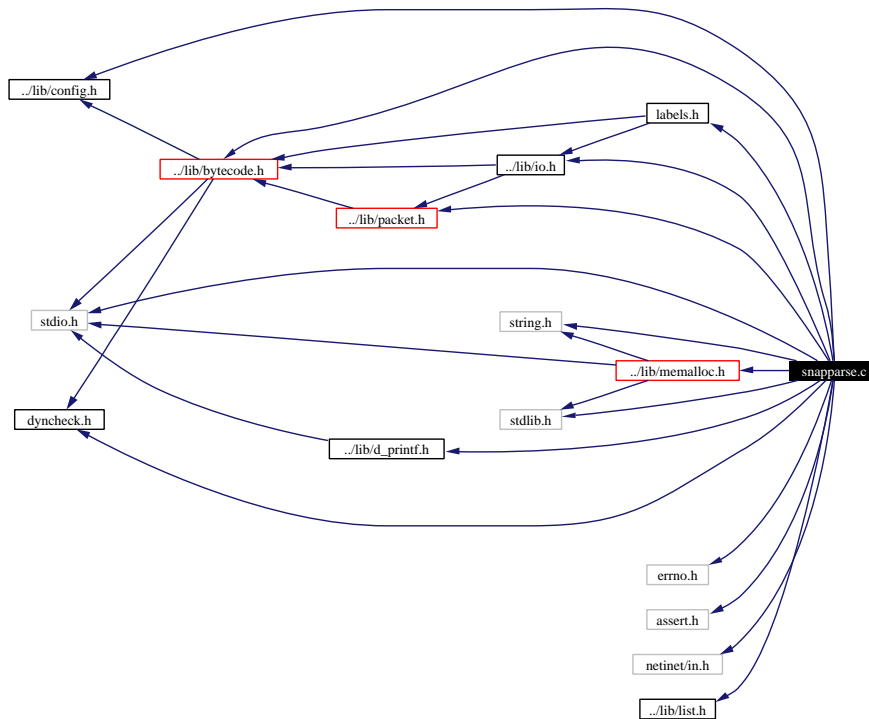
5.73.4.17 char * yytext

Definition at line 771 of file snaplex.c.

5.74 snap-1.1-wjdb/utils/snapparse.c File Reference

```
#include "../lib/config.h"
#include <errno.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#include <netinet/in.h>
#include "../lib/bytecode.h"
#include "../lib/dyncheck.h"
#include "../lib/packet.h"
#include "../lib/io.h"
#include "../lib/memalloc.h"
#include "labels.h"
#include "../lib/d_printf.h"
#include "../lib/list.h"
```

Include dependency graph for snapparse.c:



Data Structures

- union `yvalloc`
- union `yystype`

Defines

- `#define YYBISON 1`
- `#define T_INTV 257`
- `#define T_ADDRV 258`
- `#define T_STRV 259`
- `#define T_EXCV 260`
- `#define T_FLOATV 261`
- `#define T_MAIN 262`
- `#define T_EXIT 263`
- `#define T_PUSH 264`
- `#define T_POP 265`
- `#define T_POPI 266`
- `#define T_PULL 267`
- `#define T_EQ 268`
- `#define T_EQI 269`
- `#define T_PAJ 270`

- #define **T_TPAJ** 271
- #define **T_BEZ** 272
- #define **T_BNE** 273
- #define **T_STORE** 274
- #define **T_JI** 275
- #define **T_NEQ** 276
- #define **T_NEQI** 277
- #define **T_ADD** 278
- #define **T_ADDI** 279
- #define **T_SUB** 280
- #define **T_SUBI** 281
- #define **T_MULT** 282
- #define **T_MULTI** 283
- #define **T_DIV** 284
- #define **T_DIVI** 285
- #define **T_MOD** 286
- #define **T_MODI** 287
- #define **T_NEG** 288
- #define **T_NOT** 289
- #define **T_LNOT** 290
- #define **T_AND** 291
- #define **T_ANDI** 292
- #define **T_OR** 293
- #define **T_ORI** 294
- #define **T_LSHL** 295
- #define **T_LSHLI** 296
- #define **T_RSHL** 297
- #define **T_RSHLI** 298
- #define **T_RSHA** 299
- #define **T_RSHAI** 300
- #define **T_SNET** 301
- #define **T_SNETI** 302
- #define **T_BCAST** 303
- #define **T_BCASTI** 304
- #define **T_ISX** 305
- #define **T_GETRB** 306
- #define **T_GETSRC** 307
- #define **T_GETDST** 308
- #define **T_GETSPT** 309
- #define **T_HERE** 310
- #define **T_ISHERE** 311
- #define **T_ROUTE** 312
- #define **T_RTDEV** 313
- #define **T_SEND** 314
- #define **T_HOP** 315
- #define **T_FORW** 316

- #define **T_FORWTO** 317
- #define **T_DEMUX** 318
- #define **T_DEMUXI** 319
- #define **T_PRINT** 320
- #define **T_GETLD** 321
- #define **T_SETXH** 322
- #define **T_RAISEX** 323
- #define **T_PLUS** 324
- #define **T_MINUS** 325
- #define **T_LABEL** 326
- #define **T_LABELV** 327
- #define **T_PC** 328
- #define **T_MKTUP** 329
- #define **T_LEN** 330
- #define **T_NTH** 331
- #define **T_ISTUP** 332
- #define **T_LPAREN** 333
- #define **T_RPAREN** 334
- #define **T_COMMA** 335
- #define **T_SVCV** 336
- #define **T_CALLS** 337
- #define **T_GT** 338
- #define **T_GEQ** 339
- #define **T_LT** 340
- #define **T_LEQ** 341
- #define **T_GTI** 342
- #define **T_GEQI** 343
- #define **T_LTI** 344
- #define **T_LEQI** 345
- #define **T_DATA** 346
- #define **T_DFORW** 347
- #define **T_DFORWTO** 348
- #define **T_DSEND** 349
- #define **T_STACKEMPTY** 350
- #define **T_STACKCOUNT** 351
- #define **T_PULLSTACK** 352
- #define **CHECK_CODE_OVERFLOW(p)**
- #define **YYSTYPE** `ystype`
- #define **YYSTYPE_IS_TRIVIAL** 1
- #define **YYDEBUG** 0
- #define **YYFINAL** 117
- #define **YYFLAG** -32768
- #define **YYNTBASE** 99
- #define **YYTRANSLATE(x)** `((unsigned)(x) <= 352 ? yytranslate[x] : 109)`
- #define **YYLAST** 167

- #define **YYSTACK_ALLOC** malloc
- #define **YYSTACK_FREE** free
- #define **YYSTACK_GAP_MAX** (sizeof (union **yyval**) - 1)
- #define **YYSTACK_BYTES**(N)
- #define **YYCOPY**(To, From, Count)
- #define **YYSTACK_RELOCATE**(Stack)
- #define **YYSIZE_T** unsigned int
- #define **yyerrok** (yyerrstatus = 0)
- #define **yyclearin** (yychar = YYEMPTY)
- #define **YYEMPTY** -2
- #define **YYEOF** 0
- #define **YYACCEPT** goto yyacceptlab
- #define **YYABORT** goto yyabortlab
- #define **YYERROR** goto yyerrlab1
- #define **YYFAIL** goto yyerrlab
- #define **YYRECOVERING**() (!!yyerrstatus)
- #define **YYBACKUP**(Token, Value)
- #define **YYTERROR** 1
- #define **YYERRCODE** 256
- #define **YYLOC_DEFAULT**(Current, Rhs, N)
- #define **YYLEX** yylex ()
- #define **YYDPRINTF**(Args)
- #define **YYINITDEPTH** 200
- #define **YYMAXDEPTH** 10000
- #define **YYPARSE_PARAM_ARG**
- #define **YYPARSE_PARAM_DECL**
- #define **YY_DECL_NON_LSP_VARIABLES**
- #define **YY_DECL_VARIABLES** YY_DECL_NON_LSP_VARIABLES
- #define **YYPPOPSTACK** (yyvsp-, yyssp-)

Functions

- int **newwho** (**buffer_t** *b, char ***file**, int line)
- int **newtup** (**list_t** *vlist, char ***file**, int line)
- **OPCODE_T** **refine_op** (**OPCODE_T** op, **TAG_T** vtag)
- void **yyerror** (char *s)
- **YY_DECL_VARIABLES** int **yyparse** (**YYPARSE_PARAM_ARG**) **YYPARSE_PARAM_DECL**

Variables

- **packet_t** * p
- int **value_int**
- **uint32** **value_addr**
- **buffer_t** **value_str**
- int **value_exc**
- **float32** **value_float**
- int **noop**

5.74.1 Define Documentation

5.74.1.1 #define CHECK_CODE_OVERFLOW(p)

Value:

```
if ((p)->pc >= (p)->code_max) {
    fprintf(stderr,"%s:%d: code overflow\n",
            __FILE__,__LINE__);
    fflush(stderr);
    exit(1);
}
```

Definition at line 130 of file snapparse.c.

Referenced by yyparse().

5.74.1.2 #define T_ADD 278

Definition at line 27 of file snapparse.c.

5.74.1.3 #define T_ADDI 279

Definition at line 28 of file snapparse.c.

5.74.1.4 #define T_ADDRV 258

Definition at line 7 of file snapparse.c.

5.74.1.5 #define T_AND 291

Definition at line 40 of file snapparse.c.

5.74.1.6 #define T_ANDI 292

Definition at line 41 of file snapparse.c.

5.74.1.7 #define T_BCAST 303

Definition at line 52 of file snapparse.c.

5.74.1.8 #define T_BCASTI 304

Definition at line 53 of file snapparse.c.

5.74.1.9 #define T_BEZ 272

Definition at line 21 of file snapparse.c.

5.74.1.10 #define T_BNE 273

Definition at line 22 of file snapparse.c.

5.74.1.11 #define T_CALLS 337

Definition at line 86 of file snapparse.c.

5.74.1.12 #define T_COMMA 335

Definition at line 84 of file snapparse.c.

5.74.1.13 #define T_DATA 346

Definition at line 95 of file snapparse.c.

5.74.1.14 #define T_DEMUX 318

Definition at line 67 of file snapparse.c.

5.74.1.15 #define T_DEMUXI 319

Definition at line 68 of file snapparse.c.

5.74.1.16 #define T_DFORW 347

Definition at line 96 of file snapparse.c.

5.74.1.17 #define T_DFORWTO 348

Definition at line 97 of file snapparse.c.

5.74.1.18 #define T_DIV 284

Definition at line 33 of file snapparse.c.

5.74.1.19 #define T_DIVI 285

Definition at line 34 of file snapparse.c.

5.74.1.20 #define T_DSEND 349

Definition at line 98 of file snapparse.c.

5.74.1.21 #define T_EQ 268

Definition at line 17 of file snapparse.c.

5.74.1.22 #define T_EQI 269

Definition at line 18 of file snapparse.c.

5.74.1.23 #define T_EXCV 260

Definition at line 9 of file snapparse.c.

5.74.1.24 #define T_EXIT 263

Definition at line 12 of file snapparse.c.

5.74.1.25 #define T_FLOATV 261

Definition at line 10 of file snapparse.c.

5.74.1.26 #define T_FORW 316

Definition at line 65 of file snapparse.c.

5.74.1.27 #define T_FORWTO 317

Definition at line 66 of file snapparse.c.

5.74.1.28 #define T_GEQ 339

Definition at line 88 of file snapparse.c.

5.74.1.29 #define T_GEQI 343

Definition at line 92 of file snapparse.c.

5.74.1.30 #define T_GETDST 308

Definition at line 57 of file snapparse.c.

5.74.1.31 #define T_GETLD 321

Definition at line 70 of file snapparse.c.

5.74.1.32 #define T_GETRB 306

Definition at line 55 of file snapparse.c.

5.74.1.33 #define T_GETSPT 309

Definition at line 58 of file snapparse.c.

5.74.1.34 #define T_GETSRC 307

Definition at line 56 of file snapparse.c.

5.74.1.35 #define T_GT 338

Definition at line 87 of file snapparse.c.

5.74.1.36 #define T_GTI 342

Definition at line 91 of file snapparse.c.

5.74.1.37 #define T_HERE 310

Definition at line 59 of file snapparse.c.

5.74.1.38 #define T_HOP 315

Definition at line 64 of file snapparse.c.

5.74.1.39 #define T_INTV 257

Definition at line 6 of file snapparse.c.

5.74.1.40 #define T_ISHERE 311

Definition at line 60 of file snapparse.c.

5.74.1.41 #define T_ISTUP 332

Definition at line 81 of file snapparse.c.

5.74.1.42 #define T_ISX 305

Definition at line 54 of file snapparse.c.

5.74.1.43 #define T_JI 275

Definition at line 24 of file snapparse.c.

5.74.1.44 #define T_LABEL 326

Definition at line 75 of file snapparse.c.

5.74.1.45 #define T_LABELV 327

Definition at line 76 of file snapparse.c.

5.74.1.46 #define T_LEN 330

Definition at line 79 of file snapparse.c.

5.74.1.47 #define T_LEQ 341

Definition at line 90 of file snapparse.c.

5.74.1.48 #define T_LEQI 345

Definition at line 94 of file snapparse.c.

5.74.1.49 #define T_LNOT 290

Definition at line 39 of file snapparse.c.

5.74.1.50 #define T_LPAREN 333

Definition at line 82 of file snapparse.c.

5.74.1.51 #define T_LSHL 295

Definition at line 44 of file snapparse.c.

5.74.1.52 #define T_LSHLI 296

Definition at line 45 of file snapparse.c.

5.74.1.53 #define T_LT 340

Definition at line 89 of file snapparse.c.

5.74.1.54 #define T_LTI 344

Definition at line 93 of file snapparse.c.

5.74.1.55 #define T_MAIN 262

Definition at line 11 of file snapparse.c.

5.74.1.56 #define T_MINUS 325

Definition at line 74 of file snapparse.c.

5.74.1.57 #define T_MKTUP 329

Definition at line 78 of file snapparse.c.

5.74.1.58 #define T_MOD 286

Definition at line 35 of file snapparse.c.

5.74.1.59 #define T_MODI 287

Definition at line 36 of file snapparse.c.

5.74.1.60 #define T_MULT 282

Definition at line 31 of file snapparse.c.

5.74.1.61 #define T_MULTTI 283

Definition at line 32 of file snapparse.c.

5.74.1.62 #define T_NEG 288

Definition at line 37 of file snapparse.c.

5.74.1.63 #define T_NEQ 276

Definition at line 25 of file snapparse.c.

5.74.1.64 #define T_NEQI 277

Definition at line 26 of file snapparse.c.

5.74.1.65 #define T_NOT 289

Definition at line 38 of file snapparse.c.

5.74.1.66 #define T_NTH 331

Definition at line 80 of file snapparse.c.

5.74.1.67 #define T_OR 293

Definition at line 42 of file snapparse.c.

5.74.1.68 #define T_ORI 294

Definition at line 43 of file snapparse.c.

5.74.1.69 #define T_PAJ 270

Definition at line 19 of file snapparse.c.

5.74.1.70 #define T_PC 328

Definition at line 77 of file snapparse.c.

5.74.1.71 #define T_PLUS 324

Definition at line 73 of file snapparse.c.

5.74.1.72 #define T_POP 265

Definition at line 14 of file snapparse.c.

5.74.1.73 #define T_POPI 266

Definition at line 15 of file snapparse.c.

5.74.1.74 #define T_PRINT 320

Definition at line 69 of file snapparse.c.

5.74.1.75 #define T_PULL 267

Definition at line 16 of file snapparse.c.

5.74.1.76 #define T_PULLSTACK 352

Definition at line 101 of file snapparse.c.

5.74.1.77 #define T_PUSH 264

Definition at line 13 of file snapparse.c.

5.74.1.78 #define T_RAISEX 323

Definition at line 72 of file snapparse.c.

5.74.1.79 #define T_ROUTE 312

Definition at line 61 of file snapparse.c.

5.74.1.80 #define T_RPAREN 334

Definition at line 83 of file snapparse.c.

5.74.1.81 #define T_RSHA 299

Definition at line 48 of file snapparse.c.

5.74.1.82 #define T_RSHAI 300

Definition at line 49 of file snapparse.c.

5.74.1.83 #define T_RSHL 297

Definition at line 46 of file snapparse.c.

5.74.1.84 #define T_RSHLI 298

Definition at line 47 of file snapparse.c.

5.74.1.85 #define T_RTDEV 313

Definition at line 62 of file snapparse.c.

5.74.1.86 #define T_SEND 314

Definition at line 63 of file snapparse.c.

5.74.1.87 #define T_SETXH 322

Definition at line 71 of file snapparse.c.

5.74.1.88 #define T_SNET 301

Definition at line 50 of file snapparse.c.

5.74.1.89 #define T_SNETI 302

Definition at line 51 of file snapparse.c.

5.74.1.90 #define T_STACKCOUNT 351

Definition at line 100 of file snapparse.c.

5.74.1.91 #define T_STACKEMPTY 350

Definition at line 99 of file snapparse.c.

5.74.1.92 #define T_STORE 274

Definition at line 23 of file snapparse.c.

5.74.1.93 #define T_STRV 259

Definition at line 8 of file snapparse.c.

5.74.1.94 #define T_SUB 280

Definition at line 29 of file snapparse.c.

5.74.1.95 #define T_SUBI 281

Definition at line 30 of file snapparse.c.

5.74.1.96 #define T_SVCV 336

Definition at line 85 of file snapparse.c.

5.74.1.97 #define T_TPAJ 271

Definition at line 20 of file snapparse.c.

5.74.1.98 #define YY_DECL_NON_LSP_VARIABLES

Value:

```

\
int ychar;
\
\
\
YYSTYPE yylval;
\
\
\
int yynerrs;

```

Definition at line 780 of file snapparse.c.

**5.74.1.99 #define YY_DECL_VARIABLES
YY_DECL_NON_LSP_VARIABLES**

Definition at line 797 of file snapparse.c.

Referenced by yyparse().

5.74.1.100 #define YYABORT goto yyabortlab

Definition at line 591 of file snapparse.c.

Referenced by yyparse().

5.74.1.101 #define YYACCEPT goto yyacceptlab

Definition at line 590 of file snapparse.c.

Referenced by yyparse().

5.74.1.102 #define YYBACKUP(Token, Value)

Value:

```

do
\
if (ychar == YYEMPTY && yylen == 1)
\
{
\
yychar = (Token);
\
yylval = (Value);
\
yychar1 = YYTRANSLATE (yychar);
\
YYPOPSTACK;
\
goto yybackup;
\
}

```

```

    }
else
{
    yyerror ("syntax error: cannot back up");
    YYERROR;
}
while (0)

```

Definition at line 598 of file snapparse.c.

5.74.1.103 #define YYBISON 1

Definition at line 4 of file snapparse.c.

5.74.1.104 #define yyclearin (yychar = YYEMPTY)

Definition at line 587 of file snapparse.c.

5.74.1.105 #define YYCOPY(To, From, Count)

Value:

```

do
{
    register YYSIZE_T yyi;
    for (yyi = 0; yyi < (Count); yyi++)
        (To)[yyi] = (From)[yyi];
}
while (0)

```

Definition at line 540 of file snapparse.c.

5.74.1.106 #define YYDEBUG 0

Definition at line 169 of file snapparse.c.

5.74.1.107 #define YYDPRINTF(Args)

Definition at line 671 of file snapparse.c.

Referenced by yyparse().

5.74.1.108 #define YYEMPTY -2

Definition at line 588 of file snapparse.c.

Referenced by yyparse().

5.74.1.109 #define YYEOF 0

Definition at line 589 of file snapparse.c.

Referenced by yyparse().

5.74.1.110 #define YYERRCODE 256

Definition at line 616 of file snapparse.c.

5.74.1.111 #define yyerrok (yyerrstatus = 0)

Definition at line 586 of file snapparse.c.

5.74.1.112 #define YYERROR goto yyerrlab1

Definition at line 592 of file snapparse.c.

5.74.1.113 #define YYFAIL goto yyerrlab

Definition at line 596 of file snapparse.c.

5.74.1.114 #define YYFINAL 117

Definition at line 174 of file snapparse.c.

Referenced by yyparse().

5.74.1.115 #define YYFLAG -32768

Definition at line 175 of file snapparse.c.

Referenced by yyparse().

5.74.1.116 #define YYINITDEPTH 200

Definition at line 676 of file snapparse.c.

Referenced by yyparse().

5.74.1.117 #define YYLAST 167

Definition at line 388 of file snapparse.c.

Referenced by yyparse().

5.74.1.118 #define YYLEX yylex ()

Definition at line 650 of file snapparse.c.

Referenced by yyparse().

5.74.1.119 #define YYLLOC_DEFAULT(Current, Rhs, N)

Value:

```
Current.last_line = Rhs[N].last_line; \  
Current.last_column = Rhs[N].last_column;
```

Definition at line 627 of file snapparse.c.

Referenced by yyparse().

5.74.1.120 #define YYMAXDEPTH 10000

Definition at line 691 of file snapparse.c.

Referenced by yyparse().

5.74.1.121 #define YYNTBASE 99

Definition at line 176 of file snapparse.c.

Referenced by yyparse().

5.74.1.122 #define YYPARSE_PARAM_ARG

Definition at line 764 of file snapparse.c.

Referenced by yyparse().

5.74.1.123 #define YYPARSE_PARAM_DECL

Definition at line 765 of file snapparse.c.

Referenced by yyparse().

5.74.1.124 #define YYPOPSTACK (yyvsp-, yyssp-)**5.74.1.125 #define YYRECOVERING() (!!yyerrstatus)**

Definition at line 597 of file snapparse.c.

5.74.1.126 #define YYSIZE_T unsigned int

Definition at line 583 of file snapparse.c.

Referenced by yyparse().

5.74.1.127 #define YYSTACK_ALLOC malloc

Definition at line 498 of file snapparse.c.

Referenced by yyparse().

5.74.1.128 #define YYSTACK_BYTES(N)

Value:

```
((N) * (sizeof (short) + sizeof (YYSTYPE))
      + YYSTACK_GAP_MAX) \
```

Definition at line 528 of file snapparse.c.

5.74.1.129 #define YYSTACK_FREE free

Definition at line 499 of file snapparse.c.

Referenced by yyparse().

5.74.1.130 #define YYSTACK_GAP_MAX (sizeof (union yyallocc) - 1)

Definition at line 519 of file snapparse.c.

5.74.1.131 #define YYSTACK_RELOCATE(Stack)

Value:

```
do
  {
    YYSIZE_T yynewbytes;
    YYCOPY (&yyptr->Stack, Stack, yysize);
    Stack = &yyptr->Stack;
    yynewbytes = yystacksize * sizeof (*Stack) + YYSTACK_GAP_MAX;
    yyptr += yynewbytes / sizeof (*yyptr);
  }
  while (0)
```

Definition at line 556 of file snapparse.c.

Referenced by yyparse().

5.74.1.132 `#define YYSTYPE ystype`

Definition at line 165 of file snapparse.c.

Referenced by `yyparse()`.

5.74.1.133 `#define YYSTYPE_IS_TRIVIAL 1`

Definition at line 166 of file snapparse.c.

5.74.1.134 `#define YYTERROR 1`

Definition at line 615 of file snapparse.c.

Referenced by `yyparse()`.

5.74.1.135 `#define YYTRANSLATE(x) ((unsigned)(x) <= 352 ?
yytranslate[x] : 109)`

Definition at line 179 of file snapparse.c.

Referenced by `yyparse()`.

5.74.2 Function Documentation

5.74.2.1 `int newwho (buffer_t * b, char * file, int line)`

Definition at line 1924 of file snapparse.c.

References `d_printf()`, `file`, `heap_obj::flag`, `packet_t::h_alloc_ptr`, `packet_t::heap_max`, `packet_t::heap_min`, `heap_obj::len`, `buffer_t::lenb`, `MAX_HEAPOBJ_SZ`, `buffer_t::s`, and `heap_obj::s`.

Referenced by `yyparse()`.

5.74.2.2 `int newtup (list_t * vlist, char * file, int line)`

Definition at line 1952 of file snapparse.c.

References `d_printf()`, `file`, `heap_obj::flag`, `packet_t::h_alloc_ptr`, `packet_t::heap_max`, `packet_t::heap_min`, `heap_obj::len`, `len`, `length_list()`, `MAX_HEAPOBJ_SZ`, `l::next`, `heap_obj::s`, `l::v`, and `value_t`.

Referenced by `yyparse()`.

5.74.2.3 `OPCODE_T refine_op (OPCODE_T op, TAG_T vtag)`

Definition at line 1988 of file snapparse.c.

References ADDI, ADDR, DIVI, EQADR, EQEXC, EQFLT, EQI, EQINT, EQSTR, EQTUP, EXCV, FADDI, FDIVI, FGEQI, FGTI, FLEQI, FLOATV, FLTI, FMULI, FSUBI, GEQI, GTI, INTV, LEQI, LTI, MULTI, NEQI, NQADR, NQEXC, NQFLT, NQINT, NQSTR, NQTUP, OPCODE_T, PADDR, PEXC, PFLT, PINT, PSTR, PTUP, PUSH, STRV, SUBI, TAG_T, and TUPLEV.

Referenced by `yyparse()`.

5.74.2.4 `void yyerror (char * s)`

Definition at line 2177 of file `snapparse.c`.

Referenced by `yyparse()`.

5.74.2.5 `YY_DECL_VARIABLES int yyparse (YYPARSE_PARAM_ARG)`

Definition at line 809 of file `snapparse.c`.

References ADD, ADDI, ADDR, AND, ANDI, BCAST, BCASTI, BEZ, BNE, CALLS, CHECK_CODE_OVERFLOW, `packet_t::code_min`, `cons()`, COPY_VAL, DEMUX, DEMUXI, DFORW, DFORWTO, DIV, DIVI, DSEND, `snaphdr::entry_point`, EQ, EQI, EXCV, EXIT, FLOATV, FORW, FORWTO, `free_list()`, GEQ, GEQI, GET_INT, GET_TAG, GETDST, GETLD, GETRB, GETSPT, GETSRC, GT, GTI, `packet_t::hdr`, HERE, HOP, INTV, ISHERE, ISTUP, ISX, JI, LEN, LEQ, LEQI, LNOT, LSHL, LSHLI, LT, LTI, MAX_VINT, `memalloc`, MIN_VINT, MKTUP, MOD, MODI, MULT, MULTI, NEG, NEQ, NEQI, `newwho()`, `newtup()`, noop, NOT, NTH, OPCODE_T, OR, ORI, PAJ, `packet_t::pc`, POP, POPI, PRINT, PULL, PULLSTACK, PUSH, RAISEX, `refine_op()`, `register_label_def()`, `register_label_use()`, ROUTE, RSHA, RSHAI, RSHL, RSHLI, RTDEV, `buffer_t::s`, SEND, SET_ADDR, SET_FLOAT, SET_INT, SET_LIT, SET_OFFSETS, SET_OP, SET_TAG, SETXH, SNET, SNETI, `packet_t::sp`, `packet_t::stack_max`, STACKCOUNT, STACKEMPTY, STORE, STRV, SUB, SUBI, SVCV, TPAJ, TUPLEV, `value_addr`, `value_exc`, `value_float`, `value_int`, `value_t`, YY_DECL_VARIABLES, YYABORT, YYACCEPT, YYDPRINTF, YYEMPTY, YYEOF, `yyerror()`, YYFINAL, YYFLAG, YYINITDEPTH, YYLAST, YYLEX, YYLOC_DEFAULT, `yylval`, YYMAXDEPTH, YYNTBASE, YYPARSE_PARAM_ARG, YYPARSE_PARAM_DECL, YYSIZE_T, YYSTACK_ALLOC, YYSTACK_FREE, YYS-TACK_RELOCATE, YYSTYPE, YYTERROR, and YYTRANSLATE.

5.74.3 Variable Documentation

5.74.3.1 `int noop`

Definition at line 154 of file `snapparse.c`.

Referenced by `main()`, and `yyparse()`.

5.74.3.2 packet_t* p

Definition at line 128 of file snapparse.c.

5.74.3.3 uint32 value_addr

Definition at line 149 of file snapparse.c.

Referenced by yyparse().

5.74.3.4 int value_exc

Definition at line 151 of file snapparse.c.

Referenced by yyparse().

5.74.3.5 float32 value_float

Definition at line 152 of file snapparse.c.

Referenced by yyparse().

5.74.3.6 int value_int

Definition at line 148 of file snapparse.c.

Referenced by yyparse().

5.74.3.7 buffer_t value_str

Definition at line 150 of file snapparse.c.

