NAME

ddi_regs_map_setup – set up a mapping for a register address space

SYNOPSIS

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

int ddi_regs_map_setup(dev_info_t **dip*, **uint_t** *rnumber*, **caddr_t** **addrp*,
offset_t *offset*, **offset_t** *len*, **ddi_device_acc_attr_t** **accattrp*,
ddi_acc_handle_t **handlep***);**

AVAILABILITY

PCI Local Bus, SBus, ISA, EISA, MCA

ARGUMENTS

dip	Pointer to the device's **dev_info** structure.
rnumber	Index number to the register address space set.
addrp	Pointer to the mapping address base.
offset	Offset into the register address space.
len	Length to be mapped.
accattrp	Pointer to a device access attribute structure of this device (see **ddi_device_acc_attr**(9S)).
handlep	Pointer to a data access handle.

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI).

DESCRIPTION

ddi_regs_map_setup( ) maps in the register set given by *rnumber*. The register number determines which register set is mapped if more than one exists.

offset specifies the starting location within the register space and *len* indicates the size of the area to be mapped. If *len* is non-zero, it overrides the length given in the register set description. If both *len* and *offset are* **0**, the entire space is mapped. The base of the mapped register space is returned in *addrp*.

The device access attributes are specified in the location pointed by the *accattrp* argument (see **ddi_device_acc_attr**(9S) for details).

The data access handle is returned in *handlep*. *handlep* is opaque – drivers should not attempt to interpret its value. The handle is used by the system to encode information for subsequent data access function calls to maintain a consistent view between the host and the device.

RETURN VALUES

ddi_regs_map_setup( ) returns:

DDI_SUCCESS Successfully set up the mapping for data access.

DDI_FAILURE Invalid register number *rnumber*, offset *offset*, or length *len*.

DDI_REGS_ACC_CONFLICT
Cannot enable the register mapping due to access conflicts with other enabled mappings.

CONTEXT

ddi_regs_map_setup( ) must be called from user or kernel context.

SEE ALSO

ddi_regs_map_free(9F), **ddi_device_acc_attr**(9S)

Writing Device Drivers