NAME

bus_activate_resource, **bus_deactivate_resource** **-** activate or deactivate a resource

SYNOPSIS

#include <sys/param.h>
#include <sys/bus.h>

#include <machine/bus.h>
#include <sys/rman.h>
#include <machine/resource.h>

int
bus_activate_resource(*device_t dev*, *int type*, *int rid*, *struct resource *r*);

int
bus_deactivate_resource(*device_t dev*, *int type*, *int rid*, *struct resource *r*);

DESCRIPTION

These functions activate or deactivate a previously allocated resource. In general, resources must be activated before they can be accessed by the driver. Bus drivers may perform additional actions to ensure that the resource is ready to be accessed. For example, the PCI bus driver enables memory decoding in a PCI device's command register when activating a memory resource.

The arguments are as follows:

dev The device that requests ownership of the resource. Before allocation, the resource is owned by the parent bus.

type The type of resource you want to allocate. It is one of:

PCI_RES_BUS	for PCI bus numbers
SYS_RES_IRQ	for IRQs
SYS_RES_DRQ	for ISA DMA lines
SYS_RES_IOPORT	for I/O ports
SYS_RES_MEMORY	for I/O memory

rid A pointer to a bus specific handle that identifies the resource being allocated.

r A pointer to the *struct resource* returned by bus_alloc_resource(9).

Resource Mapping

Resources which can be mapped for CPU access by a bus_space(9) tag and handle will create a mapping of the entire resource when activated. The tag and handle for this mapping are stored in *r* and can be retrieved via rman_get_bustag(9) and rman_get_bushandle(9). These can be used with the bus_space(9) API to access device registers or memory described by *r*. If the mapping is associated with a virtual address, the virtual address can be retrieved via rman_get_virtual(9).

This implicit mapping can be disabled by passing the RF_UNMAPPED flag to bus_alloc_resource(9). A driver may use this if it wishes to allocate its own mappings of a resource using bus_map_resource(9).

A wrapper API for bus_space(9) is also provided that accepts the associated resource as the first argument in place of the bus_space(9) tag and handle. The functions in this wrapper API are named similarly to the bus_space(9) API except that "_space" is removed from their name. For example, **bus_read_4**() can be used in place of **bus_space_read_4**(). The wrapper API is preferred in new drivers.

These two statements both read a 32-bit register at the start of a resource:

```
bus_space_read_4(rman_get_bustag(res), rman_get_bushandle(res), 0);
bus_read_4(res, 0);
```

RETURN VALUES

Zero is returned on success, otherwise an error is returned.

SEE ALSO

bus_alloc_resource(9), bus_map_resource(9), bus_space(9), device(9), driver(9)

AUTHORS

This manual page was written by Warner Losh <*imp@FreeBSD.org*>.