5.75 snap-1.1-wjdb/utils/snapparse.tab.h File Reference

Data Structures

- union `ystype`

Defines

- `#define YYSTYPE ystype`
- `#define T_INTV 257`
- `#define T_ADDRV 258`
- `#define T_STRV 259`
- `#define T_EXCV 260`
- `#define T_FLOATV 261`
- `#define T_MAIN 262`
- `#define T_EXIT 263`
- `#define T_PUSH 264`
- `#define T_POP 265`
- `#define T_POPI 266`
- `#define T_PULL 267`
- `#define T_EQ 268`
- `#define T_EQI 269`
- `#define T_PAJ 270`
- `#define T_TPAJ 271`
- `#define T_BEZ 272`
- `#define T_BNE 273`
- `#define T_STORE 274`
- `#define T_JI 275`
- `#define T_NEQ 276`
- `#define T_NEQI 277`
- `#define T_ADD 278`
- `#define T_ADDI 279`
- `#define T_SUB 280`
- `#define T_SUBI 281`
- `#define T_MULT 282`
- `#define T_MULTI 283`
- `#define T_DIV 284`
- `#define T_DIVI 285`
- `#define T_MOD 286`
- `#define T_MODI 287`
- `#define T_NEG 288`
- `#define T_NOT 289`
- `#define T_LNOT 290`
- `#define T_AND 291`

- #define **T_ANDI** 292
- #define **T_OR** 293
- #define **T_ORI** 294
- #define **T_LSHL** 295
- #define **T_LSHLI** 296
- #define **T_RSHL** 297
- #define **T_RSHLI** 298
- #define **T_RSHA** 299
- #define **T_RSHAI** 300
- #define **T_SNET** 301
- #define **T_SNETI** 302
- #define **T_BCAST** 303
- #define **T_BCASTI** 304
- #define **T_ISX** 305
- #define **T_GETRB** 306
- #define **T_GETSRC** 307
- #define **T_GETDST** 308
- #define **T_GETSPT** 309
- #define **T_HERE** 310
- #define **T_ISHERE** 311
- #define **T_ROUTE** 312
- #define **T_RTDEV** 313
- #define **T_SEND** 314
- #define **T_HOP** 315
- #define **T_FORW** 316
- #define **T_FORWTO** 317
- #define **T_DEMUX** 318
- #define **T_DEMUXI** 319
- #define **T_PRINT** 320
- #define **T_GETLD** 321
- #define **T_SETXH** 322
- #define **T_RAISEX** 323
- #define **T_PLUS** 324
- #define **T_MINUS** 325
- #define **T_LABEL** 326
- #define **T_LABELV** 327
- #define **T_PC** 328
- #define **T_MKTUP** 329
- #define **T_LEN** 330
- #define **T_NTH** 331
- #define **T_ISTUP** 332
- #define **T_LPAREN** 333
- #define **T_RPAREN** 334
- #define **T_COMMA** 335
- #define **T_SVCV** 336
- #define **T_CALLS** 337

- `#define T_GT` 338
- `#define T_GEQ` 339
- `#define T_LT` 340
- `#define T_LEQ` 341
- `#define T_GTI` 342
- `#define T_GEQI` 343
- `#define T_LTI` 344
- `#define T_LEQI` 345
- `#define T_DATA` 346
- `#define T_DFORW` 347
- `#define T_DFORWTO` 348
- `#define T_DSEND` 349

Variables

- `YYSTYPE` `yylval`

5.75.1 Define Documentation

5.75.1.1 `#define T_ADD` 278

Definition at line 32 of file `snapparse.tab.h`.

5.75.1.2 `#define T_ADDI` 279

Definition at line 33 of file `snapparse.tab.h`.

5.75.1.3 `#define T_ADDRV` 258

Definition at line 12 of file `snapparse.tab.h`.

5.75.1.4 `#define T_AND` 291

Definition at line 45 of file `snapparse.tab.h`.

5.75.1.5 `#define T_ANDI` 292

Definition at line 46 of file `snapparse.tab.h`.

5.75.1.6 `#define T_BCAST` 303

Definition at line 57 of file `snapparse.tab.h`.

5.75.1.7 #define T_BCASTI 304

Definition at line 58 of file snapparse.tab.h.

5.75.1.8 #define T_BEZ 272

Definition at line 26 of file snapparse.tab.h.

5.75.1.9 #define T_BNE 273

Definition at line 27 of file snapparse.tab.h.

5.75.1.10 #define T_CALLS 337

Definition at line 91 of file snapparse.tab.h.

5.75.1.11 #define T_COMMA 335

Definition at line 89 of file snapparse.tab.h.

5.75.1.12 #define T_DATA 346

Definition at line 100 of file snapparse.tab.h.

5.75.1.13 #define T_DEMUX 318

Definition at line 72 of file snapparse.tab.h.

5.75.1.14 #define T_DEMUXI 319

Definition at line 73 of file snapparse.tab.h.

5.75.1.15 #define T_DFORW 347

Definition at line 101 of file snapparse.tab.h.

5.75.1.16 #define T_DFORWTO 348

Definition at line 102 of file snapparse.tab.h.

5.75.1.17 #define T_DIV 284

Definition at line 38 of file snapparse.tab.h.

5.75.1.18 #define T_DIVI 285

Definition at line 39 of file snapparse.tab.h.

5.75.1.19 #define T_DSEND 349

Definition at line 103 of file snapparse.tab.h.

5.75.1.20 #define T_EQ 268

Definition at line 22 of file snapparse.tab.h.

5.75.1.21 #define T_EQI 269

Definition at line 23 of file snapparse.tab.h.

5.75.1.22 #define T_EXCV 260

Definition at line 14 of file snapparse.tab.h.

5.75.1.23 #define T_EXIT 263

Definition at line 17 of file snapparse.tab.h.

5.75.1.24 #define T_FLOATV 261

Definition at line 15 of file snapparse.tab.h.

5.75.1.25 #define T_FORW 316

Definition at line 70 of file snapparse.tab.h.

5.75.1.26 #define T_FORWTO 317

Definition at line 71 of file snapparse.tab.h.

5.75.1.27 #define T_GEQ 339

Definition at line 93 of file snapparse.tab.h.

5.75.1.28 #define T_GEQI 343

Definition at line 97 of file snapparse.tab.h.

5.75.1.29 #define T_GETDST 308

Definition at line 62 of file snapparse.tab.h.

5.75.1.30 #define T_GETLD 321

Definition at line 75 of file snapparse.tab.h.

5.75.1.31 #define T_GETRB 306

Definition at line 60 of file snapparse.tab.h.

5.75.1.32 #define T_GETSPT 309

Definition at line 63 of file snapparse.tab.h.

5.75.1.33 #define T_GETSRC 307

Definition at line 61 of file snapparse.tab.h.

5.75.1.34 #define T_GT 338

Definition at line 92 of file snapparse.tab.h.

5.75.1.35 #define T_GTI 342

Definition at line 96 of file snapparse.tab.h.

5.75.1.36 #define T_HERE 310

Definition at line 64 of file snapparse.tab.h.

5.75.1.37 #define T_HOP 315

Definition at line 69 of file snapparse.tab.h.

5.75.1.38 #define T_INTV 257

Definition at line 11 of file snapparse.tab.h.

5.75.1.39 #define T_ISHERE 311

Definition at line 65 of file snapparse.tab.h.

5.75.1.40 #define T_ISTUP 332

Definition at line 86 of file snapparse.tab.h.

5.75.1.41 #define T_ISX 305

Definition at line 59 of file snapparse.tab.h.

5.75.1.42 #define T_JI 275

Definition at line 29 of file snapparse.tab.h.

5.75.1.43 #define T_LABEL 326

Definition at line 80 of file snapparse.tab.h.

5.75.1.44 #define T_LABELV 327

Definition at line 81 of file snapparse.tab.h.

5.75.1.45 #define T_LEN 330

Definition at line 84 of file snapparse.tab.h.

5.75.1.46 #define T_LEQ 341

Definition at line 95 of file snapparse.tab.h.

5.75.1.47 #define T_LEQI 345

Definition at line 99 of file snapparse.tab.h.

5.75.1.48 #define T_LNOT 290

Definition at line 44 of file snapparse.tab.h.

5.75.1.49 #define T_LPAREN 333

Definition at line 87 of file snapparse.tab.h.

5.75.1.50 #define T_LSHL 295

Definition at line 49 of file snapparse.tab.h.

5.75.1.51 #define T_LSHLI 296

Definition at line 50 of file snapparse.tab.h.

5.75.1.52 #define T_LT 340

Definition at line 94 of file snapparse.tab.h.

5.75.1.53 #define T_LTI 344

Definition at line 98 of file snapparse.tab.h.

5.75.1.54 #define T_MAIN 262

Definition at line 16 of file snapparse.tab.h.

5.75.1.55 #define T_MINUS 325

Definition at line 79 of file snapparse.tab.h.

5.75.1.56 #define T_MKTUP 329

Definition at line 83 of file snapparse.tab.h.

5.75.1.57 #define T_MOD 286

Definition at line 40 of file snapparse.tab.h.

5.75.1.58 #define T_MODI 287

Definition at line 41 of file snapparse.tab.h.

5.75.1.59 #define T_MULT 282

Definition at line 36 of file snapparse.tab.h.

5.75.1.60 #define T_MULTI 283

Definition at line 37 of file snapparse.tab.h.

5.75.1.61 #define T_NEG 288

Definition at line 42 of file snapparse.tab.h.

5.75.1.62 #define T_NEQ 276

Definition at line 30 of file snapparse.tab.h.

5.75.1.63 #define T_NEQI 277

Definition at line 31 of file snapparse.tab.h.

5.75.1.64 #define T_NOT 289

Definition at line 43 of file snapparse.tab.h.

5.75.1.65 #define T_NTH 331

Definition at line 85 of file snapparse.tab.h.

5.75.1.66 #define T_OR 293

Definition at line 47 of file snapparse.tab.h.

5.75.1.67 #define T_ORI 294

Definition at line 48 of file snapparse.tab.h.

5.75.1.68 #define T_PAJ 270

Definition at line 24 of file snapparse.tab.h.

5.75.1.69 #define T_PC 328

Definition at line 82 of file snapparse.tab.h.

5.75.1.70 #define T_PLUS 324

Definition at line 78 of file snapparse.tab.h.

5.75.1.71 #define T_POP 265

Definition at line 19 of file snapparse.tab.h.

5.75.1.72 #define T_POPI 266

Definition at line 20 of file snapparse.tab.h.

5.75.1.73 #define T_PRINT 320

Definition at line 74 of file snapparse.tab.h.

5.75.1.74 #define T_PULL 267

Definition at line 21 of file snapparse.tab.h.

5.75.1.75 #define T_PUSH 264

Definition at line 18 of file snapparse.tab.h.

5.75.1.76 #define T_RAISEX 323

Definition at line 77 of file snapparse.tab.h.

5.75.1.77 #define T_ROUTE 312

Definition at line 66 of file snapparse.tab.h.

5.75.1.78 #define T_RPAREN 334

Definition at line 88 of file snapparse.tab.h.

5.75.1.79 #define T_RSHA 299

Definition at line 53 of file snapparse.tab.h.

5.75.1.80 #define T_RSHAI 300

Definition at line 54 of file snapparse.tab.h.

5.75.1.81 #define T_RSHL 297

Definition at line 51 of file snapparse.tab.h.

5.75.1.82 #define T_RSHLI 298

Definition at line 52 of file snapparse.tab.h.

5.75.1.83 #define T_RTDEV 313

Definition at line 67 of file snapparse.tab.h.

5.75.1.84 #define T_SEND 314

Definition at line 68 of file snapparse.tab.h.

5.75.1.85 #define T_SETXH 322

Definition at line 76 of file snapparse.tab.h.

5.75.1.86 #define T_SNET 301

Definition at line 55 of file snapparse.tab.h.

5.75.1.87 #define T_SNETI 302

Definition at line 56 of file snapparse.tab.h.

5.75.1.88 #define T_STORE 274

Definition at line 28 of file snapparse.tab.h.

5.75.1.89 #define T_STRV 259

Definition at line 13 of file snapparse.tab.h.

5.75.1.90 #define T_SUB 280

Definition at line 34 of file snapparse.tab.h.

5.75.1.91 #define T_SUBI 281

Definition at line 35 of file snapparse.tab.h.

5.75.1.92 #define T_SVCV 336

Definition at line 90 of file snapparse.tab.h.

5.75.1.93 #define T_TPAJ 271

Definition at line 25 of file snapparse.tab.h.

5.75.1.94 #define YYSTYPE ystype

Definition at line 9 of file snapparse.tab.h.

5.75.2 Variable Documentation

5.75.2.1 YYSTYPE yylval

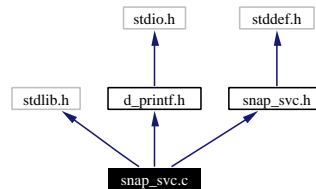
Definition at line 106 of file snapparse.tab.h.

Referenced by yyparse().

5.76 snap_svc/snap_svc.c File Reference

```
#include <stdlib.h>
#include "d_printf.h"
#include "snap_svc.h"
```

Include dependency graph for snap_svc.c:



Functions

- void **init** ()
- void **fini** ()
- **svc_returnstruct * snap_external_svclib_free_returnstruct** (struct **svc_returnstruct *svc_return**)
- **svc_returnstruct * snap_external_svclib_free_local_returnstruct** ()
- **svc_returnstruct * snap_external_svclib_getlastresult** ()

5.76.1 Function Documentation

5.76.1.1 void fini ()

Definition at line 22 of file snap_svc.c.

References [d_printf\(\)](#), [snap_external_svclib_done\(\)](#), [snap_external_svclib_free_returnstruct\(\)](#), [svc_fun_counter](#), and [svc_return](#).

5.76.1.2 void init ()

Definition at line 14 of file snap_svc.c.

References [d_printf\(\)](#), [snap_external_svclib_init\(\)](#), [svc_fun_counter](#), and [svc_return](#).

5.76.1.3 struct svc_returnstruct* snap_external_svclib_free_local_returnstruct ()

Definition at line 54 of file snap_svc.c.

References [snap_external_svclib_free_returnstruct\(\)](#).

Referenced by `if_getallneighbours()`, `snap_external_svclib_snmp_getallotherneighboursfromip()`, `snap_external_svclib_snmp_getnexthopfromip()`, and `snap_external_svclib_snmp_INTERNAL_execpdu_handler()`.

5.76.1.4 `struct svc_returnstruct* snap_external_svclib_free_returnstruct (struct svc_returnstruct * svc_return)`

Definition at line 33 of file `snap_svc.c`.

References `svc_returnitem::data`, `svc_returnstruct::length`, `svc_returnstruct::list`, and `svc_returnitem::oid`.

Referenced by `fini()`, and `snap_external_svclib_free_local_returnstruct()`.

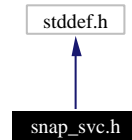
5.76.1.5 `struct svc_returnstruct* snap_external_svclib_getlastresult ()`

Definition at line 62 of file `snap_svc.c`.

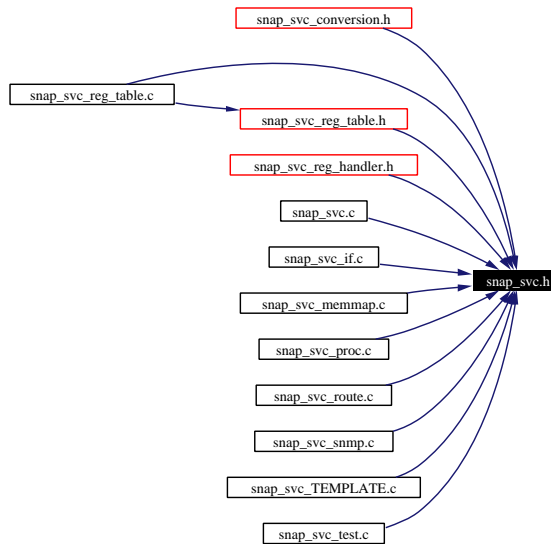
5.77 snap_svc/snap_svc.h File Reference

```
#include <stddef.h>
```

Include dependency graph for snap_svc.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **svc_returnitem**
- struct **svc_returnstruct**

Typedefs

- typedef void *(* **snapsvc_func_proto**)(void *,...)
- typedef void(* **snap_svc_init**)(void)
- typedef void(* **snap_svc_register**)(char **, **snapsvc_func_proto** *, int *, int *)
- typedef void *(* **snap_svc_getlastresult**)(void)
- typedef void *(* **snap_svc_free_local_returnstruct**)(void)

Enumerations

- enum { SVC_SNMP_TYPE_NULL, SVC_SNMP_TYPE_INT, SVC_SNMP_TYPE_ADDR, SVC_SNMP_TYPE_STRING, SVC_SNMP_TYPE_LONG }

Functions

- void **init** ()
- void **fini** ()
- void **snap_external_svclib_init** ()
- void **snap_external_svclib_done** ()
- void **snap_external_svclib_getnextfunc** (char **, snapsvc_func_proto *, int *, int *)
- svc_returnstruct * **snap_external_svclib_getlastresult** ()
- svc_returnstruct * **snap_external_svclib_free_returnstruct** (struct svc_returnstruct *)
- svc_returnstruct * **snap_external_svclib_free_local_returnstruct** ()

Variables

- unsigned int **svc_fun_counter**
- svc_returnstruct * **svc_return**

5.77.1 Typedef Documentation

5.77.1.1 typedef void>(* snap_svc_free_local_returnstruct)(void)

Definition at line 21 of file snap_svc.h.

5.77.1.2 typedef void>(* snap_svc_getlastresult)(void)

Definition at line 20 of file snap_svc.h.

5.77.1.3 typedef void(* snap_svc_init)(void)

Definition at line 18 of file snap_svc.h.

Referenced by `snap_svc_registerlib()`, and `snap_svc_unregisterlib()`.

5.77.1.4 typedef void(* snap_svc_register)(char**, snapsvc_func_proto*, int*, int*)

Definition at line 19 of file snap_svc.h.

Referenced by `snap_svc_registerlib()`.

5.77.1.5 typedef void*(* snapsvc_func_proto)(void*, ...)

if it doesn't use arguments you should create a dummy and set args=0

Definition at line 15 of file snap_svc.h.

Referenced by snap_external_svclib_getnextfunc(), snap_svc_registerlib(), and snap_svc_table_add().

5.77.2 Enumeration Type Documentation

5.77.2.1 anonymous enum

Enumeration values:

SVC_SNMP_TYPE_NULL
SVC_SNMP_TYPE_INT
SVC_SNMP_TYPE_ADDR
SVC_SNMP_TYPE_STRING
SVC_SNMP_TYPE_LONG

Definition at line 24 of file snap_svc.h.

5.77.3 Function Documentation

5.77.3.1 void fini ()

Definition at line 22 of file snap_svc.c.

References d_printf(), snap_external_svclib_done(), snap_external_svclib_free_returnstruct(), svc_fun_counter, and svc_return.

5.77.3.2 void init ()

Definition at line 14 of file snap_svc.c.

References d_printf(), snap_external_svclib_init(), svc_fun_counter, and svc_return.

5.77.3.3 void snap_external_svclib_done ()

Definition at line 39 of file snap_svc_if.c.

References d_printf().

Referenced by fini().

5.77.3.4 struct svc_returnstruct* snap_external_svclib_free_local_returnstruct ()

Definition at line 54 of file snap_svc.c.

References snap_external_svclib_free_returnstruct().

Referenced by if_getallneighbours(), snap_external_svclib_snmp_getallotherneighboursfromip(), snap_external_svclib_snmp_getnexthopfromip(), and snap_external_svclib_snmp_INTERNAL_execpdu_handler().

5.77.3.5 struct svc_returnstruct* snap_external_svclib_free_returnstruct (struct svc_returnstruct *)

Definition at line 33 of file snap_svc.c.

References svc_returnitem::data, svc_returnstruct::length, svc_returnstruct::list, and svc_returnitem::oid.

Referenced by fini(), and snap_external_svclib_free_local_returnstruct().

5.77.3.6 struct svc_returnstruct* snap_external_svclib_getlastresult ()

Definition at line 62 of file snap_svc.c.

5.77.3.7 void snap_external_svclib_getnextfunc (char **, snapsvc_func_proto *, int *, int *)

Definition at line 53 of file snap_svc_if.c.

References snap_external_svclib_testfunc(), snap_external_svclib_testintfunc(), snap_external_svclib_teststrfunc(), snapsvc_func_proto, svc_fun_counter, and SVC_SNMP_TYPE_NULL.

5.77.3.8 void snap_external_svclib_init ()

Definition at line 31 of file snap_svc_if.c.

References d_printf().

Referenced by init().

5.77.4 Variable Documentation**5.77.4.1 unsigned int svc_fun_counter**

Definition at line 42 of file snap_svc.h.

Referenced by fini(), init(), and snap_external_svclib_getnextfunc().

5.77.4.2 struct svc_returnstruct* svc_return

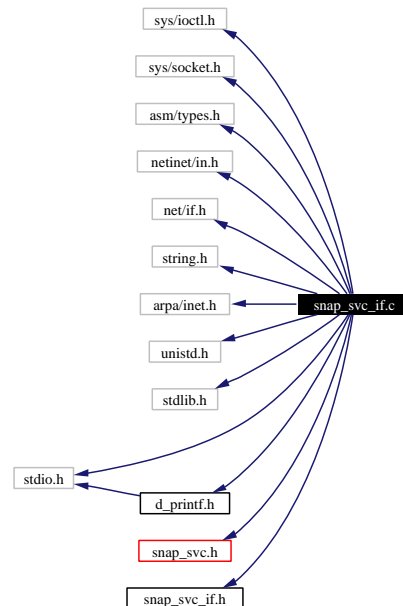
Definition at line 43 of file snap_svc.h.

Referenced by fini(), if_getallneighbours(), init(), snap_external_svclib_snmp_getallotherneighboursfromip(), snap_external_svclib_snmp_gethop(), snap_external_svclib_snmp_getiface(), snap_external_svclib_snmp_getifnumber(), snap_external_svclib_snmp_getnexthopfromip(), snap_external_svclib_snmp_INTERNAL_execpdu_handler(), and snap_external_svclib_snmp_isupiface().

5.78 snap_svc/snap_svc_if.c File Reference

```
#include <sys/ioctl.h>
#include <sys/socket.h>
#include <asm/types.h>
#include <netinet/in.h>
#include <net/if.h>
#include <string.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include "d_printf.h"
#include "snap_svc.h"
#include "snap_svc_if.h"
```

Include dependency graph for snap_svc_if.c:



Functions

- void `snap_external_svclib_init` ()
- void `snap_external_svclib_done` ()

- void **snap_external_svclib_getnextfunc** (char **snapsvc_name, snapsvc_func_proto *snapsvc_func, int *snapsvc_args, int *snapsvc_rets)
- int **snap_svc_ifip_init** ()
- int **if_get_interface_count** (void *useless)
- char * **if_get_interface_name** (int dIfNo)
- int **if_getiface** (const char *strIfName, int flag)
- int **if_getiface_up** (const char *strIfName)
- int **if_setiface** (const char *strIfName, int flag, int state)
- int **if_setiface_up** (const char *strIfName, int state)
- unsigned int **if_getnextiface** (unsigned int iface_idx)
- unsigned int **if_getifaceidx** (uint32_t ip)
- char * **if_getoutiface** (uint32_t incoming_ip)
- uint32_t **if_gethopfromiface** (char *if_outname)
- uint32_t **if_getnexthop** (uint32_t incoming_ip)
- void **if_getallneighbours** (uint32_t incoming_ip)

Variables

- unsigned int **snap_svc_if_count** = 0
- unsigned int **snap_svc_if_maxidx** = 0
- **snap_svc_ifip_item** * **iface_list** = NULL

5.78.1 Function Documentation

5.78.1.1 int if_get_interface_count (void * *useless*)

Definition at line 174 of file snap_svc.if.c.

References `d_printf()`.

Referenced by `snap_external_svclib_getnextfunc()`.

5.78.1.2 char* if_get_interface_name (int *dIfNo*)

Definition at line 201 of file snap_svc.if.c.

References `d_printf()`.

Referenced by `snap_external_svclib_getnextfunc()`.

5.78.1.3 void if_getallneighbours (uint32_t *incoming_ip*)

Definition at line 401 of file snap_svc.if.c.

References `snap_svc_ifip_item::addr`, `d_printf()`, `svc_returnitem::data`, `if_gethopfromiface()`, `if_getifaceidx()`, `if_getnextiface()`, `svc_returnstruct::length`, `svc_returnstruct::list`, `snap_external_svclib_free_local_returnstruct()`, `svc_return`, `SVC_SNMP_TYPE_ADDR`, and `svc_returnitem::type`.

Referenced by snap_external_svclib_getnextfunc().

5.78.1.4 uint32_t if_gethopfromiface (char * *if_outname*)

Definition at line 356 of file snap_svc_if.c.

References d_printf().

Referenced by if_getallneighbours(), and if_getnexthop().

5.78.1.5 int if_getiface (const char * *strIfName*, int *flag*)

Definition at line 225 of file snap_svc_if.c.

References d_printf().

Referenced by if_getiface_up(), and snap_external_svclib_getnextfunc().

5.78.1.6 int if_getiface_up (const char * *strIfName*)

Definition at line 242 of file snap_svc_if.c.

References if_getiface().

Referenced by if_setiface(), and snap_external_svclib_getnextfunc().

5.78.1.7 unsigned int if_getifaceidx (uint32_t *ip*)

Definition at line 319 of file snap_svc_if.c.

References snap_svc_ifip_item::addr, d_printf(), snap_svc_ifip_item::if_index, and snap_svc_if_maxidx.

Referenced by if_getallneighbours(), and if_getoutiface().

5.78.1.8 uint32_t if_getnexthop (uint32_t *incoming_ip*)

Definition at line 392 of file snap_svc_if.c.

References if_gethopfromiface(), and if_getoutiface().

Referenced by snap_external_svclib_getnextfunc().

5.78.1.9 unsigned int if_getnextiface (unsigned int *iface_idx*)

Definition at line 295 of file snap_svc_if.c.

References snap_svc_ifip_item::addr, d_printf(), and snap_svc_if_maxidx.

Referenced by if_getallneighbours(), and if_getoutiface().

5.78.1.10 char* if_getoutiface (uint32_t *incoming_ip*)

Definition at line 334 of file snap_svc.if.c.

References `d_printf()`, `if_getifaceidx()`, `if_getnextiface()`, `snap_svc_ifip_item::if_name`, and `snap_svc_ifip_init()`.

Referenced by `if_getnexthop()`.

5.78.1.11 int if_setiface (const char * *strIfName*, int *flag*, int *state*)

Definition at line 248 of file snap_svc.if.c.

References `d_printf()`, and `if_getiface_up()`.

Referenced by `if_setiface_up()`, and `snap_external_svclib_getnextfunc()`.

5.78.1.12 int if_setiface_up (const char * *strIfName*, int *state*)

Definition at line 273 of file snap_svc.if.c.

References `if_setiface()`.

Referenced by `snap_external_svclib_getnextfunc()`.

5.78.1.13 void snap_external_svclib_done ()

Definition at line 39 of file snap_svc.if.c.

References `d_printf()`, `snap_svc_ifip_item::if_name`, and `snap_svc_if_maxidx`.

5.78.1.14 void snap_external_svclib_getnextfunc (char ** *snapsvc_name*, snapsvc_func_proto * *snapsvc_func*, int * *snapsvc_args*, int * *snapsvc_rets*)

Definition at line 53 of file snap_svc.if.c.

References `if_get_interface_count()`, `if_get_interface_name()`, `if_getallneighbours()`, `if_getiface()`, `if_getiface_up()`, `if_getnexthop()`, `if_setiface()`, `if_setiface_up()`, `snapsvc_func_proto`, `svc_fun_counter`, `SVC_SNMP_TYPE_ADDR`, `SVC_SNMP_TYPE_INT`, `SVC_SNMP_TYPE_NULL`, and `SVC_SNMP_TYPE_STRING`.

5.78.1.15 void snap_external_svclib_init ()

Definition at line 31 of file snap_svc.if.c.

References `d_printf()`, and `snap_svc_ifip_init()`.

5.78.1.16 int snap_svc_ifip_init ()

Definition at line 109 of file snap_svc_if.c.

References `d_printf()`, `snap_svc_ifip_item::if_index`, `snap_svc_if_count`, and `snap_svc_if_maxidx`.

Referenced by `if_getoutiface()`, and `snap_external_svclib_init()`.

5.78.2 Variable Documentation**5.78.2.1 struct snap_svc_ifip_item* iface_list = NULL**

Definition at line 28 of file snap_svc_if.c.

