NAME

ddi_prop_op, ddi_getprop, ddi_getlongprop, ddi_getlongprop_buf, ddi_getproplen – get property information for leaf device drivers

SYNOPSIS

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

int ddi_prop_op(dev_t *dev*, **dev_info_t** **dip*, **ddi_prop_op_t** *prop_op,* **int** *flags*, **char** **name*, **caddr_t** *valuep*, **int** **lengthp***);**

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

int ddi_getlongprop(dev_t *dev*, **dev_info_t** **dip*, **int** *flags*, **char** **name*, **caddr_t** *valuep*, **int** **lengthp***);**

int ddi_getlongprop_buf(dev_t *dev*, **dev_info_t** **dip*, **int** *flags*, **char** **name*, **caddr_t** *valuep*, **int** **lengthp***);**

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

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI).

ARGUMENTS

dev — Device number associated with property or **DDI_DEV_T_ANY** as the *wildcard* device number.

dip — Pointer to a device info node.

prop_op — Property operator.

flags — Possible flag values are some combination of:

- **DDI_PROP_DONTPASS** — do not pass request to parent device information node if property not found
- **DDI_PROP_CANSLEEP** — the routine may sleep while allocating memory
- **DDI_PROP_NOTPROM** — do not look at PROM properties (ignored on architectures that do not support PROM properties).

name — String containing the name of the property.

valuep — If *prop_op* is **PROP_LEN_AND_VAL_BUF**, this should be a pointer to the users buffer. If *prop_op* is **PROP_LEN_AND_VAL_ALLOC**, this should be the *address* of a pointer.

lengthp — On exit, **lengthp* will contain the property length. If *prop_op* is **PROP_LEN_AND_VAL_BUF** then before calling **ddi_prop_op( )**, *lengthp* should point to an **int** that contains the length of callers buffer.

defvalue — The value that **ddi_getprop( )** returns if the property is not found.

DESCRIPTION

ddi_prop_op( ) gets arbitrary-size properties for leaf devices. The routine searches the device's property list. If it does not find the property at the device level, it examines the *flags* argument, and if **DDI_PROP_DONTPASS** is set, then **ddi_prop_op( )** returns **DDI_PROP_NOT_FOUND**. Otherwise, it passes the request to the next level of the device info tree. If it does find the property, but the property has been explicitly undefined, it returns **DDI_PROP_UNDEFINED**. Otherwise it returns either the property length, or both the length and value of the property to the caller via the *valuep* and *lengthp* pointers, depending on the value of *prop_op*, as described below, and returns **DDI_PROP_SUCCESS**. If a property cannot be found at all, **DDI_PROP_NOT_FOUND** is returned.

Usually, the *dev* argument should be set to the actual device number that this property applies to. However, if the *dev* argument is **DDI_DEV_T_ANY**, the *wildcard dev*, then **ddi_prop_op( )** will match the request based on *name* only (regardless of the actual *dev* the property was created with). This property/dev match is done according to the property search order which is to first search software properties created by the driver in *last-in, first-out* (LIFO) order, next search software properties created by the *system* in LIFO order, then search PROM properties if they exist in the system architecture.

Property operations are specified by the *prop_op* argument. If *prop_op* is **PROP_LEN**, then **ddi_prop_op( )** just sets the callers length, **lengthp*, to the property length and returns the value **DDI_PROP_SUCCESS** to the caller. The *valuep* argument is not used in this case. Property lengths are **0** for boolean properties, **sizeof (int)** for integer properties, and size in bytes for long (variable size) properties.

If *prop_op* is **PROP_LEN_AND_VAL_BUF**, then *valuep* should be a pointer to a user-supplied buffer whose length should be given in **lengthp* by the caller. If the requested property exists, **ddi_prop_op( )** first sets **lengthp* to the property length. It then examines the size of the buffer supplied by the caller, and if it is large enough, copies the property value into that buffer, and returns **DDI_PROP_SUCCESS**. If the named property exists but the buffer supplied is too small to hold it, it returns **DDI_PROP_BUF_TOO_SMALL**.

If *prop_op* is **PROP_LEN_AND_VAL_ALLOC**, and the property is found, **ddi_prop_op( )** sets **lengthp* to the property length. It then attempts to allocate a buffer to return to the caller using the **kmem_alloc**(9F) routine, so that memory can be later recycled using **kmem_free**(9F). The driver is expected to call **kmem_free( )** with the returned address and size when it is done using the allocated buffer. If the allocation is successful, it sets **valuep* to point to the allocated buffer, copies the property value into the buffer and returns **DDI_PROP_SUCCESS**. Otherwise, it returns **DDI_PROP_NO_MEMORY**. Note that the *flags* argument may affect the behavior of memory allocation in **ddi_prop_op( )**. In particular, if **DDI_PROP_CANSLEEP** is set, then the routine will wait until memory is available to copy the requested property.

ddi_getprop( ) returns boolean and integer-size properties. It is a convenience wrapper for **ddi_prop_op( )** with *prop_op* set to **PROP_LEN_AND_VAL_BUF**, and the buffer is provided by the wrapper. By convention, this function returns a **1** for boolean (zero-length) properties.

ddi_getlongprop( ) returns arbitrary-size properties. It is a convenience wrapper for **ddi_prop_op( )** with *prop_op* set to **PROP_LEN_AND_VAL_ALLOC**, so that the routine will allocate space to hold the buffer that will be returned to the caller via **valuep*.

ddi_getlongprop_buf( ) returns arbitrary-size properties. It is a convenience wrapper for **ddi_prop_op( )** with *prop_op* set to **PROP_LEN_AND_VAL_BUF** so the user must supply a buffer.

ddi_getproplen( ) returns the length of a given property. It is a convenience wrapper for **ddi_prop_op( )** with *prop_op* set to **PROP_LEN**.

RETURN VALUES

ddi_prop_op( )
ddi_getlongprop( )
ddi_getlongprop_buf( )
ddi_getproplen( ) return:

DDI_PROP_SUCCESS	Property found and returned.
DDI_PROP_NOT_FOUND	Property not found.
DDI_PROP_UNDEFINED	Property already explicitly undefined.
DDI_PROP_NO_MEMORY	Property found, but unable to allocate memory. *lengthp* points to the correct property length.
DDI_PROP_BUF_TOO_SMALL	Property found, but the supplied buffer is too small. *lengthp* points to the correct property length.

ddi_getprop( ) returns:

The value of the property or the value passed into the routine as *defvalue* if the property is not found. By convention, the value of zero length properties (boolean properties) are returned as the integer value 1.

CONTEXT

These functions can be called from user or interrupt context, provided **DDI_PROP_CANSLEEP** is not set; if it is set, they can be called from user context only.

SEE ALSO

ddi_prop_create(9F), **kmem_alloc**(9F), **kmem_free**(9F)

Writing Device Drivers