NAME

cap_syslog, **cap_vsyslog**, **cap_openlog**, **cap_closelog**, **cap_setlogmask** - library for syslog in capability mode

LIBRARY

library "libcap_syslog"

SYNOPSIS

#include <libcasper.h>
#include <casper/cap_syslog.h>

void
cap_syslog(*cap_channel_t *chan*, *int pri*, *const char *fmt*, ...);

void
cap_vsyslog(*cap_channel_t *chan*, *int priority*, *const char *fmt*, *va_list ap*);

void
cap_openlog(*cap_channel_t *chan, const char *ident*, *int logopt*, *int facility*);

void
cap_closelog(*cap_channel_t *chan*);

int
cap_setlogmask(*cap_channel_t *chan, int maskpri*);

DESCRIPTION

The functions **cap_syslog**() **cap_vsyslog**() **cap_openlog**() **cap_closelog**() **cap_setlogmask**() are respectively equivalent to syslog(3), vsyslog(3), openlog(3), closelog(3), setlogmask(3) except that the connection to the **system.syslog** service needs to be provided.

EXAMPLES

The following example first opens a capability to casper and then uses this capability to create the **system.syslog** casper service to log messages.

```
cap_channel_t *capcas, *capsyslog;

/* Open capability to Casper. */
capcas = cap_init();
if (capcas == NULL)
```

```
        err(1, "Unable to contact Casper");

/* Enter capability mode sandbox. */
if (cap_enter() < 0 && errno != ENOSYS)
        err(1, "Unable to enter capability mode");

/* Use Casper capability to create capability to the system.syslog service. */
capsyslog = cap_service_open(capcas, "system.syslog");
if (capsyslog == NULL)
        err(1, "Unable to open system.syslog service");

/* Close Casper capability, we don't need it anymore. */
cap_close(capcas);

/* Let's log something. */
cap_syslog(capsyslog, LOG_NOTICE, "System logs from capability mode.");
```

SEE ALSO

cap_enter(2), closelog(3), err(3), openlog(3), setlogmask(3), syslog(3), vsyslog(3), capsicum(4), nv(9)

HISTORY

The **cap_syslog** service first appeared in FreeBSD 10.3.

AUTHORS

Mariusz Zaborski <*oshogbo@FreeBSD.org*>