5.78.2.2 unsigned int snap_svc_if_count = 0

Definition at line 26 of file snap_svc_if.c.

Referenced by `snap_svc_ifip_init()`.

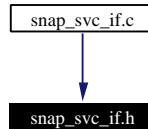
5.78.2.3 unsigned int snap_svc_if_maxidx = 0

Definition at line 27 of file snap_svc_if.c.

Referenced by `if_getifaceidx()`, `if_getnextiface()`, `snap_external_svclib_done()`, and `snap_svc_ifip_init()`.

5.79 snap_svc/snap_svc_if.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **snap_svc_ifp_item**

Functions

- int **if_get_interface_count** (void *useless)
- char * **if_get_interface_name** (int dIfNo)
- int **if_getiface** (const char *strIfName, int flag)
- int **if_getiface_up** (const char *strIfName)
- int **if_setiface** (const char *strIfName, int flag, int state)
- int **if_setiface_up** (const char *strIfName, int state)
- uint32_t **if_getnexthop** (uint32_t incoming_ip)
- void **if_getallneighbours** (uint32_t incoming_ip)
- int **snap_svc_ifp_init** ()

5.79.1 Function Documentation

5.79.1.1 int if_get_interface_count (void * *useless*)

Definition at line 174 of file snap_svc.if.c.

References `d_printf()`.

Referenced by `snap_external_svclib_getnextfunc()`.

5.79.1.2 char* if_get_interface_name (int *dIfNo*)

Definition at line 201 of file snap_svc.if.c.

References `d_printf()`.

Referenced by `snap_external_svclib_getnextfunc()`.

5.79.1.3 void if_getallneighbours (uint32_t *incoming_ip*)

Definition at line 401 of file snap_svc_if.c.

References snap_svc_ifip_item::addr, d_printf(), svc_returnitem::data, if_gethopfromiface(), if_getifaceidx(), if_getnextiface(), svc_returnstruct::length, svc_returnstruct::list, snap_external_svclib_free_local_returnstruct(), svc_return, SVC_SNMP_TYPE_ADDR, and svc_returnitem::type.

Referenced by snap_external_svclib_getnextfunc().

5.79.1.4 int if_getiface (const char * *strIfName*, int *flag*)

Definition at line 225 of file snap_svc_if.c.

References d_printf().

Referenced by if_getiface_up(), and snap_external_svclib_getnextfunc().

5.79.1.5 int if_getiface_up (const char * *strIfName*)

Definition at line 242 of file snap_svc_if.c.

References if_getiface().

Referenced by if_setiface(), and snap_external_svclib_getnextfunc().

5.79.1.6 uint32_t if_getnexthop (uint32_t *incoming_ip*)

Definition at line 392 of file snap_svc_if.c.

References if_gethopfromiface(), and if_getoutiface().

Referenced by snap_external_svclib_getnextfunc().

5.79.1.7 int if_setiface (const char * *strIfName*, int *flag*, int *state*)

Definition at line 248 of file snap_svc_if.c.

References d_printf(), and if_getiface_up().

Referenced by if_setiface_up(), and snap_external_svclib_getnextfunc().

5.79.1.8 int if_setiface_up (const char * *strIfName*, int *state*)

Definition at line 273 of file snap_svc_if.c.

References if_setiface().

Referenced by snap_external_svclib_getnextfunc().

5.79.1.9 int snap_svc_ifip_init ()

Definition at line 109 of file snap_svc.if.c.

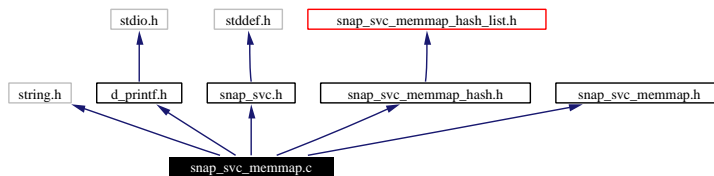
References `d_printf()`, `snap_svc_ifip_item::if_index`, `snap_svc_if_count`, and `snap_svc_if_maxidx`.

Referenced by `if_getoutiface()`, and `snap_external_svclib_init()`.

5.80 snap_svc/snap_svc_memmap.c File Reference

```
#include <string.h>
#include "d_printf.h"
#include "snap_svc.h"
#include "snap_svc_memmap_hash.h"
#include "snap_svc_memmap.h"
```

Include dependency graph for snap_svc_memmap.c:



Functions

- int **mystrcmp** (char *s1, char *s2)
- void **snap_external_svclib_init** ()
- void **snap_external_svclib_done** ()
- void **snap_external_svclib_getnextfunc** (char **snapsvc_name, snapsvc_func_proto *snapsvc_func, int *snapsvc_args, int *snapsvc_rets)
- void **snap_svc_memmap_add_value** (char *key, unsigned long pValue)
- void **snap_svc_memmap_add_string** (char *key, char *pValue)
- unsigned long **snap_svc_memmap_lookup_int** (char *key)
- void * **snap_svc_memmap_lookup_string** (char *key)
- void **snap_svc_memmap_del** (char *key)

Variables

- hash_table_t * **snap_svc_memmap_hashtable** = NULL

5.80.1 Function Documentation

5.80.1.1 int mystrcmp (char * s1, char * s2)

Definition at line 18 of file snap_svc_memmap.c.

Referenced by snap_external_svclib_init().

5.80.1.2 void snap_external_svclib_done ()

Definition at line 45 of file snap_svc_memmap.c.

References `d_printf()`.

**5.80.1.3 void snap_external_svclib_getnextfunc (char **
snapsvc_name, snapsvc_func_proto * snapsvc_func, int *
snapsvc_args, int * snapsvc_rets)**

Definition at line 51 of file snap_svc_memmap.c.

References `snap_svc_memmap_add_string()`, `snap_svc_memmap_add_value()`, `snap_svc_memmap_del()`, `snap_svc_memmap_lookup_int()`, `snap_svc_memmap_lookup_string()`, `snapsvc_func_proto`, `svc_fun_counter`, `SVC_SNMP_TYPE_INT`, `SVC_SNMP_TYPE_NULL`, and `SVC_SNMP_TYPE_STRING`.

5.80.1.4 void snap_external_svclib_init ()

Definition at line 35 of file snap_svc_memmap.c.

References `d_printf()`, `ht_create()`, and `mystrcmp()`.

**5.80.1.5 void snap_svc_memmap_add_string (char * key, char *
p Value)**

Definition at line 97 of file snap_svc_memmap.c.

References `ht_insert()`.

Referenced by `snap_external_svclib_getnextfunc()`.

**5.80.1.6 void snap_svc_memmap_add_value (char * key, unsigned
long p Value)**

Definition at line 91 of file snap_svc_memmap.c.

References `ht_insert()`.

Referenced by `snap_external_svclib_getnextfunc()`.

5.80.1.7 void snap_svc_memmap_del (char * key)

Definition at line 118 of file snap_svc_memmap.c.

References `ht_remove()`.

Referenced by `snap_external_svclib_getnextfunc()`.

5.80.1.8 unsigned long snap_svc_memmap_lookup_int (char * *key*)

Definition at line 106 of file snap_svc_memmap.c.

References ht_lookup().

Referenced by snap_external_svclib_getnextfunc().

5.80.1.9 void* snap_svc_memmap_lookup_string (char * *key*)

Definition at line 114 of file snap_svc_memmap.c.

References ht_lookup().

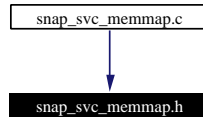
Referenced by snap_external_svclib_getnextfunc().

5.80.2 Variable Documentation**5.80.2.1 hash_table_t* snap_svc_memmap_hashtable = NULL**

Definition at line 15 of file snap_svc_memmap.c.

5.81 snap_svc/snap_svc_memmap.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void **snap_svc_memmap_add_value** (char *key, unsigned long uValue)
- void **snap_svc_memmap_add_string** (char *key, char *pValue)
- unsigned long **snap_svc_memmap_lookup_int** (char *key)
- void * **snap_svc_memmap_lookup_string** (char *key)
- void **snap_svc_memmap_del** (char *key)

5.81.1 Function Documentation

5.81.1.1 void snap_svc_memmap_add_string (char * *key*, char * *pValue*)

Definition at line 97 of file snap_svc_memmap.c.

References ht_insert().

Referenced by snap_external_svclib_getnextfunc().

5.81.1.2 void snap_svc_memmap_add_value (char * *key*, unsigned long *uValue*)

Definition at line 91 of file snap_svc_memmap.c.

References ht_insert().

Referenced by snap_external_svclib_getnextfunc().

5.81.1.3 void snap_svc_memmap_del (char * *key*)

Definition at line 118 of file snap_svc_memmap.c.

References ht_remove().

Referenced by snap_external_svclib_getnextfunc().

5.81.1.4 unsigned long snap_svc_memmap_lookup_int (char * *key*)

Definition at line 106 of file snap_svc_memmap.c.

References ht_lookup().

Referenced by snap_external_svclib_getnextfunc().

5.81.1.5 void* snap_svc_memmap_lookup_string (char * *key*)

Definition at line 114 of file snap_svc_memmap.c.

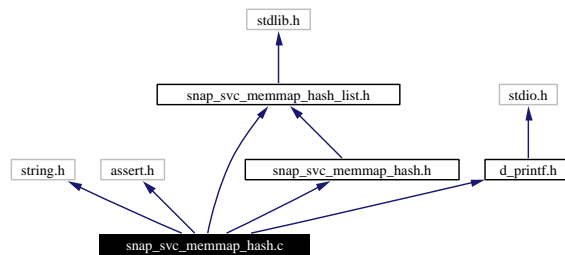
References ht_lookup().

Referenced by snap_external_svclib_getnextfunc().

5.82 snap_svc/snap_svc_memmap_hash.c File Reference

```
#include <string.h>
#include <assert.h>
#include "snap_svc_memmap_hash_list.h"
#include "snap_svc_memmap_hash.h"
#include "d_printf.h"
```

Include dependency graph for snap_svc_memmap_hash.c:



Functions

- int **hash_string** (char *s)
- **hash_table_t** * **ht_create** (int sz, int(*cmp)(const void *, const void *), int(*hash)(void *))
- void **ht_insert** (**hash_table_t** *t, void *key, void *val)
- void * **ht_lookup** (**hash_table_t** *t, void *key)
- void **ht_remove** (**hash_table_t** *t, void *key)

Variables

- int **ht_errno** = 0

5.82.1 Function Documentation

5.82.1.1 int hash_string (char * s)

Definition at line 17 of file snap_svc_memmap_hash.c.

5.82.1.2 hash_table_t* ht_create (int sz, int(* cmp)(const void *, const void *), int(* hash)(void *))

Definition at line 37 of file snap_svc_memmap_hash.c.

References `hash_table_t::cmp`, `hash_table_t::hash`, `hash_table_t::max_len`, `memalloc`, `hash_table_t::tab`, and `hash_table_t::tab_sz`.

Referenced by `snap_external_svclib_init()`, and `snap_svc_table_init()`.

5.82.1.3 void ht_insert (hash_table_t * t, void * key, void * val)

Definition at line 69 of file `snap_svc_memmap_hash.c`.

References `cons()`, `d_printf()`, `hash_table_t::hash`, `pair_t::key`, `length_list()`, `hash_table_t::max_len`, `memalloc`, `hash_table_t::tab`, `hash_table_t::tab_sz`, and `pair_t::value`.

Referenced by `snap_svc_memmap_add_string()`, `snap_svc_memmap_add_value()`, and `snap_svc_table_add()`.

5.82.1.4 void* ht_lookup (hash_table_t * t, void * key)

Definition at line 112 of file `snap_svc_memmap_hash.c`.

References `hash_table_t::cmp`, `d_printf()`, `hash_table_t::hash`, `hash_table_t::tab`, and `hash_table_t::tab_sz`.

Referenced by `snap_svc_memmap_lookup_int()`, `snap_svc_memmap_lookup_string()`, and `snap_svc_table_find()`.

5.82.1.5 void ht_remove (hash_table_t * t, void * key)

Definition at line 125 of file `snap_svc_memmap_hash.c`.

References `hash_table_t::cmp`, `hash_table_t::hash`, `pair_t::key`, `l::next`, `hash_table_t::tab`, `hash_table_t::tab_sz`, and `l::v`.

Referenced by `snap_svc_memmap_del()`.

5.82.2 Variable Documentation

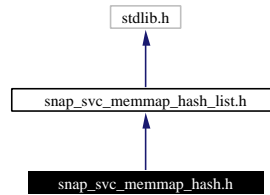
5.82.2.1 int ht_errno = 0

Definition at line 34 of file `snap_svc_memmap_hash.c`.

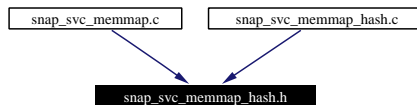
5.83 snap_svc/snap_svc_memmap_hash.h File Reference

```
#include "snap_svc_memmap_hash_list.h"
```

Include dependency graph for snap_svc_memmap_hash.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct `hash_table_t`
- struct `pair_t`

Functions

- `hash_table_t * ht_create` (int sz, int(*cmp)(const void *, const void *), int(*hash)(void *))
- void `ht_insert` (`hash_table_t` *t, void *key, void *val)
- void * `ht_lookup` (`hash_table_t` *t, void *key)
- void `ht_remove` (`hash_table_t` *t, void *key)
- int `hash_string` (char *s)

Variables

- int `ht_errno`

5.83.1 Function Documentation

5.83.1.1 int hash_string (char * s)

Definition at line 25 of file snap_hashtable.c.

5.83.1.2 `hash_table_t* ht_create (int sz, int(* cmp)(const void *, const void *), int(* hash)(void *))`

Definition at line 46 of file snap_hashtable.c.

References `hash_table_t::cmp`, `hash_table_t::hash`, `hash_table_t::max_len`, `memalloc`, `hash_table_t::tab`, and `hash_table_t::tab_sz`.

Referenced by `snap_external_svclib_init()`, and `snap_svc_table_init()`.

5.83.1.3 `void ht_insert (hash_table_t * t, void * key, void * val)`

Definition at line 78 of file snap_hashtable.c.

References `cons()`, `d_printf()`, `hash_table_t::hash`, `pair_t::key`, `length_list()`, `hash_table_t::max_len`, `memalloc`, `hash_table_t::tab`, `hash_table_t::tab_sz`, and `pair_t::value`.

Referenced by `snap_svc_memmap_add_string()`, `snap_svc_memmap_add_value()`, and `snap_svc_table_add()`.

5.83.1.4 `void* ht_lookup (hash_table_t * t, void * key)`

Definition at line 121 of file snap_hashtable.c.

References `hash_table_t::cmp`, `d_printf()`, `hash_table_t::hash`, `hash_table_t::tab`, and `hash_table_t::tab_sz`.

Referenced by `snap_svc_memmap_lookup_int()`, `snap_svc_memmap_lookup_string()`, and `snap_svc_table_find()`.

5.83.1.5 `void ht_remove (hash_table_t * t, void * key)`

Definition at line 134 of file snap_hashtable.c.

References `hash_table_t::cmp`, `hash_table_t::hash`, `pair_t::key`, `l::next`, `hash_table_t::tab`, `hash_table_t::tab_sz`, and `l::v`.

Referenced by `snap_svc_memmap_del()`.

5.83.2 Variable Documentation

5.83.2.1 `int ht_errno`

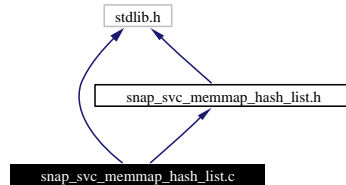
Definition at line 24 of file snap_svc_memmap_hash.h.

5.84 snap_svc/snap_svc_memmap_hash_list.c File Reference

```
#include <stdlib.h>
```

```
#include "snap_svc_memmap_hash_list.h"
```

Include dependency graph for snap_svc_memmap_hash_list.c:



Functions

- `list_t * cons` (void *v, list_t *next)
- void `free_list` (list_t *list)
- int `length_list` (list_t *list)

5.84.1 Function Documentation

5.84.1.1 list_t* `cons` (void * v, list_t * next)

Definition at line 19 of file snap_svc_memmap_hash_list.c.

References memalloc, l::next, and l::v.

Referenced by ht_insert(), and yyparse().

5.84.1.2 void `free_list` (list_t * list)

Definition at line 36 of file snap_svc_memmap_hash_list.c.

References l::next.

Referenced by yyparse().

5.84.1.3 int `length_list` (list_t * list)

Definition at line 49 of file snap_svc_memmap_hash_list.c.

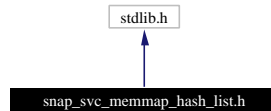
References l::next.

Referenced by ht_insert(), and newtup().

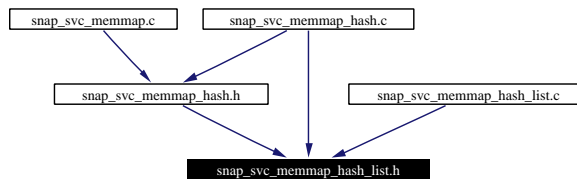
5.85 snap_svc/snap_svc_memmap_hash_list.h File Reference

```
#include <stdlib.h>
```

Include dependency graph for snap_svc_memmap_hash_list.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct `l`

Defines

- #define `memalloc(ptr, t, sz)`

Typedefs

- typedef `l` `list_t`

Functions

- `list_t * cons` (void *v, `list_t` *next)
- void `free_list` (`list_t` *list)
- int `length_list` (`list_t` *list)

5.85.1 Define Documentation

5.85.1.1 #define memalloc(ptr, t, sz)

Value:

```
{ void *_result;                                \
    _result = (void *)malloc(sz);              \|
    (ptr) = (t)_result;                         \|
}
```

Definition at line 12 of file snap_svc_memmap_hash_list.h.

5.85.2 Typedef Documentation

5.85.2.1 typedef struct l list_t

5.85.3 Function Documentation

5.85.3.1 list_t* cons (void * v, list_t * next)

Definition at line 24 of file snap_list.c.

References memalloc, l::next, and l::v.

Referenced by ht_insert(), and yyparse().

5.85.3.2 void free_list (list_t * list)

Definition at line 41 of file snap_list.c.

References l::next.

Referenced by yyparse().

5.85.3.3 int length_list (list_t * list)

Definition at line 54 of file snap_list.c.

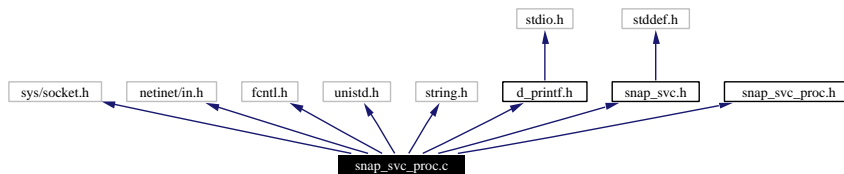
References l::next.

Referenced by ht_insert(), and newtup().

5.86 snap_svc/snap_svc_proc.c File Reference

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>
#include "d_printf.h"
#include "snap_svc.h"
#include "snap_svc_proc.h"
```

Include dependency graph for snap_svc_proc.c:



Functions

- void **snap_external_svclib_init** ()
- void **snap_external_svclib_done** ()
- void **snap_external_svclib_getnextfunc** (char **snapsvc_name, snapsvc_func_proto *snapsvc_func, int *snapsvc_args, int *snapsvc_rets)
- int **proc_sysnetip_getforwarding** (void *useless)
- int **proc_sysnetip_setforwarding** (unsigned int uValue)

5.86.1 Function Documentation

5.86.1.1 int proc_sysnetip_getforwarding (void * *useless*)

Definition at line 72 of file snap_svc_proc.c.

References d_printf().

Referenced by snap_external_svclib_getnextfunc().

5.86.1.2 int proc_sysnetip_setforwarding (unsigned int *uValue*)

Definition at line 94 of file snap_svc_proc.c.

References d_printf().

Referenced by `snap_external_svclib_getnextfunc()`.

5.86.1.3 `void snap_external_svclib_done ()`

Definition at line 27 of file `snap_svc_proc.c`.

References `d_printf()`.

5.86.1.4 `void snap_external_svclib_getnextfunc (char ** snapsvc_name, snapsvc_func_proto * snapsvc_func, int * snapsvc_args, int * snapsvc_rets)`

Definition at line 32 of file `snap_svc_proc.c`.

References `proc_sysnetip_getforwarding()`, `proc_sysnetip_setforwarding()`, `snapsvc_func_proto`, `svc_fun_counter`, `SVC_SNMP_TYPE_INT`, and `SVC_SNMP_TYPE_NULL`.

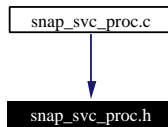
5.86.1.5 `void snap_external_svclib_init ()`

Definition at line 23 of file `snap_svc_proc.c`.

References `d_printf()`.

5.87 snap_svc/snap_svc_proc.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int **proc_sysnetip_getforwarding** (void *useless)
- int **proc_sysnetip_setforwarding** (unsigned int uValue)

5.87.1 Function Documentation

5.87.1.1 int proc_sysnetip_getforwarding (void * *useless*)

Definition at line 72 of file snap_svc_proc.c.

References d_printf().

Referenced by snap_external_svclib_getnextfunc().

5.87.1.2 int proc_sysnetip_setforwarding (unsigned int *uValue*)

Definition at line 94 of file snap_svc_proc.c.

References d_printf().

Referenced by snap_external_svclib_getnextfunc().

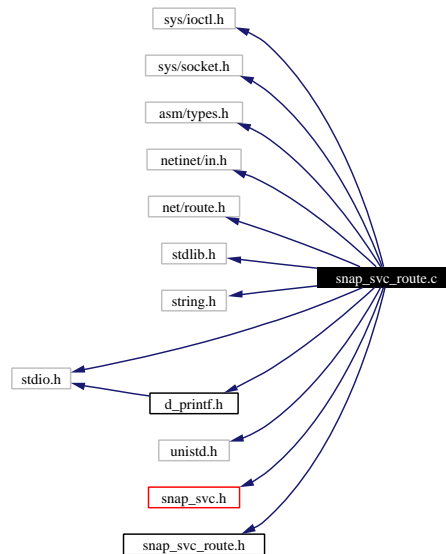
5.88 snap_svc/snap_svc_route.c File Reference

```

#include <sys/ioctl.h>
#include <sys/socket.h>
#include <asm/types.h>
#include <netinet/in.h>
#include <net/route.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <unistd.h>
#include "d_printf.h"
#include "snap_svc.h"
#include "snap_svc_route.h"

```

Include dependency graph for snap_svc_route.c:



Functions

- void **snap_external_svclib_init** ()
- void **snap_external_svclib_done** ()
- void **snap_external_svclib_getnextfunc** (char **snapsvc_name, **snapsvc_func_proto** *snapsvc_func, int *snapsvc_args, int *snapsvc_rets)
- **_u32 ip_to_u32** (_u8 ip1, _u8 ip2, _u8 ip3, _u8 ip4)

- void **printip** (`__u32 ip`, `int dLevel`)
- void **make_sockaddr_in** (`__u32 ip_in`, `struct sockaddr *outaddr`)
- int **handle_request** (`short action`, `__u32 ip_dest`, `__u32 ip_mask`, `__u32 ip_gateway`, `char *device`)
- int **route_add** (`__u32 ip_dest`, `__u32 ip_mask`, `__u32 ip_gateway`, `char *device`)
- int **route_del** (`__u32 ip_dest`, `__u32 ip_mask`, `__u32 ip_gateway`, `char *device`)

5.88.1 Function Documentation

5.88.1.1 int handle_request (short *action*, __u32 *ip_dest*, __u32 *ip_mask*, __u32 *ip_gateway*, char * *device*)

Definition at line 84 of file snap_svc_route.c.

References `d_printf()`, `make_sockaddr_in()`, `printip()`, `rtenry::rt_dev`, `rtenry::rt_dst`, `rtenry::rt_flags`, `rtenry::rt_gateway`, and `rtenry::rt_genmask`.

Referenced by `route_add()`, and `route_del()`.

5.88.1.2 __u32 ip_to__u32 (__u8 *ip1*, __u8 *ip2*, __u8 *ip3*, __u8 *ip4*)

Definition at line 61 of file snap_svc_route.c.

5.88.1.3 void make_sockaddr_in (__u32 *ip_in*, struct sockaddr * *outaddr*)

Definition at line 70 of file snap_svc_route.c.

Referenced by `handle_request()`.

5.88.1.4 void printip (__u32 *ip*, int *dLevel*)

Definition at line 66 of file snap_svc_route.c.

References `d_printf()`.

Referenced by `handle_request()`.

5.88.1.5 int route_add (__u32 *ip_dest*, __u32 *ip_mask*, __u32 *ip_gateway*, char * *device*)

Definition at line 146 of file snap_svc_route.c.

References `handle_request()`.

Referenced by `snap_external_svclib_getnextfunc()`.

5.88.1.6 `int route_del (_u32 ip_dest, _u32 ip_mask, _u32 ip_gateway, char * device)`

Definition at line 150 of file `snap_svc_route.c`.

References `handle_request()`.

Referenced by `snap_external_svclib_getnextfunc()`.

5.88.1.7 `void snap_external_svclib_done ()`

Definition at line 30 of file `snap_svc_route.c`.

References `d_printf()`.

5.88.1.8 `void snap_external_svclib_getnextfunc (char ** snapsvc_name, snapsvc_func_proto * snapsvc_func, int * snapsvc_args, int * snapsvc_rets)`

Definition at line 35 of file `snap_svc_route.c`.

References `route_add()`, `route_del()`, `snapsvc_func_proto`, `svc_fun_counter`, and `SVC_SNMP_TYPE_NULL`.

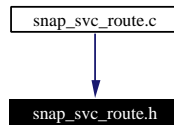
5.88.1.9 `void snap_external_svclib_init ()`

Definition at line 26 of file `snap_svc_route.c`.

References `d_printf()`.

5.89 snap_svc/snap_svc_route.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int **route_add** (u32 ip_dest, u32 ip_mask, u32 ip_gateway, char *device)
- int **route_del** (u32 ip_dest, u32 ip_mask, u32 ip_gateway, char *device)

5.89.1 Function Documentation

5.89.1.1 int route_add (u32 ip_dest, u32 ip_mask, u32 ip_gateway, char * device)

Definition at line 146 of file snap_svc_route.c.

References handle_request().

Referenced by snap_external_svclib_getnextfunc().

5.89.1.2 int route_del (u32 ip_dest, u32 ip_mask, u32 ip_gateway, char * device)

Definition at line 150 of file snap_svc_route.c.

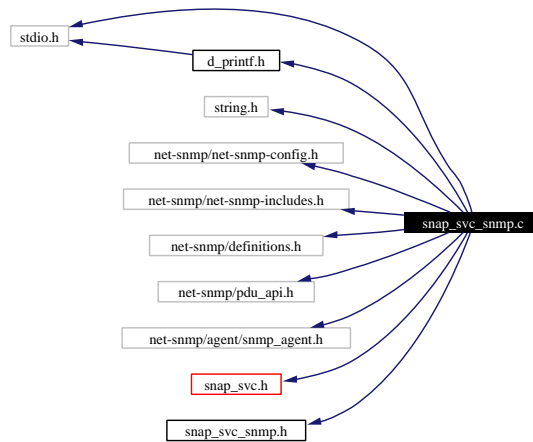
References handle_request().

Referenced by snap_external_svclib_getnextfunc().

5.90 snap_svc/snap_svc_snmp.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <net-snmp/net-snmp-config.h>
#include <net-snmp/net-snmp-includes.h>
#include <net-snmp/definitions.h>
#include <net-snmp/pdu_api.h>
#include <net-snmp/agent/snmp_agent.h>
#include "snap_svc.h"
#include "snap_svc_snmp.h"
#include "d_printf.h"
```

Include dependency graph for snap_svc_snmp.c:



Functions

- void **snap_external_svclib_init** ()
- void **snap_external_svclib_done** ()
- void **snap_external_svclib_getnextfunc** (char **snapsvc_name, snapsvc_func_proto *snapsvc_func, int *snapsvc_args, int *snapsvc_rets)
- int **snap_external_svclib_snmp_INTERNAL_execpdu_handler** (int op, netsnmp_session *session, int reqid, netsnmp_pdu *pdu, void *useless)
- int **snap_external_svclib_snmp_init** (void *useless)
- int **snap_external_svclib_snmp_init_ip** (unsigned int ip)
- int **snap_external_svclib_snmp_initpdu** (unsigned int msg_type)
- int **snap_external_svclib_snmp_addvar_null** (char *name)

- int **snap_external_svclib_snmp_addvar_withvalue** (char *name, int type, void *value, int valuelen)
- int **snap_external_svclib_snmp_execpdu** (void *useless)
- int **snap_external_svclib_snmp_close** (void *useless)
- int **snap_external_svclib_snmp_getsingle** (char *name)
- int **snap_external_svclib_snmp_setsingle** (char *name, int type, void *value, int valuelen)
- int **snap_external_svclib_snmp_getiface** (unsigned long in_ip)
- unsigned long **snap_external_svclib_snmp_gethop** (int iface)
- int **snap_external_svclib_snmp_getifnumber** (void *useless)
- int **snap_external_svclib_snmp_isupiface** (int iface)
- int **snap_external_svclib_snmp_getnextiface** (int old_iface)
- int **snap_external_svclib_snmp_getnextifacefromip** (unsigned long in_ip)
- unsigned long **snap_external_svclib_snmp_getnexthopfromip** (unsigned long in_ip)
- unsigned long **snap_external_svclib_snmp_getnexthopfromip** (unsigned long in_ip)
- int **snap_external_svclib_snmp_getallotherneighboursfromip** (unsigned long in_ip)

Variables

- int **callback_master_num**
- unsigned short **svc_snmp_active** = 0
- unsigned short **svc_snmp_pdu_load** = 0
- netsnmp_pdu * **svc_snmp_pdu** = NULL
- netsnmp_session * **svc_snmp_session** = NULL
- int **svc_snmp_pdu_waiting** = 0

5.90.1 Function Documentation

5.90.1.1 void snap_external_svclib_done ()

Definition at line 35 of file snap_svc_snmp.c.

