

NAME

`pcap_init` – initialize the library

SYNOPSIS

```
#include <pcap/pcap.h>
```

```
char errbuf[PCAP_ERRBUF_SIZE];
```

```
int pcap_init(unsigned int opts, char *errbuf);
```

DESCRIPTION

`pcap_init()` is used to initialize the Packet Capture library. *opts* specifies options for the library; currently, the options are:

PCAP_CHAR_ENC_LOCAL

Treat all strings supplied as arguments, and return all strings to the caller, as being in the local character encoding.

PCAP_CHAR_ENC_UTF_8

Treat all strings supplied as arguments, and return all strings to the caller, as being in UTF-8.

On UNIX-like systems, the local character encoding is assumed to be UTF-8, so no character encoding transformations are done.

On Windows, the local character encoding is the local ANSI code page.

If `pcap_init()` is not called, strings are treated as being in the local ANSI code page on Windows, `pcap_lookupdev(3PCAP)` will succeed if there is a device on which to capture, and `pcap_create(3PCAP)` makes an attempt to check whether the string passed as an argument is a UTF-16LE string - note that this attempt is unsafe, as it may run past the end of the string - to handle `pcap_lookupdev()` returning a UTF-16LE string. Programs that don't call `pcap_init()` should, on Windows, call `pcap_wsockinit()` to initialize Winsock; this is not necessary if `pcap_init()` is called, as `pcap_init()` will initialize Winsock itself on Windows.

RETURN VALUE

`pcap_init()` returns **0** on success and **-1** on failure. If **-1** is returned, *errbuf* is filled in with an appropriate error message. *errbuf* is assumed to be able to hold at least **PCAP_ERRBUF_SIZE** chars.

BACKWARD COMPATIBILITY

This function became available in libpcap release 1.10.0. In previous releases, on Windows, all strings supplied as arguments, and all strings returned to the caller, are in the local character encoding.

SEE ALSO

`pcap(3PCAP)`