NAME

ddi_dma_getwin – activate a new DMA window

SYNOPSIS

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

int ddi_dma_getwin(ddi_dma_handle_t handle, uint_t win,
        off_t *offp, uint_t *lenp, ddi_dma_cookie_t *cookiep, uint_t *ccountp);
```

ARGUMENTS

handle	The DMA handle previously allocated by a call to **ddi_dma_alloc_handle**(9F).
win	Number of the window to activate.
offp	Pointer to an offset. Upon a successful return, *offp* will contain the new offset indicating the beginning of the window within the object.
lenp	Upon a successful return, *lenp* will contain the size, in bytes, of the current window.
cookiep	A pointer to the first **ddi_dma_cookie**(9S) structure.
ccountp	Upon a successful return, *ccountp* will contain the number of cookies for this DMA window.

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI).

DESCRIPTION

ddi_dma_getwin( ) activates a new DMA window. If a DMA resource allocation request returns **DDI_DMA_PARTIAL_MAP** indicating that resources for less than the entire object were allocated, the current DMA window can be changed by a call to **ddi_dma_getwin( )**.

The caller must first determine the number of DMA windows, *N*, using **ddi_dma_numwin**(9F). **ddi_dma_getwin( )** takes a DMA window number from the range *[0..N-1]* as the parameter *win* and makes it the current DMA window.

ddi_dma_getwin( ) fills in the first DMA cookie pointed to by *cookiep* with the appropriate address, length, and bus type. **ccountp* is set to the number of DMA cookies representing this DMA object. Subsequent DMA cookies must be retrieved using **ddi_dma_nextcookie**(9F).

ddi_dma_getwin( ) takes care of underlying resource synchronizations required to shift the window. However accessing the data prior to or after moving the window requires further synchronization steps using **ddi_dma_sync**(9F).

ddi_dma_getwin( ) is normally called from an interrupt routine. The first invocation of the DMA engine is done from the driver. All subsequent invocations of the DMA engine are done from the interrupt routine. The interrupt routine checks to see if the request has been completed. If it has, the interrupt routine returns without invoking another DMA transfer. Otherwise, it calls **ddi_dma_getwin( )** to shift the current window and start another DMA transfer.

RETURN VALUES

ddi_dma_getwin( ) returns:

DDI_SUCCESS Resources for the specified DMA window are allocated.

DDI_FAILURE *win* is not a valid window index.

CONTEXT

ddi_dma_getwin( ) can be called from user, kernel, or interrupt context.

SEE ALSO

ddi_dma_addr_bind_handle(9F), **ddi_dma_alloc_handle**(9F), **ddi_dma_buf_bind_handle**(9F), **ddi_dma_nextcookie**(9F), **ddi_dma_numwin**(9F), **ddi_dma_sync**(9F), **ddi_dma_unbind_handle**(9F), **ddi_dma_cookie**(9S)

Writing Device Drivers