References `d_printf()`, `snap_external_svclib_snmp_close()`, `svc_snmp_active`, `svc_snmp_pdu`, `svc_snmp_pdu_load`, `svc_snmp_pdu_waiting`, and `svc_snmp_session`.

5.90.1.2 void snap_external_svclib_getnextfunc (char ** *snapsvc_name*, snapsvc_func_proto * *snapsvc_func*, int * *snapsvc_args*, int * *snapsvc_rets*)

Definition at line 53 of file snap_svc_snmp.c.

References `snap_external_svclib_snmp_addvar_null()`, `snap_external_svclib_snmp_addvar_withvalue()`, `snap_external_svclib_snmp_close()`,

[snap_external_svclib_snmp_exeCPdu\(\)](#), [snap_external_svclib_snmp_getallotherneighboursfromip\(\)](#), [snap_external_svclib_snmp_getnexthopfromip\(\)](#), [snap_external_svclib_snmp_getsingle\(\)](#), [snap_external_svclib_snmp_init\(\)](#), [snap_external_svclib_snmp_init_ip\(\)](#), [snap_external_svclib_snmp_initpdu\(\)](#), [snap_external_svclib_snmp_setsingle\(\)](#), [snapsvc_func_proto](#), [svc_fun_counter](#), and [SVC_SNMP_TYPE_NULL](#).

5.90.1.3 void snap_external_svclib_init ()

Definition at line 30 of file `snap_svc_snmp.c`.

References [d_printf\(\)](#), and [snap_external_svclib_snmp_init\(\)](#).

5.90.1.4 int snap_external_svclib_snmp_addvar_null (char * name)

Definition at line 304 of file `snap_svc_snmp.c`.

References [d_printf\(\)](#), [snap_external_svclib_snmp_initpdu\(\)](#), and [svc_snmp_pdu](#).

Referenced by [snap_external_svclib_getnextfunc\(\)](#), [snap_external_svclib_snmp_gethop\(\)](#), and [snap_external_svclib_snmp_getsingle\(\)](#).

5.90.1.5 int snap_external_svclib_snmp_addvar_withvalue (char * name, int type, void * value, int valuelen)

Definition at line 328 of file `snap_svc_snmp.c`.

References [d_printf\(\)](#), [snap_external_svclib_snmp_initpdu\(\)](#), and [svc_snmp_pdu](#).

Referenced by [snap_external_svclib_getnextfunc\(\)](#), and [snap_external_svclib_snmp_setsingle\(\)](#).

5.90.1.6 int snap_external_svclib_snmp_close (void * useless)

Definition at line 401 of file `snap_svc_snmp.c`.

References [d_printf\(\)](#), [svc_snmp_active](#), [svc_snmp_pdu](#), and [svc_snmp_session](#).

Referenced by [snap_external_svclib_done\(\)](#), [snap_external_svclib_getnextfunc\(\)](#), [snap_external_svclib_snmp_init\(\)](#), and [snap_external_svclib_snmp_init_ip\(\)](#).

5.90.1.7 int snap_external_svclib_snmp_exeCPdu (void * useless)

Definition at line 351 of file `snap_svc_snmp.c`.

References [d_printf\(\)](#), [d_printf_timed\(\)](#), [fdset](#), [svc_snmp_pdu](#), [svc_snmp_pdu_load](#), [svc_snmp_pdu_waiting](#), and [svc_snmp_session](#).

Referenced by [snap_external_svclib_getnextfunc\(\)](#), [snap_external_svclib_snmp_gethop\(\)](#), [snap_external_svclib_snmp_getsingle\(\)](#), and [snap_external_svclib_snmp_setsingle\(\)](#).

5.90.1.8 int snap_external_svclib_snmp_getallotherneighboursfromip (unsigned long *in_ip*)

Definition at line 631 of file snap_svc_snmp.c.

References `d_printf()`, `svc_returnitem::data`, `svc_returnstruct::length`, `svc_returnitem::length`, `svc_returnstruct::list`, `snap_external_svclib_free_local_returnstruct()`, `snap_external_svclib_snmp_gethop()`, `snap_external_svclib_snmp_getifnumber()`, `snap_external_svclib_snmp_isupiface()`, `svc_return`, `SVC_SNMP_TYPE_ADDR`, `SVC_SNMP_TYPE_INT`, `SVC_SNMP_TYPE_NULL`, and `svc_returnitem::type`.

Referenced by `snap_external_svclib_getnextfunc()`.

5.90.1.9 unsigned long snap_external_svclib_snmp_gethop (int *iface*)

Definition at line 476 of file snap_svc_snmp.c.

References `d_printf()`, `svc_returnitem::data`, `svc_returnstruct::length`, `svc_returnitem::length`, `svc_returnstruct::list`, `svc_returnitem::oid`, `svc_returnitem::oid_length`, `snap_external_svclib_snmp_addvar_null()`, `snap_external_svclib_snmp_execepdu()`, `snap_external_svclib_snmp_initpdu()`, `svc_return`, `svc_snmp_pdu`, `SVC_SNMP_TYPE_INT`, and `svc_returnitem::type`.

Referenced by `snap_external_svclib_snmp_getallotherneighboursfromip()`, and `snap_internal_svclib_snmp_getnethopfromip()`.

5.90.1.10 int snap_external_svclib_snmp_getiface (unsigned long *in_ip*)

Definition at line 455 of file snap_svc_snmp.c.

References `d_printf()`, `svc_returnitem::data`, `svc_returnstruct::length`, `svc_returnitem::length`, `svc_returnstruct::list`, `snap_external_svclib_snmp_getsingle()`, `svc_return`, `SVC_SNMP_TYPE_INT`, and `svc_returnitem::type`.

Referenced by `snap_external_svclib_snmp_getnextifacefromip()`.

5.90.1.11 int snap_external_svclib_snmp_getifnumber (void * *useless*)

Definition at line 540 of file snap_svc_snmp.c.

References `d_printf()`, `svc_returnitem::data`, `svc_returnstruct::length`, `svc_returnitem::length`, `svc_returnstruct::list`, `snap_external_svclib_snmp_getsingle()`, `svc_return`, `SVC_SNMP_TYPE_INT`, and `svc_returnitem::type`.

Referenced by `snap_external_svclib_snmp_getallotherneighboursfromip()`, and `snap_external_svclib_snmp_getnextiface()`.

5.90.1.12 unsigned long snap_external_svclib_snmplib_getnexthopfromip (unsigned long *in_ip*)

Definition at line 617 of file snap_svc_snmp.c.

References svc_returnitem::data, svc_returnstruct::length, svc_returnstruct::list, snap_external_svclib_free_local_returnstruct(), snap_internal_svclib_snmplib_getnexthopfromip(), svc_return, SVC_SNMP_TYPE_ADDR, and svc_returnitem::type.

Referenced by snap_external_svclib_getnextfunc().

5.90.1.13 int snap_external_svclib_snmplib_getnextiface (int *old_iface*)

Definition at line 577 of file snap_svc_snmp.c.

References snap_external_svclib_snmplib_getifnumber(), and snap_external_svclib_snmplib_isupiface().

Referenced by snap_external_svclib_snmplib_getnextifacefromip().

5.90.1.14 int snap_external_svclib_snmplib_getnextifacefromip (unsigned long *in_ip*)

Definition at line 599 of file snap_svc_snmp.c.

References snap_external_svclib_snmplib_getiface(), and snap_external_svclib_snmplib_getnextiface().

Referenced by snap_internal_svclib_snmplib_getnexthopfromip().

5.90.1.15 int snap_external_svclib_snmplib_getsingle (char * *name*)

Definition at line 428 of file snap_svc_snmp.c.

References snap_external_svclib_snmplib_addvar_null(), snap_external_svclib_snmplib_execpdu(), and snap_external_svclib_snmplib_initpdu().

Referenced by snap_external_svclib_getnextfunc(), snap_external_svclib_snmplib_getiface(), snap_external_svclib_snmplib_getifnumber(), and snap_external_svclib_snmplib_isupiface().

5.90.1.16 int snap_external_svclib_snmplib_init (void * *useless*)

Definition at line 221 of file snap_svc_snmp.c.

References callback_master_num, d_printf(), snap_external_svclib_snmplib_close(), snap_external_svclib_snmplib_INTERNAL_execpdu_handler(), svc_snmp_active, and svc_snmp_session.

Referenced by `snap_external_svclib_getnextfunc()`, `snap_external_svclib_init()`, and `snap_external_svclib_snmp_initpdu()`.

5.90.1.17 `int snap_external_svclib_snmp_init_ip (unsigned int ip)`

Definition at line 240 of file `snap_svc_snmp.c`.

References `d_printf()`, `snap_external_svclib_snmp_close()`, `snap_external_svclib_snmp_INTERNAL_execpdu_handler()`, `svc_snmp_active`, and `svc_snmp_session`.

Referenced by `snap_external_svclib_getnextfunc()`.

5.90.1.18 `int snap_external_svclib_snmp_initpdu (unsigned int msg_type)`

Definition at line 271 of file `snap_svc_snmp.c`.

References `d_printf()`, `snap_external_svclib_snmp_init()`, `svc_snmp_pdu`, `svc_snmp_pdu_load`, and `svc_snmp_session`.

Referenced by `snap_external_svclib_getnextfunc()`, `snap_external_svclib_snmp_addvar_null()`, `snap_external_svclib_snmp_addvar_withvalue()`, `snap_external_svclib_snmp_gethop()`, `snap_external_svclib_snmp_getsingle()`, and `snap_external_svclib_snmp_setsingle()`.

5.90.1.19 `int snap_external_svclib_snmp_INTERNAL_execpdu_handler (int op, netsnmp_session * session, int reqid, netsnmp_pdu * pdu, void * useless)`

Definition at line 121 of file `snap_svc_snmp.c`.

References `d_printf()`, `svc_returnitem::data`, `svc_returnitem::length`, `svc_returnstruct::length`, `svc_returnstruct::list`, `svc_returnitem::oid`, `svc_returnitem::oid_length`, `snap_external_svclib_free_local_returnstruct()`, `svc_return`, `svc_snmp_pdu_waiting`, `SVC_SNMP_TYPE_ADDR`, `SVC_SNMP_TYPE_INT`, `SVC_SNMP_TYPE_LONG`, `SVC_SNMP_TYPE_NULL`, `SVC_SNMP_TYPE_STRING`, and `svc_returnitem::type`.

Referenced by `snap_external_svclib_snmp_init()`, and `snap_external_svclib_snmp_init_ip()`.

5.90.1.20 `int snap_external_svclib_snmp_isupiface (int iface)`

Definition at line 557 of file `snap_svc_snmp.c`.

References `d_printf()`, `svc_returnitem::data`, `svc_returnstruct::length`, `svc_returnitem::length`, `svc_returnstruct::list`, `snap_external_svclib_snmp_getsingle()`, `svc_return`, `SVC_SNMP_TYPE_INT`, and `svc_returnitem::type`.

Referenced by `snap_external_svclib_snmp_getallotherneighboursfromip()`, and `snap_external_svclib_snmp_getnextiface()`.

5.90.1.21 `int snap_external_svclib_snmp_setsingle (char * name, int type, void * value, int valuelen)`

Definition at line 439 of file `snap_svc_snmp.c`.

References `snap_external_svclib_snmp_addvar_withvalue()`, `snap_external_svclib_snmp_execpdu()`, and `snap_external_svclib_snmp_initpdu()`.

Referenced by `snap_external_svclib_getnextfunc()`.

5.90.1.22 `unsigned long snap_internal_svclib_snmp_getnexthopfromip (unsigned long in_ip)`

Definition at line 606 of file `snap_svc_snmp.c`.

References `snap_external_svclib_snmp_gethop()`, and `snap_external_svclib_snmp_getnextifacefromip()`.

Referenced by `snap_external_svclib_snmp_getnexthopfromip()`.

5.90.2 Variable Documentation

5.90.2.1 `int callback_master_num`

Definition at line 20 of file `snap_svc_snmp.c`.

Referenced by `snap_external_svclib_snmp_init()`.

5.90.2.2 `unsigned short svc_snmp_active = 0`

Definition at line 23 of file `snap_svc_snmp.c`.

Referenced by `snap_external_svclib_done()`, `snap_external_svclib_snmp_close()`, `snap_external_svclib_snmp_init()`, and `snap_external_svclib_snmp_init_ip()`.

5.90.2.3 `netsnmp_pdu* svc_snmp_pdu = NULL`

Definition at line 25 of file `snap_svc_snmp.c`.

Referenced by `snap_external_svclib_done()`, `snap_external_svclib_snmp_addvar_null()`, `snap_external_svclib_snmp_addvar_withvalue()`, `snap_external_svclib_snmp_close()`, `snap_external_svclib_snmp_execpdu()`, `snap_external_svclib_snmp_gethop()`, and `snap_external_svclib_snmp_initpdu()`.

5.90.2.4 `unsigned short svc_snmp_pdu_load = 0`

Definition at line 24 of file `snap_svc_snmp.c`.

Referenced by `snap_external_svclib_done()`, `snap_external_svclib_snmp_execpdu()`, and `snap_external_svclib_snmp_initpdu()`.

5.90.2.5 int svc_snmp_pdu_waiting = 0

Definition at line 27 of file snap_svc_snmp.c.

Referenced by snap_external_svclib_done(), snap_external_svclib_snmp_execpdu(), and snap_external_svclib_snmp_INTERNAL_execpdu_handler().

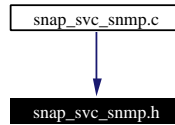
5.90.2.6 netsnmp_session* svc_snmp_session = NULL

Definition at line 26 of file snap_svc_snmp.c.

Referenced by snap_external_svclib_done(), snap_external_svclib_snmp_close(), snap_external_svclib_snmp_execpdu(), snap_external_svclib_snmp_init(), snap_external_svclib_snmp_init_ip(), and snap_external_svclib_snmp_initpdu().

5.91 snap_svc/snap_svc_snmp.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int **snap_external_svclib_snmp_init** (void *)
- int **snap_external_svclib_snmp_init_ip** (unsigned int)
- int **snap_external_svclib_snmp_initpdu** (unsigned int)
- int **snap_external_svclib_snmp_addvar_null** (char *)
- int **snap_external_svclib_snmp_addvar_withvalue** (char *, int, void *, int)
- int **snap_external_svclib_snmp_execpdu** (void *)
- int **snap_external_svclib_snmp_close** (void *)
- int **snap_external_svclib_snmp_getsingle** (char *)
- int **snap_external_svclib_snmp_setsingle** (char *, int, void *, int)
- int **snap_external_svclib_snmp_getiface** (unsigned long)
- unsigned long **snap_external_svclib_snmp_gethop** (int)
- int **snap_external_svclib_snmp_getifnumber** (void *)
- int **snap_external_svclib_snmp_isupiface** (int)
- int **snap_external_svclib_snmp_getnextiface** (int)
- int **snap_external_svclib_snmp_getnextifacefromip** (unsigned long)
- unsigned long **snap_external_svclib_snmp_getnexthopfromip** (unsigned long)
- int **snap_external_svclib_snmp_getallotherneighboursfromip** (unsigned long)

5.91.1 Function Documentation

5.91.1.1 int snap_external_svclib_snmp_addvar_null (char *)

Definition at line 304 of file snap_svc_snmp.c.

References `d_printf()`, `snap_external_svclib_snmp_initpdu()`, and `svc_snmp_pdu`.

Referenced by `snap_external_svclib_getnextfunc()`, `snap_external_svclib_snmp_gethop()`, and `snap_external_svclib_snmp_getsingle()`.

5.91.1.2 int snap_external_svclib_snmp_addvar_withvalue (char *, int, void *, int)

Definition at line 328 of file snap_svc_snmp.c.

References d_printf(), snap_external_svclib_snmp_initpdu(), and svc_snmp_pdu.

Referenced by snap_external_svclib_getnextfunc(), and snap_external_svclib_snmp_setsingle().

5.91.1.3 int snap_external_svclib_snmp_close (void *)

Definition at line 401 of file snap_svc_snmp.c.

References d_printf(), svc_snmp_active, svc_snmp_pdu, and svc_snmp_session.

Referenced by snap_external_svclib_done(), snap_external_svclib_getnextfunc(), snap_external_svclib_snmp_init(), and snap_external_svclib_snmp_init_ip().

5.91.1.4 int snap_external_svclib_snmp_execepdu (void *)

Definition at line 351 of file snap_svc_snmp.c.

References d_printf(), d_printf_timed(), fdset, svc_snmp_pdu, svc_snmp_pdu_load, svc_snmp_pdu_waiting, and svc_snmp_session.

Referenced by snap_external_svclib_getnextfunc(), snap_external_svclib_snmp_gethop(), snap_external_svclib_snmp_getsingle(), and snap_external_svclib_snmp_setsingle().

5.91.1.5 int snap_external_svclib_snmp_getallotherneighboursfromip (unsigned long)

Definition at line 631 of file snap_svc_snmp.c.

References d_printf(), svc_returnitem::data, svc_returnitem::length, svc_returnstruct::length, svc_returnstruct::list, snap_external_svclib_free_local_returnstruct(), snap_external_svclib_snmp_gethop(), snap_external_svclib_snmp_getifnumber(), snap_external_svclib_snmp_isupiface(), svc_return, SVC_SNMP_TYPE_ADDR, SVC_SNMP_TYPE_INT, SVC_SNMP_TYPE_NULL, and svc_returnitem::type.

Referenced by snap_external_svclib_getnextfunc().

5.91.1.6 unsigned long snap_external_svclib_snmp_gethop (int)

Definition at line 476 of file snap_svc_snmp.c.

References d_printf(), svc_returnitem::data, svc_returnitem::length, svc_returnstruct::length, svc_returnstruct::list, svc_returnitem::oid, svc_returnitem::oid_length, snap_external_svclib_snmp_addvar_null(), snap_

external_svclib_snmp_execeptdu(), snap_external_svclib_snmp_initpdu(), svc_return, svc_snmp_pdu, SVC_SNMP_TYPE_INT, and svc_returnitem::type.

Referenced by snap_external_svclib_snmp_getallotherneighboursfromip(), and snap_internal_svclib_snmp_getnexthopfromip().

5.91.1.7 int snap_external_svclib_snmp_getiface (unsigned long)

Definition at line 455 of file snap_svc_snmp.c.

References d_printf(), svc_returnitem::data, svc_returnitem::length, svc_returnstruct::length, svc_returnstruct::list, snap_external_svclib_snmp_getsingle(), svc_return, SVC_SNMP_TYPE_INT, and svc_returnitem::type.

Referenced by snap_external_svclib_snmp_getnextifacefromip().

5.91.1.8 int snap_external_svclib_snmp_getifnumber (void *)

Definition at line 540 of file snap_svc_snmp.c.

References d_printf(), svc_returnitem::data, svc_returnitem::length, svc_returnstruct::length, svc_returnstruct::list, snap_external_svclib_snmp_getsingle(), svc_return, SVC_SNMP_TYPE_INT, and svc_returnitem::type.

Referenced by snap_external_svclib_snmp_getallotherneighboursfromip(), and snap_external_svclib_snmp_getnextiface().

5.91.1.9 unsigned long snap_external_svclib_snmp_getnexthopfromip (unsigned long)

Definition at line 617 of file snap_svc_snmp.c.

References svc_returnitem::data, svc_returnstruct::length, svc_returnstruct::list, snap_external_svclib_free_local_returnstruct(), snap_internal_svclib_snmp_getnexthopfromip(), svc_return, SVC_SNMP_TYPE_ADDR, and svc_returnitem::type.

Referenced by snap_external_svclib_getnextfunc().

5.91.1.10 int snap_external_svclib_snmp_getnextiface (int)

Definition at line 577 of file snap_svc_snmp.c.

References snap_external_svclib_snmp_getifnumber(), and snap_external_svclib_snmp_isupiface().

Referenced by snap_external_svclib_snmp_getnextifacefromip().

5.91.1.11 int snap_external_svclib_snmp_getnextifacefromip (unsigned long)

Definition at line 599 of file snap_svc_snmp.c.

References `snap_external_svclib_snmp_getiface()`, and `snap_external_svclib_snmp_getnextiface()`.

Referenced by `snap_internal_svclib_snmp_getnexthopfromip()`.

5.91.1.12 int snap_external_svclib_snmp_getsingle (char *)

Definition at line 428 of file snap_svc_snmp.c.

References `snap_external_svclib_snmp_addvar_null()`, `snap_external_svclib_snmp_execpdu()`, and `snap_external_svclib_snmp_initpdu()`.

Referenced by `snap_external_svclib_getnextfunc()`, `snap_external_svclib_snmp_getiface()`, `snap_external_svclib_snmp_getifnumber()`, and `snap_external_svclib_snmp_isupiface()`.

5.91.1.13 int snap_external_svclib_snmp_init (void *)

Definition at line 221 of file snap_svc_snmp.c.

References `callback_master_num`, `d_printf()`, `snap_external_svclib_snmp_close()`, `snap_external_svclib_snmp_INTERNAL_execpdu_handler()`, `svc_snmp_active`, and `svc_snmp_session`.

Referenced by `snap_external_svclib_getnextfunc()`, `snap_external_svclib_init()`, and `snap_external_svclib_snmp_initpdu()`.

5.91.1.14 int snap_external_svclib_snmp_init_ip (unsigned int)

Definition at line 240 of file snap_svc_snmp.c.

References `d_printf()`, `snap_external_svclib_snmp_close()`, `snap_external_svclib_snmp_INTERNAL_execpdu_handler()`, `svc_snmp_active`, and `svc_snmp_session`.

Referenced by `snap_external_svclib_getnextfunc()`.

5.91.1.15 int snap_external_svclib_snmp_initpdu (unsigned int)

Definition at line 271 of file snap_svc_snmp.c.

References `d_printf()`, `snap_external_svclib_snmp_init()`, `svc_snmp_pdu`, `svc_snmp_pdu_load`, and `svc_snmp_session`.

Referenced by `snap_external_svclib_getnextfunc()`, `snap_external_svclib_snmp_addvar_null()`, `snap_external_svclib_snmp_addvar_withvalue()`, `snap_external_svclib_snmp_gethop()`, `snap_external_svclib_snmp_getsingle()`, and `snap_external_svclib_snmp_setsingle()`.

5.91.1.16 int snap_external_svclib_snmp_isupiface (int)

Definition at line 557 of file snap_svc_snmp.c.

References `d_printf()`, `svc_returnitem::data`, `svc_returnitem::length`, `svc_returnstruct::length`, `svc_returnstruct::list`, `snap_external_svclib_snmp_getsingle()`, `svc_return`, `SVC_SNMP_TYPE_INT`, and `svc_returnitem::type`.

Referenced by `snap_external_svclib_snmp_getallotherneighboursfromip()`, and `snap_external_svclib_snmp_getnextiface()`.

5.91.1.17 int snap_external_svclib_snmp_setsingle (char *, int, void *, int)

Definition at line 439 of file snap_svc_snmp.c.

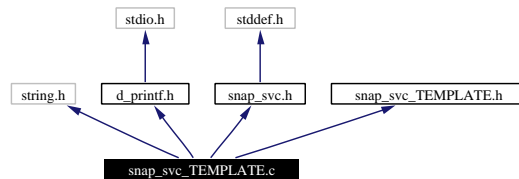
References `snap_external_svclib_snmp_addvar_withvalue()`, `snap_external_svclib_snmp_execpdu()`, and `snap_external_svclib_snmp_initpdu()`.

Referenced by `snap_external_svclib_getnextfunc()`.

5.92 snap_svc/snap_svc_TEMPLATE.c File Reference

```
#include <string.h>
#include "d_printf.h"
#include "snap_svc.h"
#include "snap_svc_TEMPLATE.h"
```

Include dependency graph for snap_svc_TEMPLATE.c:



Functions

- void `snap_external_svclib_init` ()
- void `snap_external_svclib_done` ()
- void `snap_external_svclib_getnextfunc` (char **`snapsvc_name`, `snapsvc_func_proto` *`snapsvc_func`, int *`snapsvc_args`, int *`snapsvc_rets`)

5.92.1 Function Documentation

5.92.1.1 void snap_external_svclib_done ()

Definition at line 19 of file `snap_svc_TEMPLATE.c`.

References `d_printf()`.

5.92.1.2 void snap_external_svclib_getnextfunc (char ** *snapsvc_name*, *snapsvc_func_proto* * *snapsvc_func*, int * *snapsvc_args*, int * *snapsvc_rets*)

Definition at line 24 of file `snap_svc_TEMPLATE.c`.

References `snapsvc_func_proto`, and `svc_fun_counter`.

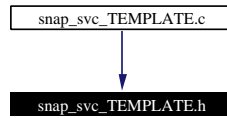
5.92.1.3 void snap_external_svclib_init ()

Definition at line 15 of file `snap_svc_TEMPLATE.c`.

References `d_printf()`.

5.93 snap_svc/snap_svc_TEMPLATE.h File Reference

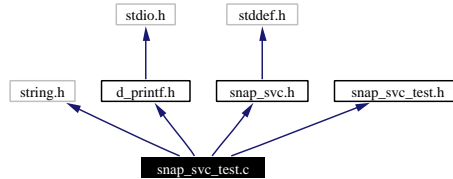
This graph shows which files directly or indirectly include this file:



5.94 snap_svc/snap_svc_test.c File Reference

```
#include <string.h>
#include "d_printf.h"
#include "snap_svc.h"
#include "snap_svc_test.h"
```

Include dependency graph for snap_svc_test.c:



Functions

- void **snap_external_svclib_init** ()
- void **snap_external_svclib_done** ()
- void **snap_external_svclib_getnextfunc** (char **snapsvc_name, snapsvc_func_proto *snapsvc_func, int *snapsvc_args, int *snapsvc_rets)
- int **snap_external_svclib_testfunc** (void *useless)
- int **snap_external_svclib_testintfunc** (int useless)
- int **snap_external_svclib_teststrfunc** (char *useless)

5.94.1 Function Documentation

5.94.1.1 void snap_external_svclib_done ()

Definition at line 74 of file snap_svc_test.c.

References `d_printf()`.

Referenced by `fini()`.

5.94.1.2 void snap_external_svclib_getnextfunc (char **snapsvc_name, snapsvc_func_proto *snapsvc_func, int *snapsvc_args, int *snapsvc_rets)

Definition at line 79 of file snap_svc_test.c.

References `snap_external_svclib_testfunc()`, `snap_external_svclib_testintfunc()`, `snap_external_svclib_teststrfunc()`, `snapsvc_func_proto`, `svc_fun_counter`, and `SVC_SNMP_TYPE_NULL`.

5.94.1.3 void snap_external_svclib_init ()

Definition at line 70 of file snap_svc_test.c.

References `d_printf()`.

Referenced by `init()`.

5.94.1.4 int snap_external_svclib_testfunc (void * *useless*)

Definition at line 110 of file snap_svc_test.c.

Referenced by `snap_external_svclib_getnextfunc()`.

5.94.1.5 int snap_external_svclib_testintfunc (int *useless*)

Definition at line 117 of file snap_svc_test.c.

Referenced by `snap_external_svclib_getnextfunc()`.

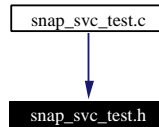
5.94.1.6 int snap_external_svclib_teststrfunc (char * *useless*)

Definition at line 127 of file snap_svc_test.c.

