NAME

ddi_prop_create, ddi_prop_modify, ddi_prop_remove, ddi_prop_remove_all, ddi_prop_undefine – create, remove, or modify properties for leaf device drivers

SYNOPSIS

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

int ddi_prop_create(dev_t *dev*, **dev_info_t** **dip*, **int** *flags*, **char** **name*, **caddr_t** *valuep*, **int** *length***);**

int ddi_prop_undefine(dev_t *dev*, **dev_info_t** **dip*, **int** *flags*, **char** **name***);**

int ddi_prop_modify(dev_t *dev*, **dev_info_t** **dip*, **int** *flags*, **char** **name*, **caddr_t** *valuep*, **int** *length***);**

int ddi_prop_remove(dev_t *dev*, **dev_info_t** **dip*, **char** **name***);**

void ddi_prop_remove_all(dev_info_t **dip***);**

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI).

ARGUMENTS

ddi_prop_create( )

dev	**dev_t** of the device.
dip	**dev_info_t** pointer of the device.
flags	flag modifiers. The only possible flag value is **DDI_PROP_CANSLEEP:** Memory allocation may sleep.
name	name of property.
valuep	pointer to property value.
length	property length.

ddi_prop_undefine( )

dev	**dev_t** of the device.
dip	**dev_info_t** pointer of the device.
flags	flag modifiers. The only possible flag value is **DDI_PROP_CANSLEEP:** Memory allocation may sleep.
name	name of property.

ddi_prop_modify( )

dev	**dev_t** of the device.
dip	**dev_info_t** pointer of the device.
flags	flag modifiers. The only possible flag value is **DDI_PROP_CANSLEEP:** Memory allocation may sleep.
name	name of property.
valuep	pointer to property value.
length	property length.

ddi_prop_remove( )

dev	**dev_t** of the device.
dip	**dev_info_t** pointer of the device.
name	name of property.

ddi_prop_remove_all( )

dip	**dev_info_t** pointer of the device.

DESCRIPTION

Device drivers have the ability to create and manage their own properties as well as gain access to properties that the system creates on behalf of the driver. A driver uses **ddi_getproplen**(9F) to query whether or not a specific property exists.

Property creation is done by creating a new property definition in the driver's property list associated with *dip*.

Property definitions are stacked; they are added to the beginning of the driver's property list when created. Thus, when searched for, the most recent matching property definition will be found and its value will be return to the caller.

ddi_prop_create( )

ddi_prop_create( ) adds a property to the device's property list. If the property is not associated with any particular *dev* but is associated with the physical device itself, then the argument *dev* should be the special device **DDI_DEV_T_NONE.** If you do not have a *dev* for your device (for example during **attach**(9E) time), you can create one using **makedevice**(9F) with a major number of **DDI_MAJOR_T_UNKNOWN**. **ddi_prop_create( )** will then make the correct *dev* for your device.

For boolean properties, you must set *length* to **0.** For all other properties, the *length* argument must be set to the number of bytes used by the data structure representing the property being created.

Note that creating a property involves allocating memory for the property list, the property name and the property value. If *flags* does not contain **DDI_PROP_CANSLEEP, ddi_prop_create( )** returns **DDI_PROP_NO_MEMORY** on memory allocation failure or **DDI_SUCCESS** if the allocation succeeded. If **DDI_PROP_CANSLEEP** was set, the caller may sleep until memory becomes available.

ddi_prop_undefine( )

ddi_prop_undefine( ) is a special case of property creation where the value of the property is set to undefined. This property has the effect of terminating a property search at the current devinfo node, rather than allowing the search to proceed up to ancestor devinfo nodes. See **ddi_prop_op**(9F).

Note that undefining properties does involve memory allocation, and therefore, is subject to the same memory allocation constraints as **ddi_prop_create( )**.

ddi_prop_modify( )

ddi_prop_modify( ) modifies the length and the value of a property. If **ddi_prop_modify( )** finds the property in the driver's property list, allocates memory for the property value and returns **DDI_PROP_SUCCESS.** If the property was not found, the function returns **DDI_PROP_NOT_FOUND.**

Note that modifying properties does involve memory allocation, and therefore, is subject to the same memory allocation constraints as **ddi_prop_create( )**.

ddi_prop_remove( )

ddi_prop_remove( ) unlinks a property from the device's property list. If **ddi_prop_remove( )** finds the property (an exact match of both *name* and *dev*), it unlinks the property, frees its memory, and returns **DDI_PROP_SUCCESS,** otherwise, it returns **DDI_PROP_NOT_FOUND.**

ddi_prop_remove_all( )

ddi_prop_remove_all( ) removes the properties of all the **dev_t**'s associated with the *dip*. It is called before unloading a driver.

RETURN VALUES

ddi_prop_create( )

DDI_PROP_SUCCESS	on success.
DDI_PROP_NO_MEMORY	on memory allocation failure.
DDI_PROP_INVAL_ARG	if an attempt is made to create a property with *dev* equal to **DDI_DEV_T_ANY** or if *name* is **NULL** or *name* is the **NULL** string.

ddi_prop_undefine( )

DDI_PROP_SUCCESS	on success.
DDI_PROP_NO_MEMORY	on memory allocation failure.
DDI_PROP_INVAL_ARG	if an attempt is made to create a property with *dev* **DDI_DEV_T_ANY** or if *name* is **NULL** or *name* is the **NULL** string.

ddi_prop_modify( )

DDI_PROP_SUCCESS	on success.
DDI_PROP_NO_MEMORY	on memory allocation failure.
DDI_PROP_INVAL_ARG	if an attempt is made to create a property with *dev* equal to **DDI_DEV_T_ANY** or if *name* is **NULL** or *name* is the **NULL** string.
DDI_PROP_NOT_FOUND	on property search failure.

ddi_prop_remove( )

DDI_PROP_SUCCESS	on success.
DDI_PROP_INVAL_ARG	if an attempt is made to create a property with *dev* equal to **DDI_DEV_T_ANY** or if *name* is **NULL** or *name* is the **NULL** string.
DDI_PROP_NOT_FOUND	on property search failure.

CONTEXT

If **DDI_PROP_CANSLEEP** is set, these functions can only be called from user context; otherwise, they can be called from interrupt or user context.

EXAMPLES

Create a property called *nblocks* for each partition on a disk.

```
for (minor = 0; minor < 8; minor ++) {
        (void) ddi_prop_create(makedevice(DDI_MAJOR_T_UNKNOWN, minor),
          dev, DDI_PROP_CANSLEEP, "nblocks", 8192, sizeof (int));
        ...
}
```

SEE ALSO

attach(9E), **ddi_prop_op**(9F), **makedevice**(9F), **driver.conf**(4)

Writing Device Drivers