NAME

scsi_pktalloc, scsi_resalloc, scsi_pktfree, scsi_resfree – SCSI packet utility routines

SYNOPSIS

```
#include <sys/scsi/scsi.h>

struct scsi_pkt *scsi_pktalloc(struct scsi_address *ap, int cmdlen, int statuslen,
        int (*callback)(void));

struct scsi_pkt *scsi_resalloc(struct scsi_address *ap, int cmdlen, int statuslen,
        opaque_t dmatoken, int (*callback)(void));

void scsi_pktfree(struct scsi_pkt *pkt);

void scsi_resfree(struct scsi_pkt *pkt);
```

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI).

ARGUMENTS

ap	Pointer to a **scsi_address** structure.
cmdlen	The required length for the SCSI command descriptor block (CDB) in bytes.
statuslen	The required length for the SCSI status completion block (SCB) in bytes.
dmatoken	Pointer to an implementation-dependent object.
callback	A pointer to a callback function, or **NULL_FUNC** or **SLEEP_FUNC.**
pkt	Pointer to a **scsi_pkt**(9S) structure.

DESCRIPTION

scsi_pktalloc( ) requests the host adapter driver to allocate a command packet. For commands that have a data transfer associated with them, **scsi_resalloc( )** should be used.

ap is a pointer to a **scsi_address** structure. Allocator routines use it to determine the associated host adapter.

cmdlen is the required length for the SCSI command descriptor block. This block is allocated such that a kernel virtual address is established in the **pkt_cdbp** field of the allocated **scsi_pkt** structure.

statuslen is the required length for the SCSI status completion block. The address of the allocated block is placed into the **pkt_scbp** field of the **scsi_pkt** structure.

dmatoken is a pointer to an implementation dependent object which defines the length, direction, and address of the data transfer associated with this SCSI packet (command). The *dmatoken* must be a pointer to a **buf**(9S) structure. If *dmatoken* is **NULL**, no DMA resources are required by this SCSI command, so none are allocated. Only one transfer direction is allowed per command. If there is an unexpected data transfer phase (either no data transfer phase expected, or the wrong direction encountered), the command is terminated with the **pkt_reason** set to **CMD_DMA_DERR.** *dmatoken* provides the information to determine if the transfer count is correct.

callback indicates what the allocator routines should do when resources are not available:

NULL_FUNC	Do not wait for resources. Return a **NULL** pointer.
SLEEP_FUNC	Wait indefinitely for resources.
Other Values	*callback* points to a function which is called when resources may have become available. *callback* **must** return either **0** (indicating that it attempted to allocate resources but again failed to do so), in which case it is put back on a list to be called again later, or **1** indicating either success in allocating resources or indicating that it no longer cares for a retry.

scsi_pktfree( ) frees the packet.

scsi_resfree( ) free all resources held by the packet and the packet itself.

RETURN VALUES

Both allocation routines return a pointer to a **scsi_pkt** structure on success, or **NULL** on failure.

CONTEXT

If *callback* is **SLEEP_FUNC,** then this routine may only be called from user-level code. Otherwise, it may be called from either user or interrupt level. The *callback* function may not block or call routines that block. Both deallocation routines can be called from user or interrupt context.

SEE ALSO

scsi_dmafree(9F), **scsi_dmaget**(9F), **buf**(9S), **scsi_pkt**(9S)

Writing Device Drivers