Referenced by `snap_external_svclib_getnextfunc()`.

5.95 snap_svc/snap_svc_test.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int **snap_external_svclib_testfunc** (void *)
- int **snap_external_svclib_testintfunc** (int)
- int **snap_external_svclib_teststrfunc** (char *)

5.95.1 Function Documentation

5.95.1.1 int snap_external_svclib_testfunc (void *)

Definition at line 110 of file snap_svc_test.c.

Referenced by snap_external_svclib_getnextfunc().

5.95.1.2 int snap_external_svclib_testintfunc (int)

Definition at line 117 of file snap_svc_test.c.

Referenced by snap_external_svclib_getnextfunc().

5.95.1.3 int snap_external_svclib_teststrfunc (char *)

Definition at line 127 of file snap_svc_test.c.

Referenced by snap_external_svclib_getnextfunc().

Index

- - intl.h, 113
 - _GNU_SOURCE
 - libsnap.c, 118
 - _PATH_DEV_ROUTE
 - pathnames.h, 143
 - _PATH_ETHERS
 - pathnames.h, 143
 - _PATH_PROCNET_ARP
 - pathnames.h, 143
 - _PATH_PROCNET_ATALK
 - pathnames.h, 143
 - _PATH_PROCNET_AX25
 - pathnames.h, 143
 - _PATH_PROCNET_AX25_ROUTE
 - pathnames.h, 143
 - _PATH_PROCNET_DEV
 - pathnames.h, 143
 - _PATH_PROCNET_IFINET6
 - pathnames.h, 143
 - _PATH_PROCNET_IPX
 - pathnames.h, 144
 - _PATH_PROCNET_IPX_ROUTE
 - pathnames.h, 144
 - _PATH_PROCNET_IP_ACC
 - pathnames.h, 143
 - _PATH_PROCNET_IP_BLK
 - pathnames.h, 143
 - _PATH_PROCNET_IP_FWD
 - pathnames.h, 144
 - _PATH_PROCNET_IP_MASQ
 - pathnames.h, 144
 - _PATH_PROCNET_NDISC
 - pathnames.h, 144
 - _PATH_PROCNET_NR
 - pathnames.h, 144
 - _PATH_PROCNET_NR_NEIGH
 - pathnames.h, 144
 - _PATH_PROCNET_NR_NODES
 - pathnames.h, 144
 - _PATH_PROCNET_RARP
 - pathnames.h, 144
 - _PATH_PROCNET_RAW
 - pathnames.h, 145
 - _PATH_PROCNET_RAW6
 - pathnames.h, 145
 - _PATH_PROCNET_ROSE_ROUTE
 - pathnames.h, 145
 - _PATH_PROCNET_ROUTE
 - pathnames.h, 145
 - _PATH_PROCNET_ROUTE6
 - pathnames.h, 145
 - _PATH_PROCNET_RTCACHE
 - pathnames.h, 145
 - _PATH_PROCNET_TCP
 - pathnames.h, 145
 - _PATH_PROCNET_TCP6
 - pathnames.h, 145
 - _PATH_PROCNET_UDP
 - pathnames.h, 145
 - _PATH_PROCNET_UDP6
 - pathnames.h, 145
 - _PATH_PROCNET_UNIX
 - pathnames.h, 146
 - _SNAP_KERN_IFACE
 - snap_kern_iface.h, 172
 - activate
 - hwtype, 20
 - activate.init
 - net-support.h, 135
 - activate.ld
 - net-support.h, 135
 - adaptive_hwtype
 - hw.c, 105
 - ADD
 - bytecode.h, 63
 - add_snap_handler
 - libsnap.c, 119
 - libsnap.h, 123
-

- ADDI
 - bytecode.h, 63
- addr
 - interface, 22
 - snap_svc_ifip_item, 39
- addr_t
 - router.h, 155
- ADDRV
 - bytecode.h, 63
- af
 - aftype, 11
- af.c
 - afinit, 55
 - afname, 56
 - aftrans, 56
 - aftrans_def, 55
 - aftrans_opt, 55
 - ax25_aftype, 56
 - ddp_aftype, 56
 - ec_aftype, 56
 - flag_ax25, 56
 - flag_ddp, 56
 - flag_econet, 56
 - flag_inet, 56
 - flag_inet6, 56
 - flag_ipx, 57
 - flag_netrom, 57
 - flag_unx, 57
 - get_afntype, 55
 - get_aftype, 55
 - inet6_aftype, 57
 - inet_aftype, 57
 - ipx_aftype, 57
 - netrom_aftype, 57
 - unix_aftype, 57
 - unspec_aftype, 57
- afinit
 - af.c, 55
- afname
 - af.c, 56
 - net-support.h, 136
- aftrans
 - af.c, 56
- AFTRANS_CNT
 - net-support.h, 132
- aftrans_def
 - af.c, 55
 - net-support.h, 135
- aftrans_opt
 - af.c, 55
- net-support.h, 135
- AFTRANS_OPTS
 - net-support.h, 132
- aftrans_t, 9
 - alias, 9
 - flag, 9
 - name, 9
- aftype, 11
 - af, 11
 - alen, 11
 - getmask, 11
 - herror, 11
 - input, 11
 - name, 11
 - print, 11
 - rinput, 12
 - rprint, 12
 - sprint, 12
 - title, 12
- alen
 - aftype, 11
 - hwtype, 20
- alias
 - aftrans_t, 9
- AND
 - bytecode.h, 64
- ANDI
 - bytecode.h, 64
- arcnet_hwtype
 - hw.c, 105
- argc
 - cmdline_args, 14
- argv
 - cmdline_args, 14
- ash_hwtype
 - hw.c, 105
- ax25_aftype
 - af.c, 56
- ax25_hwtype
 - hw.c, 106
- AX25_rinput
 - net-support.h, 135
- AX25_rprint
 - net-support.h, 135
- ax25_sock
 - sockets.h, 200
- basename
 - kinject.c, 231
 - snap_exec.c, 219

- snap_sendandreceive.c, 224
- snapas.c, 239
- snapdis.c, 243
- BCAST
 - bytecode.h, 64
- BCASTI
 - bytecode.h, 64
- BEGIN
 - snaplex.c, 247
- BEZ
 - bytecode.h, 64
- bindaddr
 - libsnap.c, 120
- BNE
 - bytecode.h, 64
- BOGUSV
 - bytecode.h, 64
- broadaddr
 - interface, 22
- buffer_handler
 - snap_demux_handler.h, 215
- buffer_t, 13
 - lenb, 13
 - s, 13
- bytecode.h
 - ADD, 63
 - ADDI, 63
 - ADDRV, 63
 - AND, 64
 - ANDI, 64
 - BCAST, 64
 - BCASTI, 64
 - BEZ, 64
 - BNE, 64
 - BOGUSV, 64
 - CALLS, 64
 - COPY_LIT, 65
 - COPY_VAL, 65
 - DEMUX, 65
 - DEMUXI, 65
 - DFORW, 65
 - DFORWTO, 65
 - DIV, 65
 - DIVI, 66
 - DSEND, 66
 - DYNCHECK_TAG, 66
 - EQ, 66
 - EQADR, 66
 - EQEXC, 66
 - EQFLT, 66
 - EQI, 66
 - EQINT, 67
 - EQSTR, 67
 - EQTUP, 67
 - EXCV, 67
 - EXIT, 67
 - FADDI, 67
 - FDIVI, 67
 - FGEQI, 68
 - FGTI, 68
 - FLEQI, 68
 - FLOATV, 68
 - FLTI, 68
 - FLTINTPAIR, 68
 - FMULI, 68
 - FORW, 69
 - FORWTO, 69
 - FSUBI, 69
 - GEQ, 69
 - GEQI, 69
 - GET_ADDR, 69
 - GET_ADDR_VAL, 69
 - GET_BOXED, 70
 - GET_FLOAT, 70
 - GET_FLT_VAL, 70
 - GET_INT, 70
 - GET_LIT, 70
 - GET_LIT_VAL, 70
 - GET_OFFS, 70
 - GET_OP, 71
 - GET_STR_VAL, 71
 - GET_TAG, 71
 - GETDST, 71
 - GETLD, 71
 - GETRB, 71
 - GETSPT, 71
 - GETSRC, 72
 - GT, 72
 - GTI, 72
 - HERE, 72
 - HOP, 72
 - instr_t, 83
 - INTV, 72
 - IS_HEAP_VAL, 72
 - ISHERE, 72
 - ISTUP, 73
 - ISX, 73
 - JI, 73
 - LEN, 73
 - LENTYPE, 73

LEQ, 73
 LEQI, 73
 LNOT, 73
 LSHL, 74
 LSHLI, 74
 LT, 74
 LTI, 74
 MAX_HEAPOBJ_SZ, 74
 MAX_VINT, 74
 MIN_VINT, 74
 MKTUP, 74
 MOD, 75
 MODI, 75
 MULT, 75
 MULTI, 75
 NEG, 75
 NEQ, 75
 NEQI, 75
 NOT, 75
 NQADR, 76
 NQEXC, 76
 NQFLT, 76
 NQINT, 76
 NQSTR, 76
 NQTUP, 76
 NTH, 76
 NUM_OPS, 77
 OPCODE_T, 77
 OR, 77
 ORI, 77
 PADDR, 77
 PAJ, 77
 PEXC, 77
 PFLT, 77
 PINT, 77
 POP, 78
 POPI, 78
 PRINT, 78
 PSTR, 78
 PTUP, 78
 PULL, 78
 PULLSTACK, 78
 PUSH, 78
 RAISEX, 79
 ROUTE, 79
 RSHA, 79
 RSHAI, 79
 RSHL, 79
 RSHLI, 79
 RTDEV, 79
 SEND, 79
 SET_ADDR, 80
 SET_FLOAT, 80
 SET_INT, 80
 SET_LIT, 80
 SET_OFFS, 81
 SET_OP, 81
 SET_TAG, 81
 SETXH, 81
 SMALL_INSTRS, 81
 SMALL_VALUES, 81
 SNET, 81
 SNETI, 81
 STACKCOUNT, 81
 STACKEMPTY, 82
 STORE, 82
 STRV, 82
 SUB, 82
 SUBI, 82
 SVCV, 82
 TAG_T, 82
 TAGSZ, 83
 TPAJ, 83
 TUPLEV, 83
 value_t, 83
 XOR, 83
 XORI, 83
 ZERO_VALUE_T, 83
 callback_master_num
 snap_svc_snmp.c, 330
 CALLS
 bytecode.h, 64
 cbuf
 snapas.c, 239
 CHECK_CODE_OVERFLOW
 snapparse.c, 261
 clear_snap_handler
 libsnap.c, 119
 libsnap.h, 123
 cmdline_args, 14
 argc, 14
 argv, 14
 cmp
 hash_table_t, 16
 code_max
 packet_t, 28
 code_min
 packet_t, 28
 code_sizeb

- const.c, 228
- const.h, 88
- snaphdr, 41
- collisions
 - user_net_device_stats, 46
- compare_longints
 - snap_exec.c, 219
 - snap_sendandreceive.c, 224
- config.h
 - CONFIG_IP_SNAP_SMALL_-
INSTRS, 85
 - CONFIG_IP_SNAP_SMALL_-
VALUES, 85
 - float32, 86
 - float64, 86
 - int32, 86
 - IS_LITTLE_ENDIAN, 85
 - uint32, 86
- CONFIG_IP_SNAP_SMALL_-
INSTRS
 - config.h, 85
- CONFIG_IP_SNAP_SMALL_-
VALUES
 - config.h, 85
- cons
 - list.h, 124
 - snap_list.c, 173
 - snap_svc_memmap_hash_list.c,
314
 - snap_svc_memmap_hash_list.h,
316
- const.c
 - code_sizeb, 228
 - heap_sizeb, 228
 - stack_sizeb, 228
- const.h
 - code_sizeb, 88
 - DEFAULT_CODE_SIZEB, 87
 - DEFAULT_HEAP_SIZEB, 87
 - DEFAULT_STACK_SIZEB, 87
 - DEFAULT_SVC_HEAP_-
SIZEB, 87
 - heap_sizeb, 88
 - stack_sizeb, 88
- conv_string
 - snaplex.c, 253
- COPY_LIT
 - bytecode.h, 65
- COPY_VAL
 - bytecode.h, 65
- cslip6_hwtype
 - hw.c, 106
- cslip_hwtype
 - hw.c, 106
- d_printf
 - snap-1.1-wjdb/lib/d_printf.c,
89
 - snap-1.1-wjdb/lib/d_printf.h,
94
 - snap_svc/d_printf.c, 91
 - snap_svc/d_printf.h, 96
- d_printf_timed
 - snap-1.1-wjdb/lib/d_printf.c,
89
 - snap-1.1-wjdb/lib/d_printf.h,
94
 - snap_svc/d_printf.c, 92
 - snap_svc/d_printf.h, 97
- daddr
 - snaphdr, 41
- data
 - svc_returnitem, 43
- ddp_aftype
 - af.c, 56
- DDP_rinput
 - net-support.h, 135
- DDP_rprint
 - net-support.h, 135
- ddp_sock
 - sockets.h, 200
- ddpaddr
 - interface, 22
- debug_level
 - snap-1.1-wjdb/lib/d_printf.c,
90
 - snap_svc/d_printf.c, 92
- debug_level_setp
 - snap-1.1-wjdb/lib/d_printf.c,
90
 - snap_svc/d_printf.c, 92
- DEF_SVC_TAB_SZ
 - snap_svc_reg_table.h, 194
- DEFAULT_CODE_SIZEB
 - const.h, 87
- DEFAULT_HEAP_SIZEB
 - const.h, 87
- DEFAULT_STACK_SIZEB
 - const.h, 87
- DEFAULT_SVC_HEAP_SIZEB

- consts.h, 87
- DEMUX
 - bytecode.h, 65
- DEMUXI
 - bytecode.h, 65
- destaddr
 - kinject.c, 231
 - snap_exec.c, 220
 - snap_sendandreceive.c, 225
- DFORW
 - bytecode.h, 65
- DFORWTO
 - bytecode.h, 65
- DIV
 - bytecode.h, 65
- DIVI
 - bytecode.h, 66
- dlsi_hwtype
 - hw.c, 106
- do_print_antitimers
 - timers.h, 203
- do_print_individual_timers
 - timers.h, 203
- do_print_item_messages
 - timers.h, 204
- doxyintro.c, 53
- DSEND
 - bytecode.h, 66
- dstaddr
 - interface, 22
- dump_all_timers
 - timers.h, 203
- DYNCHECK
 - dyncheck.h, 98
- dyncheck.h
 - DYNCHECK, 98
 - DYNCHECK_ADDR_IN_HEAP, 98
 - DYNCHECK_IN_HEAP, 98
 - DYNCHECK_RET, 99
- DYNCHECK_ADDR_IN_HEAP
 - dyncheck.h, 98
- DYNCHECK_IN_HEAP
 - dyncheck.h, 98
- DYNCHECK_RET
 - dyncheck.h, 99
- DYNCHECK_TAG
 - bytecode.h, 66
- E_INTERN
 - net-support.h, 132
- E_LOOKUP
 - net-support.h, 132
- E_NO_ROUTE
 - exception.h, 100
- E_NON_POSITIVE_RB
 - exception.h, 100
- E_NOSUPP
 - net-support.h, 132
- E_NOT_ENOUGH_RB
 - exception.h, 100
- E_NOTFOUND
 - net-support.h, 132
- E_OPTERR
 - net-support.h, 132
- E_SERVICE_ERROR
 - exception.h, 100
- E_SERVICE_NOT_PRESENT
 - exception.h, 100
- E_SOCKET
 - net-support.h, 132
- E_USAGE
 - net-support.h, 133
- E_VERSION
 - net-support.h, 133
- ec_aftype
 - af.c, 56
- ec_socket
 - sockets.h, 200
- ecaddr
 - interface, 22
- ECHO
 - snaplex.c, 247
- EINTERN
 - net-support.h, 133
- ELSE_NOT_IN_FROM_HEAP
 - snap_io.c, 166
- ENOSUPP
 - net-support.h, 133
- entry_point
 - snaphdr, 41
- EOB_ACT_CONTINUE_SCAN
 - snaplex.c, 247
- EOB_ACT_END_OF_FILE
 - snaplex.c, 247
- EOB_ACT_LAST_MATCH
 - snaplex.c, 247
- EQ
 - bytecode.h, 66
- EQADR

- bytecode.h, 66
- EQEXC
 - bytecode.h, 66
- EQFLT
 - bytecode.h, 66
- EQI
 - bytecode.h, 66
- EQINT
 - bytecode.h, 67
- EQSTR
 - bytecode.h, 67
- EQTUP
 - bytecode.h, 67
- ESYSNOT
 - net-support.h, 133
- ether_hwtype
 - hw.c, 106
- ethsock
 - libsnap.c, 120
- exception.h
 - E_NO_ROUTE, 100
 - E_NON_POSITIVE_RB, 100
 - E_NOT_ENOUGH_RB, 100
 - E_SERVICE_ERROR, 100
 - E_SERVICE_NOT_-PRESENT, 100
 - RESERVED_BOUND, 100
- EXCV
 - bytecode.h, 67
- EXIT
 - bytecode.h, 67
- FADDI
 - bytecode.h, 67
- fddi_hwtype
 - hw.c, 106
- FDIVI
 - bytecode.h, 67
- fdset
 - snap_demux_handler.c, 212
- FGEQI
 - bytecode.h, 68
- FGTI
 - bytecode.h, 68
- file
 - snaplex.c, 253
- file_to_str
 - io.h, 115
 - snap_io.c, 167
- fini
 - snap_svc.c, 290
 - snap_svc.h, 294
- flag
 - aftrans_t, 9
 - heap_obj, 18
- flag_ax25
 - af.c, 56
 - net-support.h, 136
- FLAG_CACHE
 - net-support.h, 133
- flag_ddp
 - af.c, 56
 - net-support.h, 136
- flag_econet
 - af.c, 56
- FLAG_EXT
 - net-support.h, 133
- FLAG_FIB
 - net-support.h, 133
- flag_inet
 - af.c, 56
 - net-support.h, 137
- flag_inet6
 - af.c, 56
 - net-support.h, 137
- flag_ipx
 - af.c, 57
 - net-support.h, 137
- flag_netrom
 - af.c, 57
 - net-support.h, 137
- FLAG_NUM
 - net-support.h, 134
- FLAG_SYM
 - net-support.h, 134
- flag_unix
 - af.c, 57
 - net-support.h, 137
- FLAG_VERBOSE
 - net-support.h, 134
- flags
 - interface, 22
 - snaphdr, 41
- FLEQI
 - bytecode.h, 68
- FLEX_SCANNER
 - snaplex.c, 247
- float32
 - config.h, 86
- float64

- config.h, 86
- FLOATV
 - bytecode.h, 68
- FLTI
 - bytecode.h, 68
- FLTINTPAIR
 - bytecode.h, 68
- FMULI
 - bytecode.h, 68
- FORW
 - bytecode.h, 69
- FORWTO
 - bytecode.h, 69
- fprintf_addr
 - packet.c, 138
 - snap_bytecode.c, 158
- fprintf_instr
 - packet.c, 138
 - snap_bytecode.c, 158
- fprintf_opcode
 - snap_bytecode.c, 159
- fprintf_packet
 - packet.c, 138
 - packet.h, 141
- fprintf_value
 - packet.c, 139
 - snap_bytecode.c, 159
- fprintf_value_heap
 - snap_bytecode.c, 159
- fprintf_value_tag
 - snap_bytecode.c, 159
- frad_hwtype
 - hw.c, 106
- free_list
 - list.h, 124
 - snap_list.c, 173
 - snap_svc_memmap_hash_list.c, 314
 - snap_svc_memmap_hash_list.h, 316
- FSUBI
 - bytecode.h, 69
- gc
 - libsnap.c, 120
- GEQ
 - bytecode.h, 69
- GEQI
 - bytecode.h, 69
- GET_ADDR
 - bytecode.h, 69
- GET_ADDR_VAL
 - bytecode.h, 69
- get_afntype
 - af.c, 55
 - net-support.h, 135
- get_aftype
 - af.c, 55
 - net-support.h, 135
- GET_BOXED
 - bytecode.h, 70
- GET_FLOAT
 - bytecode.h, 70
- GET_FLT_VAL
 - bytecode.h, 70
- get_hwntype
 - hw.c, 105
 - net-support.h, 135
- get_hwtype
 - hw.c, 105
 - net-support.h, 135
- get_iface_index
 - router.c, 152
- GET_INT
 - bytecode.h, 70
- GET_INT_HEAP
 - snap_bytecode.c, 158
- GET_LIT
 - bytecode.h, 70
- GET_LIT_VAL
 - bytecode.h, 70
- GET_OFFS
 - bytecode.h, 70
- GET_OFFS_HEAP
 - snap_bytecode.c, 158
- GET_OP
 - bytecode.h, 71
- get_sname
 - net-support.h, 136
- GET_STR_VAL
 - bytecode.h, 71
- GET_TAG
 - bytecode.h, 71
- GET_TAG_HEAP
 - snap_bytecode.c, 158
- getargs
 - net-support.h, 136
- GETDST
 - bytecode.h, 71
- GETLD

- bytecode.h, 71
- getmask
 - aftype, 11
- GETRB
 - bytecode.h, 71
- getroute_init
 - net-support.h, 136
- GETSPT
 - bytecode.h, 71
- GETSRC
 - bytecode.h, 72
- glob_conf, 15
- herehint, 15
- GT
 - bytecode.h, 72
- GTI
 - bytecode.h, 72
- h
 - heap_t, 19
- h_alloc_heap_max
 - packet_t, 28
- h_alloc_ptr
 - packet_t, 29
- handle_request
 - snap_svc_route.c, 321
- handle_snap_request
 - libsnap.c, 119
 - libsnap.h, 123
- handler
 - packet_t, 29
- has_ax25
 - interface, 22
- has_ddp
 - interface, 22
- has_econet
 - interface, 22
- has_ip
 - interface, 23
- has_ipx_bb
 - interface, 23
- has_ipx_e2
 - interface, 23
- has_ipx_e3
 - interface, 23
- has_ipx_sn
 - interface, 23
- hash
 - hash_table_t, 16
- hashtable.h, 102
- snap_hashtable.c, 161
- snap_svc_memmap_hash.c, 310
- snap_svc_memmap_hash.h, 312
- hash_table_t, 16
 - cmp, 16
 - hash, 16
 - max_len, 16
 - tab, 16
 - tab_sz, 17
- hashtable.h
 - hash_string, 102
 - ht_create, 102
 - ht_errno, 103
 - ht_insert, 103
 - ht_lookup, 103
 - ht_remove, 103
- hbuf
 - snapas.c, 239
- hdlc_hwtype
 - hw.c, 106
- hDllList
 - snap_svc_reg_handler.c, 187
- hdr
 - packet_t, 29
- header_t
 - packet.h, 141
- heap_alloc
 - interp.h, 112
- heap_max
 - packet_t, 29
- heap_min
 - packet_t, 29
- heap_obj, 18
 - flag, 18
 - len, 18
 - s, 18
- heap_sizeb
 - consts.c, 228
 - consts.h, 88
 - snaphdr, 41
- heap_t, 19
 - h, 19
 - lenb, 19
- HERE
 - bytecode.h, 72
- herehint
 - glob_conf, 15
- herror
 - aftype, 11

- hippi_hwtype
 - hw.c, 106
- HOP
 - bytecode.h, 72
- hopaddr
 - rt_lookup, 32
- ht_create
 - hashtable.h, 102
 - snap_hashtable.c, 161
 - snap_svc_memmap_hash.c, 310
 - snap_svc_memmap_hash.h, 312
- ht_errno
 - hashtable.h, 103
 - snap_hashtable.c, 162
 - snap_svc_memmap_hash.c, 311
 - snap_svc_memmap_hash.h, 313
- ht_insert
 - hashtable.h, 103
 - snap_hashtable.c, 161
 - snap_svc_memmap_hash.c, 311
 - snap_svc_memmap_hash.h, 313
- ht_lookup
 - hashtable.h, 103
 - snap_hashtable.c, 162
 - snap_svc_memmap_hash.c, 311
 - snap_svc_memmap_hash.h, 313
- ht_remove
 - hashtable.h, 103
 - snap_hashtable.c, 162
 - snap_svc_memmap_hash.c, 311
 - snap_svc_memmap_hash.h, 313
- hw.c
 - adaptive_hwtype, 105
 - arcnet_hwtype, 105
 - ash_hwtype, 105
 - ax25_hwtype, 106
 - cslip6_hwtype, 106
 - cslip_hwtype, 106
 - dlci_hwtype, 106
 - ether_hwtype, 106
 - fddi_hwtype, 106
 - frad_hwtype, 106
 - get_hwntype, 105
 - get_hwtype, 105
 - hdlc_hwtype, 106
 - hippi_hwtype, 106
 - hwinit, 105
 - lapb_hwtype, 106
 - loop_hwtype, 106
 - netrom_hwtype, 107
 - ppp_hwtype, 107
 - rose_hwtype, 107
 - sit_hwtype, 107
 - slip6_hwtype, 107
 - slip_hwtype, 107
 - tr_hwtype, 107
 - tunnel_hwtype, 107
 - unspec_hwtype, 107
- hwaddr
 - interface, 23
- hwinit
 - hw.c, 105
- hwtype, 20
 - activate, 20
 - alen, 20
 - input, 20
 - name, 20
 - print, 20
 - sprint, 20
 - title, 20
 - type, 20
- if_fetch
 - interface.c, 110
 - interface.h, 111
- if_get_interface_count
 - snap_svc_if.c, 298
 - snap_svc_if.h, 302
- if_get_interface_name
 - snap_svc_if.c, 298
 - snap_svc_if.h, 302
- if_getallneighbours
 - snap_svc_if.c, 298
 - snap_svc_if.h, 302
- if_gethopfromiface
 - snap_svc_if.c, 299
- if_getiface
 - snap_svc_if.c, 299
 - snap_svc_if.h, 303
- if_getiface_up
 - snap_svc_if.c, 299
 - snap_svc_if.h, 303
- if_getifaceidx
 - snap_svc_if.c, 299
- if_getnexthop
 - snap_svc_if.c, 299
 - snap_svc_if.h, 303
- if_getnextiface
 - snap_svc_if.c, 299
- if_getoutiface

- snap_svc_if.c, 299
- IF_IN_FROM_HEAP
 - snap_io.c, 167
- if_index
 - snap_svc_ifip_item, 39
- if_name
 - snap_svc_ifip_item, 39
- if_setiface
 - snap_svc_if.c, 300
 - snap_svc_if.h, 303
- if_setiface_up
 - snap_svc_if.c, 300
 - snap_svc_if.h, 303
- iface_list
 - snap_svc_if.c, 301
- ifaces
 - router.c, 153
- ifidx
 - rt_lookup, 32
- IN_TOSPACE_HEAP
 - snap_io.c, 167
- inbuf
 - snap_exec.c, 220
 - snap_sendandreceive.c, 225
- inet6_aftype
 - af.c, 57
- INET6_rinput
 - net-support.h, 136
- INET6_rprint
 - net-support.h, 136
- inet6_sock
 - sockets.h, 201
- inet_aftype
 - af.c, 57
- INET_rinput
 - net-support.h, 136
- INET_rprint
 - net-support.h, 136
- inet_sock
 - sockets.h, 201
- infd
 - kinject.c, 232
 - snap_exec.c, 220
 - snap_sendandreceive.c, 226
 - snapdis.c, 243
- infilename
 - snapas.c, 239
- init
 - snap_svc.c, 290
 - snap_svc.h, 294
- init_all_timers
 - timers.h, 203
- init_request
 - snap_exec.c, 219
 - snap_sendandreceive.c, 225
- init_snap
 - libsnap.c, 119
 - libsnap.h, 123
- INITIAL
 - snaplex.c, 247
- input
 - aftype, 11
 - hwtype, 20
- instr_t
 - bytecode.h, 83
- int32
 - config.h, 86
- integer
 - yystype, 52
- interface, 21
 - addr, 22
 - broadaddr, 22
 - ddpaddr, 22
 - dstaddr, 22
 - ecaddr, 22
 - flags, 22
 - has_ax25, 22
 - has_ddp, 22
 - has_econet, 22
 - has_ip, 23
 - has_ipx_bb, 23
 - has_ipx_e2, 23
 - has_ipx_e3, 23
 - has_ipx_sn, 23
 - hwaddr, 23
 - ipxaddr_bb, 23
 - ipxaddr_e2, 23
 - ipxaddr_e3, 24
 - ipxaddr_sn, 24
 - map, 24
 - metric, 24
 - mtu, 24
 - name, 24
 - netmask, 24
 - stats, 24
 - tx_queue_len, 25
 - type, 25
- interface.c
 - if_fetch, 110
 - procnetdev_vsn, 110

