NAME

cb_ops – character/block entry points structure

SYNOPSIS

#include <sys/conf.h>
#include <sys/ddi.h>
#include <sys/sunddi.h>

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI).

DESCRIPTION

cb_ops contains all entry points for drivers that support both character and block entry points. All leaf device drivers supporting direct user process access to a device should declare a **cb_ops** structure.

All drivers which safely allow multiple threads of execution in the driver at the same time must set the **D_MP** flag in the **cb_flag** field.

If the driver properly handles 64-bit offsets, it should also set the **D_64BIT** flag in the **cb_flag** field. This specifies that the driver will use the **uio_loffset** field of the **uio**(9S) structure.

mt-streams(9F) describes other flags that may be set in the **cb_flag** field.

cb_rev is the **cb_ops** structure revision number. This field must be set to **CB_REV**.

Non-STREAMS drivers should set **cb_str** to **NULL.**

The following DDI/DKI or DKI-only or DDI-only functions are provided in the character/block driver operations structure.

block/char	Function	Description
b/c	XXopen	DDI/DKI
b/c	XXclose	DDI/DKI
b	XXstrategy	DDI/DKI
b	XXprint	DDI/DKI
b	XXdump	DDI(Sun)
c	XXread	DDI/DKI
c	XXwrite	DDI/DKI
c	XXioctl	DDI/DKI
c	XXdevmap	DDI(Sun)
c	XXmmap	DKI
c	XXsegmap	DKI
c	XXchpoll	DDI/DKI
c	XXprop_op	DDI(Sun)
c	XXaread	DDI(Sun)
c	XXawrite	DDI(Sun)

STRUCTURE MEMBERS

```
int     (*cb_open)(dev_t *devp, int flag, int otyp, cred_t *credp);
int     (*cb_close)(dev_t dev, int flag, int otyp, cred_t *credp);
int     (*cb_strategy)(struct buf *bp);
```

```
int       (*cb_print)(dev_t dev, char *str);
int       (*cb_dump)(dev_t dev, caddr_t addr, daddr_t blkno, int nblk);
int       (*cb_read)(dev_t dev, struct uio *uiop, cred_t *credp);
int       (*cb_write)(dev_t dev, struct uio *uiop, cred_t *credp);
int       (*cb_ioctl)(dev_t dev, int cmd, int arg, int mode,
              cred_t *credp, int *rvalp);
int       (*cb_devmap)();
int       (*cb_mmap)(dev_t dev, off_t off, int prot);
int       (*cb_segmap)(dev_t dev, off_t off, struct as *asp,
              caddr_t *addrp, off_t len, unsigned int prot,
              unsigned int maxprot, unsigned int flags, cred_t *credp);
int       (*cb_chpoll)(dev_t dev, short events, int anyyet,
              short *reventsp, struct pollhead **phpp);
int       (*cb_prop_op)(dev_t dev, dev_info_t *dip,
              ddi_prop_op_t prop_op, int mod_flags,
              char *name, caddr_t valuep, int *length);
struct streamtab *cb_str;   /* streams information */
int       cb_flag;
int       cb_rev;
int       (*cb_aread)(dev_t dev, struct aio_req *aio, cred_t *credp);
int       (*cb_awrite)(dev_t dev, struct aio_req *aio, cred_t *credp);
```

SEE ALSO

aread(9E), **awrite**(9E), **chpoll**(9E), **close**(9E), **dump**(9E), **ioctl**(9E), **mmap**(9E), **open**(9E), **print**(9E), **prop_op**(9E), **read**(9E), **segmap**(9E), **strategy**(9E), **write**(9E), **nochpoll**(9F), **nodev**(9F), **nulldev**(9F), **dev_ops**(9S), **qinit**(9S)

Writing Device Drivers

STREAMS Programming Guide

NOTES

The **devmap** entry point is not defined in this release and should be set to **nodev**.