## NAME

tran_init_pkt, tran_destroy_pkt – SCSI HBA packet preparation and deallocation

## SYNOPSIS

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

**struct scsi_pkt** **prefix* **tran_init_pkt( struct scsi_address** **ap*, **struct scsi_pkt** **pkt*, **struct buf** **bp*, **int** *cmdlen*, **int** *statuslen*, **int** *tgtlen*, **int** *flags*, **int (****callback***)(caddr_t), caddr_t** *arg***);**

**void** *prefix* **tran_destroy_pkt(struct scsi_address** **ap*, **struct scsi_pkt** **pkt***);**

## INTERFACE LEVEL

Solaris architecture specific (Solaris DDI).

## ARGUMENTS

*ap* Pointer to a **scsi_address**(9S) structure.

*pkt* Pointer to a **scsi_pkt**(9S) structure allocated in an earlier call, or **NULL**.

*bp* Pointer to a **buf**(9S) structure if DMA resources are to be allocated for the *pkt*, or **NULL**.

*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.

*tgtlen* The length of the packet private area within the **scsi_pkt** to be allocated on behalf of the SCSI target driver.

*flags* flags for creating the packet.

*callback* Pointer to either **NULL_FUNC** or **SLEEP_FUNC**.

*arg* always **NULL**.

## DESCRIPTION

The **tran_init_pkt( )** and **tran_destroy_pkt( )** vectors in the **scsi_hba_tran** structure must be initialized during the HBA driver's **attach**(9E) to point to HBA entry points to be called when a target driver calls **scsi_init_pkt**(9F) and **scsi_destroy_pkt**(9F).

### tran_init_pkt( )

**tran_init_pkt( )** is the entry point into the HBA which is used to allocate and initialize a **scsi_pkt** structure on behalf of a SCSI target driver. If *pkt* is **NULL**, the HBA driver must use **scsi_hba_pkt_alloc**(9F) to allocate a new **scsi_pkt** structure.

If *bp* is non-**NULL**, the HBA driver must allocate appropriate DMA resources for the *pkt*, for example, via **ddi_dma_buf_setup**(9F).

If the **PKT_CONSISTENT** bit is set in *flags*, the buffer was allocated by **scsi_alloc_consistent_buf**(9F). For packets marked with **PKT_CONSISTENT** the HBA driver must synchronize any cached data transfers before calling the target driver's command completion callback.

If the **PKT_DMA_PARTIAL** bit is set in *flags*, the HBA driver should set up partial data transfers, such as setting the **DDI_DMA_PARTIAL** bit in the *flags* argument if interfaces such as **ddi_dma_buf_setup**(9F) are used.

If only partial DMA resources are available, **tran_init_pkt()** must return in the **pkt_resid** field of *pkt* the number of bytes of DMA resources not allocated.

If both *pktp* and *bp* are non-**NULL**, the **PKT_DMA_PARTIAL** bit is set in *flags* and DMA resources have already been allocated for the pkt with a previous call to **tran_init_pkt()** that returned a non-zero **pkt_resid** field, this request is to move the DMA resources for the subsequent piece of the transfer.

The contents of **scsi_address**(9S) pointed to by *ap* are copied into the **pkt_address** field of the **scsi_pkt**(9S) by **scsi_hba_pkt_alloc**(9F).

*tgtlen* is the length of the packet private area in the **scsi_pkt** structure to be allocated on behalf of the SCSI target driver.

*statuslen* is the required length for the SCSI status completion block. If the requested status length is greater than or equal to **sizeof(struct scsi_arq_status)** and the **auto_rqsense** capability has been set, automatic request sense is enabled for this packet. If the status length is less than **sizeof(struct scsi_arq_status)**, automatic request sense must be disabled for this *pkt*.

*cmdlen* is the required length for the SCSI command descriptor block.

**Note**: *tgtlen*, *statuslen*, and *cmdlen* are used only when the HBA driver allocates the **scsi_pkt**(9S); in other words, when *pkt* is **NULL**.

*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.

### tran_destroy_pkt()

**tran_destroy_pkt()** is the entry point into the HBA that must free all of the resources that were allocated to the **scsi_pkt**(9S) structure during **tran_init_pkt()**.

## RETURN VALUES

**tran_init_pkt()** must return a pointer to a **scsi_pkt**(9S) structure on success, or **NULL** on failure.

If *pkt* is **NULL** on entry, and **tran_init_pkt()** allocated a pkt via **scsi_hba_pkt_alloc**(9F) but was unable to allocate DMA resources, **tran_init_pkt()** must free the pkt via **scsi_hba_pkt_free**(9F) before returning **NULL**.

## SEE ALSO

**attach**(9E), **tran_sync_pkt**(9E), **ddi_dma_buf_setup**(9F), **scsi_alloc_consistent_buf**(9F), **scsi_destroy_pkt**(9F), **scsi_hba_attach**(9F), **scsi_hba_pkt_alloc**(9F), **scsi_hba_pkt_free**(9F), **scsi_init_pkt**(9F), **buf**(9S), **scsi_address**(9S), **scsi_hba_tran**(9S), **scsi_pkt**(9S)

*Writing Device Drivers*

## NOTES

If a DMA allocation request fails with **DDI_DMA_NOMAPPING**, the **B_ERROR** flag should be set in *bp*, and the **b_error** field should be set to **EFAULT.**

If a DMA allocation request fails with **DDI_DMA_TOOBIG**, the **B_ERROR** flag should be set in *bp*, and the **b_error** field should be set to **EINVAL**.