- interface.h
 - if_fetch, 111
 - procnetdev_vsn, 111
- internal_print_anti_time
 - timers.h, 203
- internal_print_time
 - timers.h, 203
- interp.h
 - heap_alloc, 112
 - snap_interp_packet, 112
- intl.h
 - , 113
 - N-, 113
- INTV
 - bytecode.h, 72
- io.h
 - file_to_str, 115
 - marshal_packet, 115
 - unmarshal_packet, 115
- ip_masq_info
 - net-support.h, 136
- ip_to_u32
 - snap_svc_route.c, 321
- iph
 - packet_t, 29
- IPPROTO_SNAP
 - kinject.c, 231
 - snap.h, 156
 - snap_exec.c, 219
 - snap_sendandreceive.c, 224
- ipx_aftype
 - af.c, 57
- IPX_rinput
 - net-support.h, 136
- IPX_rprint
 - net-support.h, 136
- ipx_sock
 - sockets.h, 201
- ipxaddr_bb
 - interface, 23
- ipxaddr_e2
 - interface, 23
- ipxaddr_e3
 - interface, 24
- ipxaddr_sn
 - interface, 24
- is_contiguous
 - packet_t, 29
- IS_HEAP_VAL
 - bytecode.h, 72
- IS_LITTLE_ENDIAN
 - config.h, 85
- ISHERE
 - bytecode.h, 72
- isset_snap_handler
 - libsnap.c, 119
 - libsnap.h, 123
- ISTUP
 - bytecode.h, 73
- ISX
 - bytecode.h, 73
- JI
 - bytecode.h, 73
- key
 - pair_t, 31
- kinject.c
 - basename, 231
 - destaddr, 231
 - infd, 232
 - IPPROTO_SNAP, 231
 - main, 231
 - NIPQUAD, 231
 - out_ttl, 232
 - parse_cmdline, 231
 - udpport, 232
 - usage, 231
- l, 26
 - next, 26
 - v, 26
- label_mapping_t, 27
 - name, 27
 - where, 27
- labels.c
 - MAX_NUM_LABELS, 234
 - patch_jumps, 234
 - register_label_def, 234
 - register_label_use, 234
- labels.h
 - patch_jumps, 235
 - register_label_def, 235
 - register_label_use, 236
- lapb_hwtype
 - hw.c, 106
- LEN
 - bytecode.h, 73
- len
 - heap_obj, 18

- snaplex.c, 254
- lenb
 - buffer_t, 13
 - heap_t, 19
- length
 - svc_returnitem, 43
 - svc_returnstruct, 45
- length_list
 - list.h, 124
 - snap_list.c, 173
 - snap_svc_mmap_hash_list.c, 314
 - snap_svc_mmap_hash_list.h, 316
- LENTYPE
 - bytecode.h, 73
- LEQ
 - bytecode.h, 73
- LEQI
 - bytecode.h, 73
- lexbuf
 - snaplex.c, 254
- lexbuf_len
 - snaplex.c, 254
- lexbuf_pos
 - snaplex.c, 254
- libsnap.c
 - _GNU_SOURCE, 118
 - add_snap_handler, 119
 - bindaddr, 120
 - clear_snap_handler, 119
 - ethsock, 120
 - gc, 120
 - handle_snap_request, 119
 - init_snap, 119
 - isset_snap_handler, 119
 - losock, 120
 - maxfd, 121
 - NIPQUAD, 118
 - p, 121
 - parse_cmdline_snap, 119, 120
 - ra_space, 121
 - rawiprecvsock, 121
 - rfd, 121
 - snap, 120
 - snap_receive, 120
 - udpaddr, 121
 - UDPSPORT, 119
 - usage, 120
- libsnap.h
 - add_snap_handler, 123
 - clear_snap_handler, 123
 - handle_snap_request, 123
 - init_snap, 123
 - isset_snap_handler, 123
 - SNAP_LIB_H, 122
 - snap_receive, 123
- list
 - svc_returnstruct, 45
- list.h
 - cons, 124
 - free_list, 124
 - length_list, 124
 - list_t, 124
- list_t
 - list.h, 124
 - snap_svc_mmap_hash_list.h, 316
- LNOT
 - bytecode.h, 73
- localaddr
 - snap_exec.c, 221
 - snap_sendandreceive.c, 226
- loop_hwtype
 - hw.c, 106
 - loopback.c, 125
- loopback.c
 - loop_hwtype, 125
 - unspec_hwtype, 126
- losock
 - libsnap.c, 120
- LSHL
 - bytecode.h, 74
- LSHLI
 - bytecode.h, 74
- LT
 - bytecode.h, 74
- LTI
 - bytecode.h, 74
- main
 - kinject.c, 231
 - snap_exec.c, 220
 - snap_sendandreceive.c, 225
 - snapas.c, 239
 - snapd.c, 227
 - snapdis.c, 243
- make_sockaddr_in
 - snap_svc_route.c, 321
- map

- interface, 24
- marshal_packet
 - io.h, 115
 - snap_io.c, 167
- max_filedes
 - snap_demux_handler.c, 212
- MAX_HEAPOBJ_SZ
 - bytecode.h, 74
- max_len
 - hash_table_t, 16
- MAX_MTU
 - snapnet.c, 196
- MAX_NUM_LABELS
 - labels.c, 234
- MAX_VINT
 - bytecode.h, 74
- maxfd
 - libsnap.c, 121
- memalloc
 - memalloc.h, 127
 - snap_svc_memmap_hash_list.h, 315
- memalloc.h
 - memalloc, 127
- metric
 - interface, 24
- MIN_VINT
 - bytecode.h, 74
- MKTUP
 - bytecode.h, 74
- MOD
 - bytecode.h, 75
- MODI
 - bytecode.h, 75
- mtu
 - interface, 24
- MULT
 - bytecode.h, 75
- MULTI
 - bytecode.h, 75
- mystrcmp
 - snap_svc_memmap.c, 305
 - snap_svc_reg_table.c, 192
- n
 - snap_htup, 36
- N_
 - intl.h, 113
- name
 - aftrans_t, 9
 - aftype, 11
 - hwtype, 20
 - interface, 24
 - label_mapping_t, 27
- nargs
 - snap_svc_rec, 40
- NEG
 - bytecode.h, 75
- NEQ
 - bytecode.h, 75
- NEQI
 - bytecode.h, 75
- net-support.h
 - activate_init, 135
 - activate_ld, 135
 - afname, 136
 - AFTRANS_CNT, 132
 - aftrans_def, 135
 - aftrans_opt, 135
 - AFTRANS_OPTS, 132
 - AX25_rinput, 135
 - AX25_rprint, 135
 - DDP_rinput, 135
 - DDP_rprint, 135
 - E_INTERN, 132
 - E_LOOKUP, 132
 - E_NOSUPP, 132
 - E_NOTFOUND, 132
 - E_OPTERR, 132
 - E_SOCKET, 132
 - E_USAGE, 133
 - E_VERSION, 133
 - EINTERN, 133
 - ENOSUPP, 133
 - ESYSNOT, 133
 - flag_ax25, 136
 - FLAG_CACHE, 133
 - flag_ddp, 136
 - FLAG_EXT, 133
 - FLAG_FIB, 133
 - flag_inet, 137
 - flag_inet6, 137
 - flag_ipx, 137
 - flag_netrom, 137
 - FLAG_NUM, 134
 - FLAG_SYM, 134
 - flag_unx, 137
 - FLAG_VERBOSE, 134
 - get_afntype, 135
 - get_aftype, 135

- get_hwntype, 135
- get_hwtype, 135
- get_sname, 136
- getargs, 136
- getroute_init, 136
- INET6_rinput, 136
- INET6_rprint, 136
- INET_rinput, 136
- INET_rprint, 136
- ip_masq_info, 136
- IPX_rinput, 136
- IPX_rprint, 136
- NETROM_rinput, 136
- NETROM_rprint, 136
- route_edit, 136
- route_info, 136
- RTACTION_ADD, 134
- RTACTION_DEL, 134
- RTACTION_FLUSH, 134
- RTACTION_HELP, 134
- RTACTION_SHOW, 134
- setroute_init, 136
- netmask
 - interface, 24
- netrom_aftype
 - af.c, 57
- netrom_hwtype
 - hw.c, 107
- NETROM_rinput
 - net-support.h, 136
- NETROM_rprint
 - net-support.h, 136
- newho
 - snapparse.c, 275
- newtup
 - snapparse.c, 275
- next
 - l, 26
- nexthop
 - router.c, 152
 - router.h, 155
- NIPQUAD
 - kinject.c, 231
 - libsnap.c, 118
 - router.c, 152
 - snap_exec.c, 219
 - snap_interp.c, 164
 - snap_kern_iface.c, 171
 - snap_sendandreceive.c, 224
- NO_RUNS
 - snap_sendandreceive.c, 224
- noop
 - snapas.c, 240
 - snapparse.c, 276
- NOT
 - bytecode.h, 75
- NQADR
 - bytecode.h, 76
- NQEXC
 - bytecode.h, 76
- NQFLT
 - bytecode.h, 76
- NQINT
 - bytecode.h, 76
- NQSTR
 - bytecode.h, 76
- NQTUP
 - bytecode.h, 76
- nret
 - snap_svc_rec, 40
- NTH
 - bytecode.h, 76
- num_if_entries
 - router.c, 153
- num_ifaces
 - router.c, 153
- NUM_OPS
 - bytecode.h, 77
- num_routes
 - router.c, 153
- num_rt_entries
 - router.c, 153
- oid
 - svc_returnitem, 43
- oid_length
 - svc_returnitem, 43
- OPCODE_T
 - bytecode.h, 77
- OR
 - bytecode.h, 77
- ORI
 - bytecode.h, 77
- out_ttl
 - kinject.c, 232
 - snap_exec.c, 221
 - snap_sendandreceive.c, 226
- outfd
 - snapas.c, 240
- outfile

- snapdis.c, 243
- outfilename
 - snapas.c, 240
- P
 - libsnap.c, 121
 - snapas.c, 240
 - snapparse.c, 276
- packet.c
 - fprintf_addr, 138
 - fprintf_instr, 138
 - fprintf_packet, 138
 - fprintf_value, 139
- packet.h
 - fprintf_packet, 141
 - header_t, 141
- packet_t, 28
 - code_max, 28
 - code_min, 28
 - h_alloc_heap_max, 28
 - h_alloc_ptr, 29
 - handler, 29
 - hdr, 29
 - heap_max, 29
 - heap_min, 29
 - iph, 29
 - is_contiguous, 29
 - pc, 30
 - rb, 30
 - resized, 30
 - sp, 30
 - stack_max, 30
 - stack_min, 30
- PADDR
 - bytecode.h, 77
- pair_t, 31
 - key, 31
 - value, 31
- PAJ
 - bytecode.h, 77
- parse_cmdline
 - kinject.c, 231
 - snap_exec.c, 220
 - snap_sendandreceive.c, 225
 - snapas.c, 239
 - snapdis.c, 243
- parse_cmdline_snap
 - libsnap.c, 119, 120
- patch_jumps
 - labels.c, 234
- labels.h, 235
- pathnames.h
 - _PATH_DEV_ROUTE, 143
 - _PATH_ETHERS, 143
 - _PATH_PROCNET_ARP, 143
 - _PATH_PROCNET_ATALK, 143
 - _PATH_PROCNET_AX25, 143
 - _PATH_PROCNET_AX25_-ROUTE, 143
 - _PATH_PROCNET_DEV, 143
 - _PATH_PROCNET_IFINET6, 143
 - _PATH_PROCNET_IPX, 144
 - _PATH_PROCNET_IPX_-ROUTE, 144
 - _PATH_PROCNET_IP_ACC, 143
 - _PATH_PROCNET_IP_BLK, 143
 - _PATH_PROCNET_IP_FWD, 144
 - _PATH_PROCNET_IP_-MASQ, 144
 - _PATH_PROCNET_NDISC, 144
 - _PATH_PROCNET_NR, 144
 - _PATH_PROCNET_NR_-NEIGH, 144
 - _PATH_PROCNET_NR_-NODES, 144
 - _PATH_PROCNET_RARP, 144
 - _PATH_PROCNET_RAW, 145
 - _PATH_PROCNET_RAW6, 145
 - _PATH_PROCNET_ROSE_-ROUTE, 145
 - _PATH_PROCNET_ROUTE, 145
 - _PATH_PROCNET_ROUTE6, 145
 - _PATH_PROCNET_-RTCACHE, 145
 - _PATH_PROCNET_TCP, 145
 - _PATH_PROCNET_TCP6, 145
 - _PATH_PROCNET_UDP, 145
 - _PATH_PROCNET_UDP6, 145

- `_PATH_PROCKET_UNIX`, 146
- `pbuf`
 - `snapas.c`, 240
 - `snapnet.c`, 197
- `pc`
 - `packet_t`, 30
- `PEXC`
 - `bytecode.h`, 77
- `PFLT`
 - `bytecode.h`, 77
- `PINT`
 - `bytecode.h`, 77
- `POP`
 - `bytecode.h`, 78
- `POPI`
 - `bytecode.h`, 78
- `ppp_hwtype`
 - `hw.c`, 107
- `pReturnFree`
 - `snap_svc_reg_handler.c`, 187
- `pReturnLast`
 - `snap_svc_reg_handler.c`, 187
- `PRINT`
 - `bytecode.h`, 78
- `print`
 - `aftype`, 11
 - `hwtype`, 20
- `print_anti_mtimer`
 - `timers.h`, 202
- `print_anti_timer`
 - `timers.h`, 202
- `print_flag_count`
 - `timers.h`, 204
- `print_flags`
 - `timers.h`, 204
- `print_mtimer`
 - `timers.h`, 203
- `print_timer`
 - `timers.h`, 203
- `printip`
 - `snap_svc_route.c`, 321
- `printk_addr`
 - `printval.h`, 148
- `printk_instr`
 - `printval.h`, 148
- `printk_opcode`
 - `printval.h`, 148
- `printk_value`
 - `printval.h`, 148
- `printk_value_tag`
 - `printval.h`, 148
- `printval.h`
 - `printk_addr`, 148
 - `printk_instr`, 148
 - `printk_opcode`, 148
 - `printk_value`, 148
 - `printk_value_tag`, 148
- `proc.c`
 - `proc_gen_fmt`, 149
- `proc.h`
 - `proc_gen_fmt`, 150
- `proc_gen_fmt`
 - `proc.c`, 149
 - `proc.h`, 150
- `PROC_NET_DEV_PATH`
 - `router.h`, 155
- `PROC_NET_ROUTE_PATH`
 - `router.h`, 155
- `proc_sysnetip_getforwarding`
 - `snap_svc_proc.c`, 317
 - `snap_svc_proc.h`, 319
- `proc_sysnetip_setforwarding`
 - `snap_svc_proc.c`, 317
 - `snap_svc_proc.h`, 319
- `procketdev_vsn`
 - `interface.c`, 110
 - `interface.h`, 111
- `protocols_internal`
 - `snap_demux_handler.c`, 212
- `PSTR`
 - `bytecode.h`, 78
- `ptr`
 - `yystype`, 52
- `PTUP`
 - `bytecode.h`, 78
- `PULL`
 - `bytecode.h`, 78
- `PULLSTACK`
 - `bytecode.h`, 78
- `PUSH`
 - `bytecode.h`, 78
- `ra_space`
 - `libsnap.c`, 121
- `RAISEX`
 - `bytecode.h`, 79
- `rawiprecvsock`
 - `libsnap.c`, 121
- `rb`

- packet_t, 30
- read_from_file
 - snaplex.c, 254
- read_ifaces
 - router.c, 152
 - router.h, 155
- read_routes
 - router.c, 152
 - router.h, 155
- receiveport
 - snap_demux_handler.h, 216
 - snap_exec.c, 221
 - snap_sendandreceive.c, 226
- refine_op
 - snapparse.c, 275
- register_label_def
 - labels.c, 234
 - labels.h, 235
- register_label_use
 - labels.c, 234
 - labels.h, 236
- REJECT
 - snaplex.c, 247
- RESERVED_BOUND
 - exception.h, 100
- resized
 - packet_t, 30
- rfds
 - libsnap.c, 121
- rinput
 - aftype, 12
- rose_hwtype
 - hw.c, 107
- rose_sock
 - sockets.h, 201
- ROUTE
 - bytecode.h, 79
- route_add
 - snap_svc_route.c, 321
 - snap_svc_route.h, 323
- route_del
 - snap_svc_route.c, 321
 - snap_svc_route.h, 323
- route_edit
 - net-support.h, 136
- route_info
 - net-support.h, 136
- router.c
 - get_iface_index, 152
 - ifaces, 153
 - nexthop, 152
 - NIPQUAD, 152
 - num_if_entries, 153
 - num_ifaces, 153
 - num_routes, 153
 - num_rt_entries, 153
 - read_ifaces, 152
 - read_routes, 152
 - routes, 153
- router.h
 - addr_t, 155
 - nexthop, 155
 - PROC_NET_DEV_PATH, 155
 - PROC_NET_ROUTE_PATH, 155
 - read_ifaces, 155
 - read_routes, 155
- routes
 - router.c, 153
- rprint
 - aftype, 12
- RSHA
 - bytecode.h, 79
- RSHAI
 - bytecode.h, 79
- RSHL
 - bytecode.h, 79
- RSHLI
 - bytecode.h, 79
- rt_class
 - rtentry, 33
- rt_dev
 - rtentry, 33
- rt_dst
 - rtentry, 33
- rt_flags
 - rtentry, 33
- rt_gateway
 - rtentry, 33
- rt_genmask
 - rtentry, 33
- rt_ifidx
 - rtentry, 34
- rt_irtt
 - rtentry, 34
- rt_lookup, 32
 - hopaddr, 32
 - ifidx, 32
- rt_metric
 - rtentry, 34

- rt_mtu
 - rtentry, 34
- rt_pad2
 - rtentry, 34
- rt_pad4
 - rtentry, 34
- rt_tos
 - rtentry, 34
- rt_window
 - rtentry, 34
- RTACTION_ADD
 - net-support.h, 134
- RTACTION_DEL
 - net-support.h, 134
- RTACTION_FLUSH
 - net-support.h, 134
- RTACTION_HELP
 - net-support.h, 134
- RTACTION_SHOW
 - net-support.h, 134
- RTDEV
 - bytecode.h, 79
- rtentry, 33
 - rt_class, 33
 - rt_dev, 33
 - rt_dst, 33
 - rt_flags, 33
 - rt_gateway, 33
 - rt_genmask, 33
 - rt_ifidx, 34
 - rt_irtt, 34
 - rt_metric, 34
 - rt_mtu, 34
 - rt_pad2, 34
 - rt_pad4, 34
 - rt_tos, 34
 - rt_window, 34
- rx_bytes
 - user_net_device_stats, 46
- rx_compressed
 - user_net_device_stats, 46
- rx_crc_errors
 - user_net_device_stats, 46
- rx_dropped
 - user_net_device_stats, 47
- rx_errors
 - user_net_device_stats, 47
- rx_fifo_errors
 - user_net_device_stats, 47
- rx_frame_errors
 - user_net_device_stats, 47
- rx_length_errors
 - user_net_device_stats, 47
- rx_missed_errors
 - user_net_device_stats, 47
- rx_multicast
 - user_net_device_stats, 47
- rx_over_errors
 - user_net_device_stats, 47
- rx_packets
 - user_net_device_stats, 47
- s
 - buffer_t, 13
 - heap_obj, 18
 - snap_hval, 37
- saddr
 - snaphdr, 41
- sbuf
 - snapas.c, 240
- scalar
 - snap_hval, 37
- sd
 - snap_exec.c, 221
 - snap_sendandreceive.c, 226
- SEND
 - bytecode.h, 79
- sendpkt
 - snap_exec.c, 220
 - snap_sendandreceive.c, 225
- SET_ADDR
 - bytecode.h, 80
- set_debug_level
 - snap-1.1-wjdb/lib/d_printf.c, 89
 - snap-1.1-wjdb/lib/d_printf.h, 94
 - snap_svc/d_printf.c, 92
 - snap_svc/d_printf.h, 97
- set_debug_level_int
 - snap-1.1-wjdb/lib/d_printf.c, 90
 - snap-1.1-wjdb/lib/d_printf.h, 94
 - snap_svc/d_printf.c, 92
 - snap_svc/d_printf.h, 97
- SET_FLOAT
 - bytecode.h, 80
- SET_INT
 - bytecode.h, 80

- SET_INT_HEAP
 - snap_bytecode.c, 158
- SET_LIT
 - bytecode.h, 80
- SET_OFFS
 - bytecode.h, 81
- SET_OFFS_HEAP
 - snap_bytecode.c, 158
- SET_OP
 - bytecode.h, 81
- SET_TAG
 - bytecode.h, 81
- SET_TAG_HEAP
 - snap_bytecode.c, 158
- setroute_init
 - net-support.h, 136
- SETXH
 - bytecode.h, 81
- sit_hwtype
 - hw.c, 107
- size
 - snaplex.c, 254
- skfd
 - sockets.c, 199
 - sockets.h, 201
- slip6_hwtype
 - hw.c, 107
- slip_hwtype
 - hw.c, 107
- SMALL_INSTRS
 - bytecode.h, 81
- SMALL_VALUES
 - bytecode.h, 81
- snap
 - libsnap.c, 120
- snap-1.1-wjdb/lib/af.c, 54
- snap-1.1-wjdb/lib/bytecode.h, 58
- snap-1.1-wjdb/lib/config.h, 85
- snap-1.1-wjdb/lib/consts.h, 87
- snap-1.1-wjdb/lib/d_printf.c, 89
 - d_printf, 89
 - d_printf_timed, 89
 - debug_level, 90
 - debug_level_setp, 90
 - set_debug_level, 89
 - set_debug_level_int, 90
- snap-1.1-wjdb/lib/d_printf.h, 94
 - d_printf, 94
 - d_printf_timed, 94
 - set_debug_level, 94
 - set_debug_level_int, 94
 - sysctl_snap_debug_level, 95
- snap-1.1-wjdb/lib/dyncheck.h, 98
- snap-1.1-wjdb/lib/exception.h, 100
- snap-1.1-wjdb/lib/hashtable.h, 102
- snap-1.1-wjdb/lib/hw.c, 104
- snap-1.1-wjdb/lib/inet.c, 108
- snap-1.1-wjdb/lib/interface.c, 109
- snap-1.1-wjdb/lib/interface.h, 111
- snap-1.1-wjdb/lib/interp.h, 112
- snap-1.1-wjdb/lib/intl.h, 113
- snap-1.1-wjdb/lib/io.h, 114
- snap-1.1-wjdb/lib/libsnap.c, 116
- snap-1.1-wjdb/lib/libsnap.h, 122
- snap-1.1-wjdb/lib/list.h, 124
- snap-1.1-wjdb/lib/loopback.c, 125
- snap-1.1-wjdb/lib/memalloc.h, 127
- snap-1.1-wjdb/lib/myassert.h, 129
- snap-1.1-wjdb/lib/net-support.h, 130
- snap-1.1-wjdb/lib/packet.c, 138
- snap-1.1-wjdb/lib/packet.h, 140
- snap-1.1-wjdb/lib/pathnames.h, 142
- snap-1.1-wjdb/lib/printval.h, 147
- snap-1.1-wjdb/lib/proc.c, 149
- snap-1.1-wjdb/lib/proc.h, 150
- snap-1.1-wjdb/lib/router.c, 151
- snap-1.1-wjdb/lib/router.h, 154
- snap-1.1-wjdb/lib/snap.h, 156
- snap-1.1-wjdb/lib/snap_bytecode.c, 157
- snap-1.1-wjdb/lib/snap_-hashtable.c, 161
- snap-1.1-wjdb/lib/snap_interp.c, 163
- snap-1.1-wjdb/lib/snap_io.c, 165
- snap-1.1-wjdb/lib/snap_kern_-iface.c, 169
- snap-1.1-wjdb/lib/snap_kern_-iface.h, 172
- snap-1.1-wjdb/lib/snap_list.c, 173
- snap-1.1-wjdb/lib/snap_svc_-conversion.c, 174
- snap-1.1-wjdb/lib/snap_svc_-conversion.h, 176
- snap-1.1-wjdb/lib/snap_svc_-handler.c, 178
- snap-1.1-wjdb/lib/snap_svc_-handler.h, 179

- snap-1.1-wjdb/lib/snap_svc_-
 - library_handler.c, 180
- snap-1.1-wjdb/lib/snap_svc_-
 - library_handler.h, 182
- snap-1.1-wjdb/lib/snap_svc_reg_-
 - handler.c, 185
- snap-1.1-wjdb/lib/snap_svc_reg_-
 - handler.h, 189
- snap-1.1-wjdb/lib/snap_svc_reg_-
 - table.c, 192
- snap-1.1-wjdb/lib/snap_svc_reg_-
 - table.h, 194
- snap-1.1-wjdb/lib/snapnet.c, 196
- snap-1.1-wjdb/lib/snapnet.h, 198
- snap-1.1-wjdb/lib/sockets.c, 199
- snap-1.1-wjdb/lib/sockets.h, 200
- snap-1.1-wjdb/lib/timers.h, 202
- snap-1.1-wjdb/lib/unix.c, 205
- snap-1.1-wjdb/lib/version.h, 206
- snap-1.1-wjdb/lib/warn.h, 207
- snap-1.1-wjdb/lib/wassert.h, 208
- snap-1.1-wjdb/src/snap_demux_-
 - handler.c, 209
- snap-1.1-wjdb/src/snap_demux_-
 - handler.h, 214
- snap-1.1-wjdb/src/snap_exec.c, 217
- snap-1.1-wjdb/src/snap_-
 - sendandreceive.c, 222
- snap-1.1-wjdb/src/snapd.c, 227
- snap-1.1-wjdb/utils/consts.c, 228
- snap-1.1-wjdb/utils/kinject.c, 229
- snap-1.1-wjdb/utils/labels.c, 233
- snap-1.1-wjdb/utils/labels.h, 235
- snap-1.1-wjdb/utils/snapas.c, 237
- snap-1.1-wjdb/utils/snapdis.c, 242
- snap-1.1-wjdb/utils/snaplex.c, 244
- snap-1.1-wjdb/utils/snapparse.c,
 - 256
- snap-1.1-
 - wjdb/utils/snapparse.tab.h,
 - 278
- snap.h
 - IPPROTO_SNAP, 156
- SNAP_BUFLEN
 - snap_demux_handler.h, 214
- snap_bytecode.c
 - fprintf_addr, 158
 - fprintf_instr, 158
 - fprintf_opcode, 159
 - fprintf_value, 159
- fprintf_value_heap, 159
- fprintf_value_tag, 159
- GET_INT_HEAP, 158
- GET_OFFS_HEAP, 158
- GET_TAG_HEAP, 158
- SET_INT_HEAP, 158
- SET_OFFS_HEAP, 158
- SET_TAG_HEAP, 158
- sprintf_addr, 160
- snap_demux_buffer_noop
 - snap_demux_handler.c, 210
 - snap_demux_handler.h, 215
- snap_demux_buffer_print
 - snap_demux_handler.c, 210
 - snap_demux_handler.h, 215
- snap_demux_buffer_print_unsafe
 - snap_demux_handler.c, 210
 - snap_demux_handler.h, 215
- snap_demux_close
 - snap_demux_handler.c, 210
 - snap_demux_handler.h, 215
- snap_demux_close_rawip
 - snap_demux_handler.c, 210
- snap_demux_close_udp
 - snap_demux_handler.c, 210
- snap_demux_close_unix
 - snap_demux_handler.c, 210
- snap_demux_handler
 - snap_demux_handler.c, 211
 - snap_demux_handler.h, 216
- snap_demux_handler.c
 - fdset, 212
 - max_filedes, 212
 - protocols_internal, 212
 - snap_demux_buffer_noop, 210
 - snap_demux_buffer_print, 210
 - snap_demux_buffer_print_-
 - unsafe, 210
 - snap_demux_close, 210
 - snap_demux_close_rawip, 210
 - snap_demux_close_udp, 210
 - snap_demux_close_unix, 210
 - snap_demux_handler, 211
 - snap_demux_init, 211
 - snap_demux_init_rawip, 211
 - snap_demux_init_udp, 211
 - snap_demux_init_unix, 211
 - snap_demux_receive, 211
 - snap_demux_receivefrom, 212
 - snap_demux_select, 212

