NAME ddi_dma_nextwin – get next DMA window

SYNOPSIS

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

int ddi_dma_nextwin(ddi_dma_handle_t *handle,* **ddi_dma_win_t** *win,*
ddi_dma_win_t **nwin***);**

INTERFACE LEVEL Solaris DDI specific (Solaris DDI).

ARGUMENTS

handle	A DMA *handle.*
win	The current DMA window or **NULL**.
nwin	A pointer to the next DMA window to be filled in. If *win* is **NULL**, a pointer to the first window within the object is returned.

DESCRIPTION

ddi_dma_nextwin( ) shifts the current DMA window *win* within the object referred to by *handle* to the next DMA window *nwin.* If the current window is **NULL**, the first window within the object is returned. A DMA window is a portion of a DMA object or might be the entire object. A DMA window has system resources allocated to it and is prepared to accept data transfers. Examples of system resources are DVMA mapping resources and intermediate transfer buffer resources.

All DMA objects require a window. If the DMA window represents the whole DMA object it has system resources allocated for the entire data transfer. However, if the system is unable to setup the entire DMA object due to system resource limitations, the driver writer may allow the system to allocate system resources for less than the entire DMA object. This can be accomplished by specifying the **DDI_DMA_PARTIAL** flag as a parameter to **ddi_dma_buf_setup**(9F) or **ddi_dma_addr_setup**(9F) or as part of a **ddi_dma_req**(9S) structure in a call to **ddi_dma_setup**(9F).

Only the window that has resources allocated is valid per object at any one time. The currently valid window is the one that was most recently returned from **ddi_dma_nextwin( )**. Furthermore, because a call to **ddi_dma_nextwin( )** will reallocate system resources to the new window, the previous window will become invalid. **Note:** It is a *severe* error to call **ddi_dma_nextwin( )** before any transfers into the current window are complete.

ddi_dma_nextwin( ) takes care of underlying memory synchronizations required to shift the window. However, if you want to access the data before or after moving the window, further synchronizations using **ddi_dma_sync**(9F) are required.

RETURN VALUES

ddi_dma_nextwin( ) returns:

DDI_SUCCESS	Successfully filled in the next window pointer.
DDI_DMA_DONE	There is no next window. The current window is the final window within the specified object.
DDI_DMA_STALE	*win* does not refer to the currently active window.

CONTEXT

ddi_dma_nextwin( ) can be called from user or interrupt context.

EXAMPLES

For an example see **ddi_dma_segtocookie**(9F).

SEE ALSO

ddi_dma_addr_setup(9F), **ddi_dma_buf_setup**(9F), **ddi_dma_nextseg**(9F), **ddi_dma_segtocookie**(9F), **ddi_dma_sync**(9F), **ddi_dma_req**(9S)

Writing Device Drivers