NAME

getinfo – get device driver information

SYNOPSIS

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

int *prefix* **getinfo(dev_info_t** **dip*, **ddi_info_cmd_t** *cmd,* **void** **arg*, **void** ***resultp***);**

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI). This entry point is **required** for drivers which export **cb_ops**(9S) entry points.

ARGUMENTS

dip Do not use.

cmd Command argument – valid command values are **DDI_INFO_DEVT2DEVINFO** and **DDI_INFO_DEVT2INSTANCE.**

arg Command specific argument.

resultp Pointer to where the requested information is stored.

DESCRIPTION

When *cmd* is set to **DDI_INFO_DEVT2DEVINFO**, **getinfo( )** should return the **dev_info_t** pointer associated with the **dev_t** *arg*. The **dev_info_t** pointer should be returned in the field pointed to by *resultp*.

When *cmd* is set to **DDI_INFO_DEVT2INSTANCE**, **getinfo( )** should return the instance number associated with the **dev_t** *arg*. The instance number should be returned in the field pointed to by *resultp*.

Drivers which do not export **cb_ops**(9S) entry points are not required to provide a **getinfo()** entry point, and may use **nodev**(9F) in the **devo_getinfo** field of the **dev_ops**(9S) structure. A SCSI HBA driver is an example of a driver which is not required to provide **cb_ops**(9S) entry points.

RETURN VALUES

getinfo( ) should return:

DDI_SUCCESS on success.

DDI_FAILURE on failure.

EXAMPLES

```
/*ARGSUSED*/
static int
rd_getinfo(dev_info_t *dip, ddi_info_cmd_t infocmd, void *arg, void **result)
{
        /* Note that in this simple example
         * the minor number is the instance
         * number.
         */

    devstate_t *sp;
    int error = DDI_FAILURE;
```

```
        switch (infocmd) {
        case DDI_INFO_DEVT2DEVINFO:
                if ((sp = ddi_get_soft_state(statep,
                    getminor((dev_t) arg))) != NULL) {
                        *resultp = sp->devi;
                        error = DDI_SUCCESS;
                } else
                        *result = NULL;
                break;

        case DDI_INFO_DEVT2INSTANCE:
                *resultp = (void *) getminor((dev_t) arg);
                error = DDI_SUCCESS;
                break;
        }

        return (error);
}
```

SEE ALSO

nodev(9F), **cb_ops**(9S), **dev_ops**(9S)

Writing Device Drivers