- socket_rawip, 212
- socket_udp, 213
- socket_unix, 213
- snap_demux_handler.h
 - buffer_handler, 215
 - receiveport, 216
 - SNAP_BUFLEN, 214
 - snap_demux_buffer_noop, 215
 - snap_demux_buffer_print, 215
 - snap_demux_buffer_print_-unsafe, 215
 - snap_demux_close, 215
 - snap_demux_handler, 216
 - snap_demux_init, 216
 - snap_demux_receive, 216
 - snap_demux_select, 216
 - SNAP_RAWIP, 214
 - SNAP_UDP, 215
 - SNAP_UNIX, 215
- snap_demux_init
 - snap_demux_handler.c, 211
 - snap_demux_handler.h, 216
- snap_demux_init_rawip
 - snap_demux_handler.c, 211
- snap_demux_init_udp
 - snap_demux_handler.c, 211
- snap_demux_init_unix
 - snap_demux_handler.c, 211
- snap_demux_receive
 - snap_demux_handler.c, 211
 - snap_demux_handler.h, 216
- snap_demux_receivefrom
 - snap_demux_handler.c, 212
- snap_demux_select
 - snap_demux_handler.c, 212
 - snap_demux_handler.h, 216
- snap_exec.c
 - basename, 219
 - compare_longints, 219
 - destaddr, 220
 - inbuf, 220
 - infd, 220
 - init_request, 219
 - IPPROTO_SNAP, 219
 - localaddr, 221
 - main, 220
 - NIPQUAD, 219
 - out_ttl, 221
 - parse_cmdline, 220
 - receiveport, 221
 - sd, 221
 - sendpkt, 220
 - srcaddr, 221
 - usage, 220
- snap_external_svclib_done
 - snap_svc.h, 294
 - snap_svc_if.c, 300
 - snap_svc_memmap.c, 305
 - snap_svc_proc.c, 318
 - snap_svc_route.c, 322
 - snap_svc_snmp.c, 325
 - snap_svc_TEMPLATE.c, 337
 - snap_svc_test.c, 340
- snap_external_svclib_free_local_-returnstruct
 - snap_svc.c, 290
 - snap_svc.h, 294
- snap_external_svclib_free_-returnstruct
 - snap_svc.c, 291
 - snap_svc.h, 295
- snap_external_svclib_getlastresult
 - snap_svc.c, 291
 - snap_svc.h, 295
- snap_external_svclib_getnextfunc
 - snap_svc.h, 295
 - snap_svc_if.c, 300
 - snap_svc_memmap.c, 306
 - snap_svc_proc.c, 318
 - snap_svc_route.c, 322
 - snap_svc_snmp.c, 325
 - snap_svc_TEMPLATE.c, 337
 - snap_svc_test.c, 340
- snap_external_svclib_init
 - snap_svc.h, 295
 - snap_svc_if.c, 300
 - snap_svc_memmap.c, 306
 - snap_svc_proc.c, 318
 - snap_svc_route.c, 322
 - snap_svc_snmp.c, 326
 - snap_svc_TEMPLATE.c, 337
 - snap_svc_test.c, 340
- snap_external_svclib_snmp_addvar_-null
 - snap_svc_snmp.c, 326
 - snap_svc_snmp.h, 332
- snap_external_svclib_snmp_addvar_-withvalue
 - snap_svc_snmp.c, 326
 - snap_svc_snmp.h, 332

- snap_external_svclib_snmp_close
 - snap_svc_snmp.c, 326
 - snap_svc_snmp.h, 333
- snap_external_svclib_snmp_execpdu
 - snap_svc_snmp.c, 326
 - snap_svc_snmp.h, 333
- snap_external_svclib_snmp_-
 - getallotherneighboursfromip
 - snap_svc_snmp.c, 326
 - snap_svc_snmp.h, 333
 - snap_external_svclib_snmp_gethop
 - snap_svc_snmp.c, 327
 - snap_svc_snmp.h, 333
 - snap_external_svclib_snmp_getiface
 - snap_svc_snmp.c, 327
 - snap_svc_snmp.h, 334
 - snap_external_svclib_snmp_-
 - getifnumber
 - snap_svc_snmp.c, 327
 - snap_svc_snmp.h, 334
 - snap_external_svclib_snmp_-
 - getnexthopfromip
 - snap_svc_snmp.c, 327
 - snap_svc_snmp.h, 334
 - snap_external_svclib_snmp_-
 - getnextiface
 - snap_svc_snmp.c, 328
 - snap_svc_snmp.h, 334
 - snap_external_svclib_snmp_-
 - getnextifacefromip
 - snap_svc_snmp.c, 328
 - snap_svc_snmp.h, 334
 - snap_external_svclib_snmp_-
 - getsingle
 - snap_svc_snmp.c, 328
 - snap_svc_snmp.h, 335
 - snap_external_svclib_snmp_init
 - snap_svc_snmp.c, 328
 - snap_svc_snmp.h, 335
 - snap_external_svclib_snmp_init_ip
 - snap_svc_snmp.c, 329
 - snap_svc_snmp.h, 335
 - snap_external_svclib_snmp_initpdu
 - snap_svc_snmp.c, 329
 - snap_svc_snmp.h, 335
 - snap_external_svclib_snmp_-
 - INTERNAL_execpdu_-
 - handler
 - snap_svc_snmp.c, 329
 - snap_external_svclib_snmp_-
 - isupiface
 - snap_svc_snmp.c, 329
 - snap_svc_snmp.h, 335
 - snap_external_svclib_snmp_setsingle
 - snap_svc_snmp.c, 329
 - snap_svc_snmp.h, 336
 - snap_external_svclib_testfunc
 - snap_svc_test.c, 341
 - snap_svc_test.h, 342
 - snap_external_svclib_testintfunc
 - snap_svc_test.c, 341
 - snap_svc_test.h, 342
 - snap_external_svclib_teststrfunc
 - snap_svc_test.c, 341
 - snap_svc_test.h, 342
 - snap_hashtable.c
 - hash_string, 161
 - ht_create, 161
 - ht_errno, 162
 - ht_insert, 161
 - ht_lookup, 162
 - ht_remove, 162
 - snap_htup, 36
 - n, 36
 - vals, 36
 - snap_hval, 37
 - s, 37
 - scalar, 37
 - t, 37
 - typetag, 37
 - v, 37
 - snap_internal_svclib_snmp_-
 - getnexthopfromip
 - snap_svc_snmp.c, 330
 - snap_interp.c
 - NIPQUAD, 164
 - snap_interp_packet
 - interp.h, 112
 - snap_io.c
 - ELSE_NOT_IN_FROM_-
 - HEAP, 166
 - file_to_str, 167
 - IF_IN_FROM_HEAP, 167
 - IN_TOSPACE_HEAP, 167
 - marshal_packet, 167
 - unmarshal_packet, 168
 - VERIFY, 167
 - snap_kern_iface.c
 - NIPQUAD, 171

- snap_kern_iface.h
 - _SNAP_KERN_IFACE, 172
- SNAP_LIB_H
 - libsnap.h, 122
- snap_list.c
 - cons, 173
 - free_list, 173
 - length_list, 173
- SNAP_RAWIP
 - snap_demux_handler.h, 214
- snap_receive
 - libsnap.c, 120
 - libsnap.h, 123
- snap_rcv_pkt
 - snapnet.c, 197
 - snapnet.h, 198
- snap_sendandreceive.c
 - basename, 224
 - compare_longints, 224
 - destaddr, 225
 - inbuf, 225
 - infd, 226
 - init_request, 225
 - IPPROTO_SNAP, 224
 - localaddr, 226
 - main, 225
 - NIPQUAD, 224
 - NO_RUNS, 224
 - out_ttl, 226
 - parse_cmdline, 225
 - receiveport, 226
 - sd, 226
 - sendpkt, 225
 - srcaddr, 226
 - usage, 225
- snap_svc.c
 - fini, 290
 - init, 290
 - snap_external_svclib_free_-
 - local_returnstruct, 290
 - snap_external_svclib_free_-
 - returnstruct, 291
 - snap_external_svclib_-
 - getlastresult, 291
- snap_svc.h
 - fini, 294
 - init, 294
 - snap_external_svclib_done, 294
 - snap_external_svclib_free_-
 - local_returnstruct, 294
 - snap_external_svclib_free_-
 - returnstruct, 295
 - snap_external_svclib_-
 - getlastresult, 295
 - snap_external_svclib_-
 - getnextfunc, 295
 - snap_external_svclib_init, 295
 - snap_svc_free_local_-
 - returnstruct, 293
 - snap_svc_getlastresult, 293
 - snap_svc_init, 293
 - snap_svc_register, 293
 - snapsvc_func_proto, 293
 - svc_fun_counter, 295
 - svc_return, 295
 - SVC_SNMP_TYPE_ADDR,
 - 294
 - SVC_SNMP_TYPE_INT, 294
 - SVC_SNMP_TYPE_LONG,
 - 294
 - SVC_SNMP_TYPE_NULL,
 - 294
 - SVC_SNMP_TYPE_STRING,
 - 294
- snap_svc/d_printf.c, 91
 - d_printf, 91
 - d_printf_timed, 92
 - debug_level, 92
 - debug_level_setp, 92
 - set_debug_level, 92
 - set_debug_level_int, 92
- snap_svc/d_printf.h, 96
 - d_printf, 96
 - d_printf_timed, 97
 - set_debug_level, 97
 - set_debug_level_int, 97
 - sysctl_snap_debug_level, 97
- snap_svc/snap_svc.c, 290
- snap_svc/snap_svc.h, 292
- snap_svc/snap_svc_if.c, 297
- snap_svc/snap_svc_if.h, 302
- snap_svc/snap_svc_memmap.c, 305
- snap_svc/snap_svc_memmap.h, 308
- snap_svc/snap_svc_memmap_-
 - hash.c, 310
- snap_svc/snap_svc_memmap_-
 - hash.h, 312
- snap_svc/snap_svc_memmap_hash_-
 - list.c, 314

- snap_svc/snap_svc_memmap_hash_-list.h, 315
- snap_svc/snap_svc_proc.c, 317
- snap_svc/snap_svc_proc.h, 319
- snap_svc/snap_svc_route.c, 320
- snap_svc/snap_svc_route.h, 323
- snap_svc/snap_svc_snmp.c, 324
- snap_svc/snap_svc_snmp.h, 332
- snap_svc/snap_svc_TEMPLATE.c, 337
- snap_svc/snap_svc_TEMPLATE.h, 339
- snap_svc/snap_svc_test.c, 340
- snap_svc/snap_svc_test.h, 342
- snap_svc_bind
 - snap_svc_library_handler.c, 180
 - snap_svc_library_handler.h, 183
- snap_svc_call_service
 - snap_svc_handler.h, 179
- snap_svc_close
 - snap_svc_library_handler.c, 180
 - snap_svc_library_handler.h, 183
- snap_svc_closemultiple
 - snap_svc_library_handler.c, 181
 - snap_svc_library_handler.h, 183
- snap_svc_conversion.c
 - snap_svc_convert_direct2stack, 175
 - snap_svc_convert_-returnstruct2stack, 175
 - snap_svc_convert_-stack2arguments, 175
 - snap_svc_convert_-stack2returnstruct, 175
- snap_svc_conversion.h
 - snap_svc_convert_direct2stack, 177
 - snap_svc_convert_-returnstruct2stack, 177
 - snap_svc_convert_-stack2arguments, 177
 - snap_svc_convert_-stack2returnstruct, 177
- snap_svc_convert_direct2stack
 - snap_svc_conversion.c, 175
 - snap_svc_conversion.h, 177
- snap_svc_convert_-returnstruct2stack
 - snap_svc_conversion.c, 175
 - snap_svc_conversion.h, 177
- snap_svc_convert_stack2arguments
 - snap_svc_conversion.c, 175
 - snap_svc_conversion.h, 177
- snap_svc_convert_-stack2returnstruct
 - snap_svc_conversion.c, 175
 - snap_svc_conversion.h, 177
- snap_svc_fileselector
 - snap_svc_library_handler.h, 182
- snap_svc_free_local_returnstruct
 - snap_svc.h, 293
- snap_svc_getlastresult
 - snap_svc.h, 293
- snap_svc_handler.h
 - snap_svc_call_service, 179
 - snap_svc_handler_close, 179
 - snap_svc_handler_init, 179
 - snap_svc_handler_reinit, 179
- snap_svc_handler_close
 - snap_svc_handler.h, 179
- snap_svc_handler_init
 - snap_svc_handler.h, 179
- snap_svc_handler_reinit
 - snap_svc_handler.h, 179
- snap_svc_if.c
 - if_get_interface_count, 298
 - if_get_interface_name, 298
 - if_getallneighbours, 298
 - if_gethopfromiface, 299
 - if_getiface, 299
 - if_getiface_up, 299
 - if_getifaceidx, 299
 - if_getnexthop, 299
 - if_getnextiface, 299
 - if_getoutiface, 299
 - if_setiface, 300
 - if_setiface_up, 300
 - iface_list, 301
 - snap_external_svclib_done, 300
 - snap_external_svclib_-getnextfunc, 300
 - snap_external_svclib_init, 300
 - snap_svc_if_count, 301
 - snap_svc_if_maxidx, 301
 - snap_svc_ifip_init, 300

- snap_svc_if.h
 - if_get_interface_count, 302
 - if_get_interface_name, 302
 - if_getallneighbours, 302
 - if_getiface, 303
 - if_getiface_up, 303
 - if_getnexthop, 303
 - if_setiface, 303
 - if_setiface_up, 303
 - snap_svc_ifip_init, 303
- snap_svc_if_count
 - snap_svc_if.c, 301
- snap_svc_if_maxidx
 - snap_svc_if.c, 301
- snap_svc_ifip_init
 - snap_svc_if.c, 300
 - snap_svc_if.h, 303
- snap_svc_ifip_item, 39
 - addr, 39
 - if_index, 39
 - if_name, 39
- snap_svc_init
 - snap_svc.h, 293
- snap_svc_library_handler.c
 - snap_svc_bind, 180
 - snap_svc_close, 180
 - snap_svc_closemultiple, 181
 - snap_svc_logerrors, 181
 - snap_svc_open, 181
 - snap_svc_openmultiple, 181
 - snap_svc_openmultiple_-selector_empty, 181
 - snap_svc_openmultiple_-selector_snapsvc, 181
- snap_svc_library_handler.h
 - snap_svc_bind, 183
 - snap_svc_close, 183
 - snap_svc_closemultiple, 183
 - snap_svc_fileselector, 182
 - snap_svc_logerrors, 183
 - snap_svc_open, 183
 - snap_svc_openmultiple, 183
 - snap_svc_openmultiple_-selector_empty, 183
 - snap_svc_openmultiple_-selector_snapsvc, 184
- snap_svc_logerrors
 - snap_svc_library_handler.c, 181
 - snap_svc_library_handler.h, 183
- snap_svc_memmap.c
 - mystrcmp, 305
 - snap_external_svclib_done, 305
 - snap_external_svclib_-getnextfunc, 306
 - snap_external_svclib_init, 306
 - snap_svc_memmap_add_string, 306
 - snap_svc_memmap_add_value, 306
 - snap_svc_memmap_del, 306
 - snap_svc_memmap_hashtable, 307
 - snap_svc_memmap_lookup_int, 306
 - snap_svc_memmap_lookup_-string, 307
- snap_svc_memmap.h
 - snap_svc_memmap_add_string, 308
 - snap_svc_memmap_add_value, 308
 - snap_svc_memmap_del, 308
 - snap_svc_memmap_lookup_int, 308
 - snap_svc_memmap_lookup_-string, 309
- snap_svc_memmap_add_string
 - snap_svc_memmap.c, 306
 - snap_svc_memmap.h, 308
- snap_svc_memmap_add_value
 - snap_svc_memmap.c, 306
 - snap_svc_memmap.h, 308
- snap_svc_memmap_del
 - snap_svc_memmap.c, 306
 - snap_svc_memmap.h, 308
- snap_svc_memmap_hash.c
 - hash_string, 310
 - ht_create, 310
 - ht_errno, 311
 - ht_insert, 311
 - ht_lookup, 311
 - ht_remove, 311
- snap_svc_memmap_hash.h
 - hash_string, 312
 - ht_create, 312
 - ht_errno, 313
 - ht_insert, 313
 - ht_lookup, 313
 - ht_remove, 313

- snap_svc_memmap_hash_list.c
 - cons, 314
 - free_list, 314
 - length_list, 314
- snap_svc_memmap_hash_list.h
 - cons, 316
 - free_list, 316
 - length_list, 316
 - list_t, 316
 - memalloc, 315
- snap_svc_memmap_hashtable
 - snap_svc_memmap.c, 307
- snap_svc_memmap_lookup_int
 - snap_svc_memmap.c, 306
 - snap_svc_memmap.h, 308
- snap_svc_memmap_lookup_string
 - snap_svc_memmap.c, 307
 - snap_svc_memmap.h, 309
- snap_svc_open
 - snap_svc_library_handler.c, 181
 - snap_svc_library_handler.h, 183
- snap_svc_openmultiple
 - snap_svc_library_handler.c, 181
 - snap_svc_library_handler.h, 183
- snap_svc_openmultiple_selector_-empty
 - snap_svc_library_handler.c, 181
 - snap_svc_library_handler.h, 183
- snap_svc_openmultiple_selector_-snapsvc
 - snap_svc_library_handler.c, 181
 - snap_svc_library_handler.h, 184
- snap_svc_proc.c
 - proc_sysnetip_getforwarding, 317
 - proc_sysnetip_setforwarding, 317
 - snap_external_svclib_done, 318
 - snap_external_svclib_-getnextfunc, 318
 - snap_external_svclib_init, 318
- snap_svc_proc.h
 - proc_sysnetip_getforwarding, 319
 - proc_sysnetip_setforwarding, 319
- snap_svc_rec, 40
 - nargs, 40
 - nret, 40
 - snapsvc_func, 40
- snap_svc_reg_handler.c
 - hDllList, 187
 - pReturnFree, 187
 - pReturnLast, 187
 - snap_svc_register_fini, 185
 - snap_svc_register_-freelaststruct, 186
 - snap_svc_register_init, 186
 - snap_svc_register_initialized, 187
 - snap_svc_register_-returnlaststruct, 186
 - snap_svc_registerall, 186
 - snap_svc_registeralllibs, 186
 - snap_svc_registerlib, 186
 - snap_svc_reregisterall, 186
 - snap_svc_unregisterall, 187
 - snap_svc_unregisteralllibs, 187
 - snap_svc_unregisterlib, 187
- snap_svc_reg_handler.h
 - snap_svc_register_fini, 190
 - snap_svc_register_-freelaststruct, 190
 - snap_svc_register_init, 190
 - snap_svc_register_-returnlaststruct, 190
 - snap_svc_registerall, 190
 - snap_svc_registeralllibs, 190
 - snap_svc_registerlib, 191
 - snap_svc_reregisterall, 191
 - snap_svc_unregisterall, 191
 - snap_svc_unregisteralllibs, 191
 - snap_svc_unregisterlib, 191
 - tDll, 190
 - tDllList, 190
- snap_svc_reg_table.c
 - mystrcmp, 192
 - snap_svc_table_add, 192
 - snap_svc_table_find, 193
 - snap_svc_table_fini, 193
 - snap_svc_table_init, 193
 - snap_svc_table_initialized, 193
- snap_svc_reg_table.h
 - DEF_SVC_TAB_SZ, 194
 - snap_svc_table_add, 195
 - snap_svc_table_find, 195

- snap_svc_table_fini, 195
 - snap_svc_table_init, 195
- snap_svc_register
 - snap_svc.h, 293
- snap_svc_register_fini
 - snap_svc_reg_handler.c, 185
 - snap_svc_reg_handler.h, 190
- snap_svc_register_freelaststruct
 - snap_svc_reg_handler.c, 186
 - snap_svc_reg_handler.h, 190
- snap_svc_register_init
 - snap_svc_reg_handler.c, 186
 - snap_svc_reg_handler.h, 190
- snap_svc_register_initialized
 - snap_svc_reg_handler.c, 187
- snap_svc_register_returnlaststruct
 - snap_svc_reg_handler.c, 186
 - snap_svc_reg_handler.h, 190
- snap_svc_registerall
 - snap_svc_reg_handler.c, 186
 - snap_svc_reg_handler.h, 190
- snap_svc_registeralllibs
 - snap_svc_reg_handler.c, 186
 - snap_svc_reg_handler.h, 190
- snap_svc_registerlib
 - snap_svc_reg_handler.c, 186
 - snap_svc_reg_handler.h, 191
- snap_svc_reregisterall
 - snap_svc_reg_handler.c, 186
 - snap_svc_reg_handler.h, 191
- snap_svc_route.c
 - handle_request, 321
 - ip_to_u32, 321
 - make_sockaddr_in, 321
 - printip, 321
 - route_add, 321
 - route_del, 321
 - snap_external_svclib_done, 322
 - snap_external_svclib_
 - getnextfunc, 322
 - snap_external_svclib_init, 322
- snap_svc_route.h
 - route_add, 323
 - route_del, 323
- snap_svc_snmp.c
 - callback_master_num, 330
 - snap_external_svclib_done, 325
 - snap_external_svclib_
 - getnextfunc, 325
 - snap_external_svclib_init, 326
- snap_external_svclib_snmp_
 - addvar_null, 326
- snap_external_svclib_snmp_
 - addvar_withvalue, 326
- snap_external_svclib_snmp_
 - close, 326
- snap_external_svclib_snmp_
 - execpdu, 326
- snap_external_svclib_snmp_
 - getallotherneighboursfromip, 326
- snap_external_svclib_snmp_
 - gethop, 327
- snap_external_svclib_snmp_
 - getiface, 327
- snap_external_svclib_snmp_
 - getifnumber, 327
- snap_external_svclib_snmp_
 - getnexthopfromip, 327
- snap_external_svclib_snmp_
 - getnextiface, 328
- snap_external_svclib_snmp_
 - getnextifacefromip, 328
- snap_external_svclib_snmp_
 - getsingle, 328
- snap_external_svclib_snmp_
 - init, 328
- snap_external_svclib_snmp_
 - init_ip, 329
- snap_external_svclib_snmp_
 - initpdu, 329
- snap_external_svclib_snmp_
 - INTERNAL_execpdu_
 - handler, 329
- snap_external_svclib_snmp_
 - isupiface, 329
- snap_external_svclib_snmp_
 - setsingle, 329
- snap_internal_svclib_snmp_
 - getnexthopfromip, 330
- svc_snmp_active, 330
- svc_snmp_pdu, 330
- svc_snmp_pdu_load, 330
- svc_snmp_pdu_waiting, 330
- svc_snmp_session, 331

- snap_svc_snmp.h
 - snap_external_svclib_snmp_
 - addvar_null, 332
 - snap_external_svclib_snmp_
 - addvar_withvalue, 332

- snap_external_svclib_snmp_-close, 333
 - snap_external_svclib_snmp_-execpdu, 333
 - snap_external_svclib_snmp_-getallotherneighboursfromip, 333
 - snap_external_svclib_snmp_-gethop, 333
 - snap_external_svclib_snmp_-getiface, 334
 - snap_external_svclib_snmp_-getifnumber, 334
 - snap_external_svclib_snmp_-getnexthopfromip, 334
 - snap_external_svclib_snmp_-getnextiface, 334
 - snap_external_svclib_snmp_-getnextifacefromip, 334
 - snap_external_svclib_snmp_-getsingle, 335
 - snap_external_svclib_snmp_-init, 335
 - snap_external_svclib_snmp_-init_ip, 335
 - snap_external_svclib_snmp_-initpdu, 335
 - snap_external_svclib_snmp_-isupiface, 335
 - snap_external_svclib_snmp_-setsingle, 336
- snap_svc_table_add
 - snap_svc_reg_table.c, 192
 - snap_svc_reg_table.h, 195
- snap_svc_table_find
 - snap_svc_reg_table.c, 193
 - snap_svc_reg_table.h, 195
- snap_svc_table_fini
 - snap_svc_reg_table.c, 193
 - snap_svc_reg_table.h, 195
- snap_svc_table_init
 - snap_svc_reg_table.c, 193
 - snap_svc_reg_table.h, 195
- snap_svc_table_initialized
 - snap_svc_reg_table.c, 193
- snap_svc_TEMPLATE.c
 - snap_external_svclib_done, 337
 - snap_external_svclib_-getnextfunc, 337
 - snap_external_svclib_init, 337
- snap_svc_test.c
 - snap_external_svclib_done, 340
 - snap_external_svclib_-getnextfunc, 340
 - snap_external_svclib_init, 340
 - snap_external_svclib_testfunc, 341
 - snap_external_svclib_-testintfunc, 341
 - snap_external_svclib_-teststrfunc, 341
- snap_svc_test.h
 - snap_external_svclib_testfunc, 342
 - snap_external_svclib_-testintfunc, 342
 - snap_external_svclib_-teststrfunc, 342
- snap_svc_unregisterall
 - snap_svc_reg_handler.c, 187
 - snap_svc_reg_handler.h, 191
- snap_svc_unregisteralllibs
 - snap_svc_reg_handler.c, 187
 - snap_svc_reg_handler.h, 191
- snap_svc_unregisterlib
 - snap_svc_reg_handler.c, 187
 - snap_svc_reg_handler.h, 191
- SNAP_UDP
 - snap_demux_handler.h, 215
- SNAP_UNIX
 - snap_demux_handler.h, 215
- SNAP_VERSION
 - version.h, 206
- snap_yy_input
 - snaplex.c, 253
- snapas.c
 - basename, 239
 - cbuf, 239
 - hbuf, 239
 - infilename, 239
 - main, 239
 - noop, 240
 - outfd, 240
 - outfilename, 240
 - p, 240
 - parse_cmdline, 239
 - pbuf, 240
 - sbuf, 240
 - usage, 239
 - yydebug, 240

- yyin, 240
- yyparse, 239
- snapd.c
 - main, 227
- snapdis.c
 - basename, 243
 - infd, 243
 - main, 243
 - outfile, 243
 - parse_cmdline, 243
 - usage, 243
- snaphdr, 41
 - code_sizeb, 41
 - daddr, 41
 - entry_point, 41
 - flags, 41
 - heap_sizeb, 41
 - saddr, 41
 - sport, 42
 - stack_sizeb, 42
 - version, 42
- snaplex.c
 - BEGIN, 247
 - conv_string, 253
 - ECHO, 247
 - EOB_ACT_CONTINUE_-
SCAN, 247
 - EOB_ACT_END_OF_FILE,
247
 - EOB_ACT_LAST_MATCH,
247
 - file, 253
 - FLEX_SCANNER, 247
 - INITIAL, 247
 - len, 254
 - lexbuf, 254
 - lexbuf_len, 254
 - lexbuf_pos, 254
 - read_from_file, 254
 - REJECT, 247
 - size, 254
 - snap_yy_input, 253
 - unput, 247
 - value_addr, 254
 - value_exc, 254
 - value_float, 254
 - value_int, 255
 - value_str, 255
 - YY_AT_BOL, 247
 - yy_bp, 255
 - YY_BREAK, 247
 - YY_BUF_SIZE, 248
 - YY_BUFFER_EOF_-
PENDING, 248
 - YY_BUFFER_NEW, 248
 - YY_BUFFER_NORMAL, 248
 - YY_BUFFER_STATE, 252
 - YY_CHAR, 252
 - YY_CURRENT_BUFFER,
248
 - YY_DECL, 248
 - YY_DO_BEFORE_ACTION,
248
 - YY_END_OF_BUFFER, 248
 - YY_END_OF_BUFFER_-
CHAR, 248
 - YY_EXIT_FAILURE, 249
 - YY_FATAL_ERROR, 249
 - YY_FLEX_MAJOR_-
VERSION, 249
 - YY_FLEX_MINOR_-
VERSION, 249
 - YY_FLUSH_BUFFER, 249
 - YY_INPUT, 249
 - YY_MORE_ADJ, 249
 - yy_new_buffer, 249
 - YY_NEW_FILE, 249
 - YY_NO_POP_STATE, 249
 - YY_NO_PUSH_STATE, 249
 - YY_NO_TOP_STATE, 250
 - YY_NULL, 250
 - YY_NUM_RULES, 250
 - YY_PROTO, 250, 253
 - YY_READ_BUF_SIZE, 250
 - YY_RESTORE_YY_MORE_-
OFFSET, 250
 - YY_RULE_SETUP, 250
 - YY_SC_TO_UI, 250
 - yy_set_bol, 250
 - yy_set_interactive, 251
 - yy_size_t, 252
 - YY_START, 251
 - YY_START_STACK_INCR,
251
 - YY_STATE_EOF, 251
 - yy_state_type, 253
 - yyconst, 251
 - yyin, 255
 - yylen, 255
 - yyless, 251, 252

