

NAME

Mon::SNMP – decode SNMP trap

SYNOPSIS

```
use Mon::SNMP;

$trap = new Mon::SNMP;

$trap->buffer($snmptrap);

%traphash = $trap->decode;

$error = $trap->error;
```

DESCRIPTION

Mon::SNMP provides methods to decode SNMP trap PDUs. It is based on Graham Barr's Convert::BER module, and its purpose is to provide SNMP trap handling to "mon".

It is not complete, so don't bother trying to use it unless you are ready to debug and write some code.

METHODS**new**

creates a new Mon::SNMP object.

buffer (buffer)

Assigns a raw SNMP trap message to the object.

decode

Decodes a SNMP trap message, and returns a hash of the variable assignments for the SNMP header and trap protocol data unit of the associated message. The hash consists of the following members:

version	=>	SNMP version (1)
community	=>	community string
ent_OID	=>	enterprise OID of originating agent
agentaddr	=>	IP address of originating agent
generic_trap	=>	/COLDSTART WARMSTART LINKDOWN LINKUP AUTHFAIL
specific_trap	=>	specific trap type (integer)
timeticks	=>	timeticks (integer)
varbindlist	=>	{ oid1 => value, oid2 => value, ... }

ERRORS

All methods return a hash with no elements upon errors which they detect, and the detail of the error is available from the

EXAMPLES

```
use Mon::SNMP;

$trap = new Mon::SNMP;

$trap->buffer($snmptrap);

%traphash = $trap->decode;

foreach $oid (keys $traphash{ "varbindlist" }) {
    $val = $traphash{ "varbindlist" }{ $oid };
    print "oid($oid) = val($val)\n";
}
```

ENVIRONMENT

None.

SEE ALSO

Graham Barr's Convert::BER module.



NOTES**CAVEATS**

Mon::SNMP depends upon Convert::BER to do the real work.

