NAME

pcap_activate − activate a capture handle

SYNOPSIS

#include <pcap/pcap.h>

int pcap_activate(pcap_t *p);

DESCRIPTION

pcap_activate() is used to activate a packet capture handle to look at packets on the network, with the options that were set on the handle being in effect.

RETURN VALUE

pcap_activate() returns **0** on success without warnings, a non-zero positive value on success with warnings, and a negative value on error. A non-zero return value indicates what warning or error condition occurred.

The possible warning values are:

PCAP_WARNING_PROMISC_NOTSUP
Promiscuous mode was requested, but the capture source doesn't support promiscuous mode.

PCAP_WARNING_TSTAMP_TYPE_NOTSUP
The time stamp type specified in a previous **pcap_set_tstamp_type**(3PCAP) call isn't supported by the capture source (the time stamp type is left as the default),

PCAP_WARNING
Another warning condition occurred; **pcap_geterr**(3PCAP) or **pcap_perror**(3PCAP) may be called with *p* as an argument to fetch or display a message describing the warning condition.

The possible error values are:

PCAP_ERROR_ACTIVATED
The handle has already been activated.

PCAP_ERROR_NO_SUCH_DEVICE
The capture source specified when the handle was created doesn't exist.

PCAP_ERROR_PERM_DENIED
The process doesn't have permission to open the capture source.

PCAP_ERROR_PROMISC_PERM_DENIED
The process has permission to open the capture source but doesn't have permission to put it into promiscuous mode.

PCAP_ERROR_RFMON_NOTSUP
Monitor mode was specified but the capture source doesn't support monitor mode.

PCAP_ERROR_IFACE_NOT_UP
The capture source device is not up.

PCAP_ERROR
Another error occurred. **pcap_geterr**() or **pcap_perr or**() may be called with *p* as an argument to fetch or display a message describing the error.

If **PCAP_WARNING_PROMISC_NOTSUP**, **PCAP_ERROR_NO_SUCH_DEVICE**, or **PCAP_ERROR_PERM_DENIED** is returned, **pcap_geterr**() or **pcap_perror**() may be called with *p* as an argument to fetch or display an message giving additional details about the problem that might be useful for debugging the problem if it's unexpected.

Additional warning and error codes may be added in the future; a program should check for positive, negative, and zero return codes, and treat all positive return codes as warnings and all negative return codes as errors. **pcap_statustostr**(3PCAP) can be called, with a warning or error code as an argument, to fetch a message describing the warning or error code.

If **pcap_activate**() fails, the *pcap_t ** is not closed and freed; it should be closed using **pcap_close**().

SEE ALSO

pcap(3PCAP)