- yymore, 252
- yyout, 255
- YYSTATE, 252
- yyterminate, 252
- yytext, 255
- yytext_ptr, 252
- snapnet.c
 - MAX_MTU, 196
 - pbuf, 197
 - snap_recv_pkt, 197
- snapnet.h
 - snap_recv_pkt, 198
- snapparse.c
 - CHECK_CODE_
OVERFLOW, 261
 - newho, 275
 - newtup, 275
 - noop, 276
 - p, 276
 - refine_op, 275
 - T_ADD, 261
 - T_ADDI, 261
 - T_ADDRV, 261
 - T_AND, 261
 - T_ANDI, 261
 - T_BCAST, 261
 - T_BCASTI, 261
 - T_BEZ, 261
 - T_BNE, 262
 - T_CALLS, 262
 - T_COMMA, 262
 - T_DATA, 262
 - T_DEMUX, 262
 - T_DEMUXI, 262
 - T_DFORW, 262
 - T_DFORWTO, 262
 - T_DIV, 262
 - T_DIVI, 262
 - T_DSEND, 262
 - T_EQ, 263
 - T_EQI, 263
 - T_EXCV, 263
 - T_EXIT, 263
 - T_FLOATV, 263
 - T_FORW, 263
 - T_FORWTO, 263
 - T_GEQ, 263
 - T_GEQI, 263
 - T_GETDST, 263
 - T_GETLD, 263
 - T_GETRB, 264
 - T_GETSPT, 264
 - T_GETSRC, 264
 - T_GT, 264
 - T_GTI, 264
 - T_HERE, 264
 - T_HOP, 264
 - T_INTV, 264
 - T_ISHERE, 264
 - T_ISTUP, 264
 - T_ISX, 264
 - T_JI, 265
 - T_LABEL, 265
 - T_LABELV, 265
 - T_LEN, 265
 - T_LEQ, 265
 - T_LEQI, 265
 - T_LNOT, 265
 - T_LPAREN, 265
 - T_LSHL, 265
 - T_LSHLI, 265
 - T_LT, 265
 - T_LTI, 266
 - T_MAIN, 266
 - T_MINUS, 266
 - T_MKTUP, 266
 - T_MOD, 266
 - T_MODI, 266
 - T_MULT, 266
 - T_MULTI, 266
 - T_NEG, 266
 - T_NEQ, 266
 - T_NEQI, 266
 - T_NOT, 267
 - T_NTH, 267
 - T_OR, 267
 - T_ORI, 267
 - T_PAJ, 267
 - T_PC, 267
 - T_PLUS, 267
 - T_POP, 267
 - T_POPI, 267
 - T_PRINT, 267
 - T_PULL, 267
 - T_PULLSTACK, 268
 - T_PUSH, 268
 - T_RAISEX, 268
 - T_ROUTE, 268
 - T_RPAREN, 268
 - T_RSHA, 268

-
- T_RSHAI, 268
 - T_RSHL, 268
 - T_RSHLI, 268
 - T_RTDEV, 268
 - T_SEND, 268
 - T_SETXH, 269
 - T_SNET, 269
 - T_SNETI, 269
 - T_STACKCOUNT, 269
 - T_STACKEMPTY, 269
 - T_STORE, 269
 - T_STRV, 269
 - T_SUB, 269
 - T_SUBI, 269
 - T_SVCV, 269
 - T_TPAJ, 269
 - value_addr, 277
 - value_exc, 277
 - value_float, 277
 - value_int, 277
 - value_str, 277
 - YY_DECL_NON_LSP_-
 VARIABLES, 270
 - YY_DECL_VARIABLES, 270
 - YYABORT, 270
 - YYACCEPT, 270
 - YYBACKUP, 270
 - YYBISON, 271
 - yyclearin, 271
 - YYCOPY, 271
 - YYDEBUG, 271
 - YYDPRINTF, 271
 - YYEMPTY, 271
 - YYEOF, 271
 - YYERRCODE, 272
 - yyerrok, 272
 - YYERROR, 272
 - yyerror, 276
 - YYFAIL, 272
 - YYFINAL, 272
 - YYFLAG, 272
 - YYINITDEPTH, 272
 - YYLAST, 272
 - YYLEX, 272
 - YYLOC_DEFAULT, 273
 - YYMAXDEPTH, 273
 - YYNTBASE, 273
 - yyparse, 276
 - YYPARSE_PARAM_ARG,
 273
 - YYPARSE_PARAM_DECL,
 273
 - YYPOPSTACK, 273
 - YYRECOVERING, 273
 - YYSIZE_T, 273
 - YYSTACK_ALLOC, 274
 - YYSTACK_BYTES, 274
 - YYSTACK_FREE, 274
 - YYSTACK_GAP_MAX, 274
 - YYSTACK_RELOCATE, 274
 - YYSTYPE, 274
 - YYSTYPE_IS_TRIVIAL, 275
 - YYTERROR, 275
 - YYTRANSLATE, 275
 - snapparse.tab.h
 - T_ADD, 280
 - T_ADDI, 280
 - T_ADDRV, 280
 - T_AND, 280
 - T_ANDI, 280
 - T_BCAST, 280
 - T_BCASTI, 280
 - T_BEZ, 281
 - T_BNE, 281
 - T_CALLS, 281
 - T_COMMA, 281
 - T_DATA, 281
 - T_DEMUX, 281
 - T_DEMUXI, 281
 - T_DFORW, 281
 - T_DFORWTO, 281
 - T_DIV, 281
 - T_DIVI, 281
 - T_DSEND, 282
 - T_EQ, 282
 - T_EQI, 282
 - T_EXCV, 282
 - T_EXIT, 282
 - T_FLOATV, 282
 - T_FORW, 282
 - T_FORWTO, 282
 - T_GEQ, 282
 - T_GEQI, 282
 - T_GETDST, 282
 - T_GETLD, 283
 - T_GETRB, 283
 - T_GETSPT, 283
 - T_GETSRC, 283
 - T_GT, 283
 - T_GTI, 283
-

T_HERE, 283
T_HOP, 283
T_INTV, 283
T_ISHERE, 283
T_ISTUP, 283
T_ISX, 284
T_JI, 284
T_LABEL, 284
T_LABELV, 284
T_LEN, 284
T_LEQ, 284
T_LEQI, 284
T_LNOT, 284
T_LPAREN, 284
T_LSHL, 284
T_LSHLI, 284
T_LT, 285
T_LTI, 285
T_MAIN, 285
T_MINUS, 285
T_MKTUP, 285
T_MOD, 285
T_MODI, 285
T_MULT, 285
T_MULTI, 285
T_NEG, 285
T_NEQ, 285
T_NEQI, 286
T_NOT, 286
T_NTH, 286
T_OR, 286
T_ORI, 286
T_PAJ, 286
T_PC, 286
T_PLUS, 286
T_POP, 286
T_POPI, 286
T_PRINT, 286
T_PULL, 287
T_PUSH, 287
T_RAISEX, 287
T_ROUTE, 287
T_RPAREN, 287
T_RSHA, 287
T_RSHAI, 287
T_RSHL, 287
T_RSHLI, 287
T_RTDEV, 287
T_SEND, 287
T_SETXH, 288

T_SNET, 288
T_SNETI, 288
T_STORE, 288
T_STRV, 288
T_SUB, 288
T_SUBI, 288
T_SVCV, 288
T_TPAJ, 288
yylval, 289
YYSTYPE, 288

snapsvc.func
 snap_svc_rec, 40
snapsvc.func.proto
 snap_svc.h, 293
SNET
 bytecode.h, 81
SNETI
 bytecode.h, 81
socket_rawip
 snap_demux_handler.c, 212
socket_udp
 snap_demux_handler.c, 213
socket_unix
 snap_demux_handler.c, 213
sockets.c
 skfd, 199
 sockets_open, 199
sockets.h
 ax25_sock, 200
 ddp_sock, 200
 ec_sock, 200
 inet6_sock, 201
 inet_sock, 201
 ipx_sock, 201
 rose_sock, 201
 skfd, 201
 sockets_open, 200
sockets_open
 sockets.c, 199
 sockets.h, 200
sp
 packet_t, 30
sport
 snaphdr, 42
sprint
 aftype, 12
 hwtype, 20
sprintf_addr
 snap_bytecode.c, 160
srcaddr

- snap_exec.c, 221
 - snap_sendandreceive.c, 226
- stack_max
 - packet_t, 30
- stack_min
 - packet_t, 30
- stack_sizeb
 - consts.c, 228
 - consts.h, 88
 - snaphdr, 42
- STACKCOUNT
 - bytecode.h, 81
- STACKEMPTY
 - bytecode.h, 82
- stats
 - interface, 24
- STORE
 - bytecode.h, 82
- STRV
 - bytecode.h, 82
- SUB
 - bytecode.h, 82
- SUBI
 - bytecode.h, 82
- svc_fun_counter
 - snap_svc.h, 295
- svc_return
 - snap_svc.h, 295
- svc_returnitem, 43
 - data, 43
 - length, 43
 - oid, 43
 - oid_length, 43
 - type, 44
- svc_returnstruct, 45
 - length, 45
 - list, 45
- svc_snmp_active
 - snap_svc_snmp.c, 330
- svc_snmp_pdu
 - snap_svc_snmp.c, 330
- svc_snmp_pdu_load
 - snap_svc_snmp.c, 330
- svc_snmp_pdu_waiting
 - snap_svc_snmp.c, 330
- svc_snmp_session
 - snap_svc_snmp.c, 331
- SVC_SNMP_TYPE_ADDR
 - snap_svc.h, 294
- SVC_SNMP_TYPE_INT
 - snap_svc.h, 294
- SVC_SNMP_TYPE_LONG
 - snap_svc.h, 294
- SVC_SNMP_TYPE_NULL
 - snap_svc.h, 294
- SVC_SNMP_TYPE_STRING
 - snap_svc.h, 294
- SVCV
 - bytecode.h, 82
- sysctl_snap_debug_level
 - snap-1.1-wjdb/lib/d_printf.h, 95
 - snap_svc/d_printf.h, 97
- t
 - snap_hval, 37
- T_ADD
 - snapparse.c, 261
 - snapparse.tab.h, 280
- T_ADDI
 - snapparse.c, 261
 - snapparse.tab.h, 280
- T_ADDRV
 - snapparse.c, 261
 - snapparse.tab.h, 280
- T_AND
 - snapparse.c, 261
 - snapparse.tab.h, 280
- T_ANDI
 - snapparse.c, 261
 - snapparse.tab.h, 280
- T_BCAST
 - snapparse.c, 261
 - snapparse.tab.h, 280
- T_BCASTI
 - snapparse.c, 261
 - snapparse.tab.h, 280
- T_BEZ
 - snapparse.c, 261
 - snapparse.tab.h, 281
- T_BNE
 - snapparse.c, 262
 - snapparse.tab.h, 281
- T_CALLS
 - snapparse.c, 262
 - snapparse.tab.h, 281
- T_COMMA
 - snapparse.c, 262
 - snapparse.tab.h, 281
- T_DATA

- snapparse.c, 262
- snapparse.tab.h, 281
- T_DEMUX
 - snapparse.c, 262
 - snapparse.tab.h, 281
- T_DEMUXI
 - snapparse.c, 262
 - snapparse.tab.h, 281
- T_DFORW
 - snapparse.c, 262
 - snapparse.tab.h, 281
- T_DFORWTO
 - snapparse.c, 262
 - snapparse.tab.h, 281
- T_DIV
 - snapparse.c, 262
 - snapparse.tab.h, 281
- T_DIVI
 - snapparse.c, 262
 - snapparse.tab.h, 281
- T_DSEND
 - snapparse.c, 262
 - snapparse.tab.h, 282
- T_EQ
 - snapparse.c, 263
 - snapparse.tab.h, 282
- T_EQI
 - snapparse.c, 263
 - snapparse.tab.h, 282
- T_EXCV
 - snapparse.c, 263
 - snapparse.tab.h, 282
- T_EXIT
 - snapparse.c, 263
 - snapparse.tab.h, 282
- T_FLOATV
 - snapparse.c, 263
 - snapparse.tab.h, 282
- T_FORW
 - snapparse.c, 263
 - snapparse.tab.h, 282
- T_FORWTO
 - snapparse.c, 263
 - snapparse.tab.h, 282
- T_GEQ
 - snapparse.c, 263
 - snapparse.tab.h, 282
- T_GEQI
 - snapparse.c, 263
 - snapparse.tab.h, 282
- T_GETDST
 - snapparse.c, 263
 - snapparse.tab.h, 282
- T_GETLD
 - snapparse.c, 263
 - snapparse.tab.h, 283
- T_GETRB
 - snapparse.c, 264
 - snapparse.tab.h, 283
- T_GETSPT
 - snapparse.c, 264
 - snapparse.tab.h, 283
- T_GETSRC
 - snapparse.c, 264
 - snapparse.tab.h, 283
- T_GT
 - snapparse.c, 264
 - snapparse.tab.h, 283
- T_GTI
 - snapparse.c, 264
 - snapparse.tab.h, 283
- T_HERE
 - snapparse.c, 264
 - snapparse.tab.h, 283
- T_HOP
 - snapparse.c, 264
 - snapparse.tab.h, 283
- T_INTV
 - snapparse.c, 264
 - snapparse.tab.h, 283
- T_ISHERE
 - snapparse.c, 264
 - snapparse.tab.h, 283
- T_ISTUP
 - snapparse.c, 264
 - snapparse.tab.h, 283
- T_ISX
 - snapparse.c, 264
 - snapparse.tab.h, 284
- T_JI
 - snapparse.c, 265
 - snapparse.tab.h, 284
- T_LABEL
 - snapparse.c, 265
 - snapparse.tab.h, 284
- T_LABELV
 - snapparse.c, 265
 - snapparse.tab.h, 284
- T_LEN
 - snapparse.c, 265

- snapparse.tab.h, 284
- T_LEQ
 - snapparse.c, 265
 - snapparse.tab.h, 284
- T_LEQI
 - snapparse.c, 265
 - snapparse.tab.h, 284
- T_LNOT
 - snapparse.c, 265
 - snapparse.tab.h, 284
- T_LPAREN
 - snapparse.c, 265
 - snapparse.tab.h, 284
- T_LSHL
 - snapparse.c, 265
 - snapparse.tab.h, 284
- T_LSHLI
 - snapparse.c, 265
 - snapparse.tab.h, 284
- T_LT
 - snapparse.c, 265
 - snapparse.tab.h, 285
- T_LTI
 - snapparse.c, 266
 - snapparse.tab.h, 285
- T_MAIN
 - snapparse.c, 266
 - snapparse.tab.h, 285
- T_MINUS
 - snapparse.c, 266
 - snapparse.tab.h, 285
- T_MKTUP
 - snapparse.c, 266
 - snapparse.tab.h, 285
- T_MOD
 - snapparse.c, 266
 - snapparse.tab.h, 285
- T_MODI
 - snapparse.c, 266
 - snapparse.tab.h, 285
- T_MULT
 - snapparse.c, 266
 - snapparse.tab.h, 285
- T_MULTI
 - snapparse.c, 266
 - snapparse.tab.h, 285
- T_NEG
 - snapparse.c, 266
 - snapparse.tab.h, 285
- T_NEQ
 - snapparse.c, 266
 - snapparse.tab.h, 285
- T_NEQI
 - snapparse.c, 266
 - snapparse.tab.h, 286
- T_NOT
 - snapparse.c, 267
 - snapparse.tab.h, 286
- T_NTH
 - snapparse.c, 267
 - snapparse.tab.h, 286
- T_OR
 - snapparse.c, 267
 - snapparse.tab.h, 286
- T_ORI
 - snapparse.c, 267
 - snapparse.tab.h, 286
- T_PAJ
 - snapparse.c, 267
 - snapparse.tab.h, 286
- T_PC
 - snapparse.c, 267
 - snapparse.tab.h, 286
- T_PLUS
 - snapparse.c, 267
 - snapparse.tab.h, 286
- T_POP
 - snapparse.c, 267
 - snapparse.tab.h, 286
- T_POPI
 - snapparse.c, 267
 - snapparse.tab.h, 286
- T_PRINT
 - snapparse.c, 267
 - snapparse.tab.h, 286
- T_PULL
 - snapparse.c, 267
 - snapparse.tab.h, 287
- T_PULLSTACK
 - snapparse.c, 268
- T_PUSH
 - snapparse.c, 268
 - snapparse.tab.h, 287
- T_RAISEX
 - snapparse.c, 268
 - snapparse.tab.h, 287
- T_ROUTE
 - snapparse.c, 268
 - snapparse.tab.h, 287
- T_RPAREN

- snapparse.c, 268
- snapparse.tab.h, 287
- T_RSHA
 - snapparse.c, 268
 - snapparse.tab.h, 287
- T_RSHAI
 - snapparse.c, 268
 - snapparse.tab.h, 287
- T_RSHL
 - snapparse.c, 268
 - snapparse.tab.h, 287
- T_RSHLI
 - snapparse.c, 268
 - snapparse.tab.h, 287
- T_RTDEV
 - snapparse.c, 268
 - snapparse.tab.h, 287
- T_SEND
 - snapparse.c, 268
 - snapparse.tab.h, 287
- T_SETXH
 - snapparse.c, 269
 - snapparse.tab.h, 288
- T_SNET
 - snapparse.c, 269
 - snapparse.tab.h, 288
- T_SNETI
 - snapparse.c, 269
 - snapparse.tab.h, 288
- T_STACKCOUNT
 - snapparse.c, 269
- T_STACKEMPTY
 - snapparse.c, 269
- T_STORE
 - snapparse.c, 269
 - snapparse.tab.h, 288
- T_STRV
 - snapparse.c, 269
 - snapparse.tab.h, 288
- T_SUB
 - snapparse.c, 269
 - snapparse.tab.h, 288
- T_SUBI
 - snapparse.c, 269
 - snapparse.tab.h, 288
- T_SVCV
 - snapparse.c, 269
 - snapparse.tab.h, 288
- T_TPAJ
 - snapparse.c, 269
 - snapparse.tab.h, 288
- tab
 - hash_table_t, 16
- tab_sz
 - hash_table_t, 17
- TAG_T
 - bytecode.h, 82
- TAGSZ
 - bytecode.h, 83
- tDll
 - snap_svc_reg_handler.h, 190
- tDllList
 - snap_svc_reg_handler.h, 190
- timers.h
 - do_print_antitimers, 203
 - do_print_individual_timers, 203
 - do_print_item_messages, 204
 - dump_all_timers, 203
 - init_all_timers, 203
 - internal_print_anti_time, 203
 - internal_print_time, 203
 - print_anti_mtimer, 202
 - print_anti_timer, 202
 - print_flag_count, 204
 - print_flags, 204
 - print_mtimer, 203
 - print_timer, 203
- title
 - aftype, 12
 - hwtype, 20
- TPAJ
 - bytecode.h, 83
- tr_hwtype
 - hw.c, 107
- tunnel_hwtype
 - hw.c, 107
- TUPLEV
 - bytecode.h, 83
- tx_aborted_errors
 - user_net_device_stats, 47
- tx_bytes
 - user_net_device_stats, 47
- tx_carrier_errors
 - user_net_device_stats, 48
- tx_compressed
 - user_net_device_stats, 48
- tx_dropped
 - user_net_device_stats, 48
- tx_errors
 - user_net_device_stats, 48

- tx_fifo_errors
 - user_net_device_stats, 48
- tx_heartbeat_errors
 - user_net_device_stats, 48
- tx_packets
 - user_net_device_stats, 48
- tx_queue_len
 - interface, 25
- tx_window_errors
 - user_net_device_stats, 48
- type
 - hwtype, 20
 - interface, 25
 - svc.returnitem, 44
- typetag
 - snap_hval, 37
- udpaddr
 - libsnap.c, 121
- UDPPORT
 - libsnap.c, 119
- udpport
 - kinject.c, 232
- uint32
 - config.h, 86
- unix.c
 - unspec_aftype, 205
- unix_aftype
 - af.c, 57
- unmarshal_packet
 - io.h, 115
 - snap_io.c, 168
- unput
 - snaplex.c, 247
- unspec_aftype
 - af.c, 57
 - unix.c, 205
- unspec_hwtype
 - hw.c, 107
 - loopback.c, 126
- usage
 - kinject.c, 231
 - libsnap.c, 120
 - snap_exec.c, 220
 - snap_sendandreceive.c, 225
 - snapas.c, 239
 - snapdis.c, 243
- user_net_device_stats, 46
 - collisions, 46
 - rx_bytes, 46
 - rx_compressed, 46
 - rx_crc_errors, 46
 - rx_dropped, 47
 - rx_errors, 47
 - rx_fifo_errors, 47
 - rx_frame_errors, 47
 - rx_length_errors, 47
 - rx_missed_errors, 47
 - rx_multicast, 47
 - rx_over_errors, 47
 - rx_packets, 47
 - tx_aborted_errors, 47
 - tx_bytes, 47
 - tx_carrier_errors, 48
 - tx_compressed, 48
 - tx_dropped, 48
 - tx_errors, 48
 - tx_fifo_errors, 48
 - tx_heartbeat_errors, 48
 - tx_packets, 48
 - tx_window_errors, 48
- v
 - l, 26
 - snap_hval, 37
- vals
 - snap_htup, 36
- value
 - pair_t, 31
- value_addr
 - snaplex.c, 254
 - snapparse.c, 277
- value_exc
 - snaplex.c, 254
 - snapparse.c, 277
- value_float
 - snaplex.c, 254
 - snapparse.c, 277
- value_int
 - snaplex.c, 255
 - snapparse.c, 277
- value_str
 - snaplex.c, 255
 - snapparse.c, 277
- value_t
 - bytecode.h, 83
- VERIFY
 - snap_io.c, 167
- version
 - snaphdr, 42

- version.h
 - SNAP_VERSION, 206
- warn
 - warn.h, 207
- warn.h
 - warn, 207
- wassert
 - wassert.h, 208
- wassert.h
 - wassert, 208
- where
 - label_mapping_t, 27
- XOR
 - bytecode.h, 83
- XORI
 - bytecode.h, 83
- YY_AT_BOL
 - snaplex.c, 247
- yy_at_bol
 - yy_buffer_state, 49
- yy_bp
 - snaplex.c, 255
- YY_BREAK
 - snaplex.c, 247
- yy_buf_pos
 - yy_buffer_state, 49
- YY_BUF_SIZE
 - snaplex.c, 248
- yy_buf_size
 - yy_buffer_state, 49
- YY_BUFFER_EOF_PENDING
 - snaplex.c, 248
- YY_BUFFER_NEW
 - snaplex.c, 248
- YY_BUFFER_NORMAL
 - snaplex.c, 248
- YY_BUFFER_STATE
 - snaplex.c, 252
- yy_buffer_state, 49
 - yy_at_bol, 49
 - yy_buf_pos, 49
 - yy_buf_size, 49
 - yy_buffer_status, 49
 - yy_ch_buf, 49
 - yy_fill_buffer, 49
 - yy_input_file, 49
 - yy_is_interactive, 49
 - yy_is_our_buffer, 50
 - yy_n_chars, 50
 - yy_new_buffer, 49
- yy_is_our_buffer, 50
- yy_n_chars, 50
- yy_buffer_status
 - yy_buffer_state, 49
- yy_ch_buf
 - yy_buffer_state, 49
- YY_CHAR
 - snaplex.c, 252
- YY_CURRENT_BUFFER
 - snaplex.c, 248
- YY_DECL
 - snaplex.c, 248
- YY_DECL_NON_LSP_-
 - VARIABLES
 - snapparse.c, 270
- YY_DECL_VARIABLES
 - snapparse.c, 270
- YY_DO_BEFORE_ACTION
 - snaplex.c, 248
- YY_END_OF_BUFFER
 - snaplex.c, 248
- YY_END_OF_BUFFER_CHAR
 - snaplex.c, 248
- YY_EXIT_FAILURE
 - snaplex.c, 249
- YY_FATAL_ERROR
 - snaplex.c, 249
- yy_fill_buffer
 - yy_buffer_state, 49
- YY_FLEX_MAJOR_VERSION
 - snaplex.c, 249
- YY_FLEX_MINOR_VERSION
 - snaplex.c, 249
- YY_FLUSH_BUFFER
 - snaplex.c, 249
- YY_INPUT
 - snaplex.c, 249
- yy_input_file
 - yy_buffer_state, 49
- yy_is_interactive
 - yy_buffer_state, 49
- yy_is_our_buffer
 - yy_buffer_state, 50
- YY_MORE_ADJ
 - snaplex.c, 249
- yy_n_chars
 - yy_buffer_state, 50
- yy_new_buffer
 - snaplex.c, 249
- YY_NEW_FILE

- snaplex.c, 249
- YY_NO_POP_STATE
 - snaplex.c, 249
- YY_NO_PUSH_STATE
 - snaplex.c, 249
- YY_NO_TOP_STATE
 - snaplex.c, 250
- YY_NULL
 - snaplex.c, 250
- YY_NUM_RULES
 - snaplex.c, 250
- YY_PROTO
 - snaplex.c, 250, 253
- YY_READ_BUF_SIZE
 - snaplex.c, 250
- YY_RESTORE_YY_MORE_-
 - OFFSET
 - snaplex.c, 250
- YY_RULE_SETUP
 - snaplex.c, 250
- YY_SC_TO_UI
 - snaplex.c, 250
- yy_set_bol
 - snaplex.c, 250
- yy_set_interactive
 - snaplex.c, 251
- yy_size_t
 - snaplex.c, 252
- YY_START
 - snaplex.c, 251
- YY_START_STACK_INCR
 - snaplex.c, 251
- YY_STATE_EOF
 - snaplex.c, 251
- yy_state_type
 - snaplex.c, 253
- YYABORT
 - snapparse.c, 270
- YYACCEPT
 - snapparse.c, 270
- yyalloc, 51
 - yyss, 51
 - yyvs, 51
- YYBACKUP
 - snapparse.c, 270
- YYBISON
 - snapparse.c, 271
- yyclearin
 - snapparse.c, 271
- yyconst
 - snaplex.c, 251
- YYCOPY
 - snapparse.c, 271
- YYDEBUG
 - snapparse.c, 271
- yydebug
 - snapas.c, 240
- YYDPRINTF
 - snapparse.c, 271
- YYEMPTY
 - snapparse.c, 271
- YYEOF
 - snapparse.c, 271
- YYERRCODE
 - snapparse.c, 272
- yyerrok
 - snapparse.c, 272
- YYERROR
 - snapparse.c, 272
- yyerror
 - snapparse.c, 276
- YYFAIL
 - snapparse.c, 272
- YYFINAL
 - snapparse.c, 272
- YYFLAG
 - snapparse.c, 272
- yyin
 - snapas.c, 240
 - snaplex.c, 255
- YYINITDEPTH
 - snapparse.c, 272
- YYLAST
 - snapparse.c, 272
- yyleng
 - snaplex.c, 255
- yyless
 - snaplex.c, 251, 252
- YYLEX
 - snapparse.c, 272
- YYLOC_DEFAULT
 - snapparse.c, 273
- yylval
 - snapparse.tab.h, 289
- YYMAXDEPTH
 - snapparse.c, 273
- yymore
 - snaplex.c, 252
- YYNTBASE
 - snapparse.c, 273

- yyout
 - snaplex.c, 255
- yyvsparse
 - snapas.c, 239
 - snapparse.c, 276
- YYPARSE_PARAM_ARG
 - snapparse.c, 273
- YYPARSE_PARAM_DECL
 - snapparse.c, 273
- YYPOPSTACK
 - snapparse.c, 273
- YYRECOVERING
 - snapparse.c, 273
- YYSIZE_T
 - snapparse.c, 273
- yyss
 - yyalloc, 51
- YYSTACK_ALLOC
 - snapparse.c, 274
- YYSTACK_BYTES
 - snapparse.c, 274
- YYSTACK_FREE
 - snapparse.c, 274
- YYSTACK_GAP_MAX
 - snapparse.c, 274
- YYSTACK_RELOCATE
 - snapparse.c, 274
- YYSTATE
 - snaplex.c, 252
- YYSTYPE
 - snapparse.c, 274
 - snapparse.tab.h, 288
- yystate, 52
 - integer, 52
 - ptr, 52
- YYSTYPE_IS_TRIVIAL
 - snapparse.c, 275
- yyterminate
 - snaplex.c, 252
- YYTERROR
 - snapparse.c, 275
- yytext
 - snaplex.c, 255
- yytext_ptr
 - snaplex.c, 252
- YYTRANSLATE
 - snapparse.c, 275
- yyvs
 - yyalloc, 51
- ZERO_VALUE_T
 - bytecode.h, 83