**NAME**

ddi_mem_rep_putb, ddi_mem_rep_putw, ddi_mem_rep_putl, ddi_mem_rep_putll – write multiple data to mapped device in the memory space or allocated DMA memory

**SYNOPSIS**

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

void ddi_mem_rep_putb(ddi_acc_handle_t handle, uchar_t *host_addr,
        uchar_t *dev_addr, uint_t repcount, ulong_t flags);

void ddi_mem_rep_putw(ddi_acc_handle_t handle, ushort_t *host_addr,
        ushort_t *dev_addr, uint_t repcount, ulong_t flags);

void ddi_mem_rep_putl(ddi_acc_handle_t handle, ulong_t *host_addr,
        ulong_t *dev_addr, uint_t repcount, ulong_t flags);

void ddi_mem_rep_putll(ddi_acc_handle_t handle, unsigned long long *host_addr,
        unsigned long long *dev_addr, uint_t repcount, ulong_t flags);
```

**ARGUMENTS**

*handle* The data access handle returned from setup calls, such as **ddi_regs_map_setup**(9F).

*host_addr* Base host address.

*dev_addr* Base device address.

*repcount* Number of data accesses to perform.

*flags* Device address flags:

**DDI_DEV_AUTOINCR**
Automatically increment the device address, *dev_addr*, during data accesses.

**DDI_DEV_NO_AUTOINCR**
Do not advance the device address, *dev_addr*, during data accesses.

**INTERFACE LEVEL**

Solaris DDI specific (Solaris DDI).

**DESCRIPTION**

These routines generate multiple writes to memory space or allocated DMA memory. *repcount* data is copied from the host address, *host_addr*, to the device address, *dev_addr*, in memory space. For each input datum, the **ddi_mem_rep_putb( )**, **ddi_mem_rep_putw( )**, **ddi_mem_rep_putl( )**, and **ddi_mem_rep_putll( )** functions write 8 bits, 16 bits, 32 bits and 64 bits of data, respectively, to the device address. *dev_addr* and *host_addr* must be aligned to the datum boundary described by the function.

Each individual datum will automatically be translated to maintain a consistent view between the host and the device based on the encoded information in the data access handle. The translation may involve byte-swapping if the host and the device have incompatible endian characteristics.

When the *flags* argument is set to **DDI_DEV_AUTOINCR**, these functions will treat the device address, *dev_addr*, as a memory buffer location on the device and increments its address on the next input datum. However, when the *flags* argument is set to **DDI_DEV_NO_AUTOINCR**, the same device address will be used for every datum access. For example, this flag may be useful when writing from a data register.

**CONTEXT**

These functions can be called from user, kernel, or interrupt context.

**SEE ALSO**

**ddi_mem_getb**(9F), **ddi_mem_putb**(9F), **ddi_mem_rep_getb**(9F), **ddi_regs_map_setup**(9F), **ddi_device_acc_attr**(9S)