allocateBufferOfLength:actualStart:actualLength:
executeRequest:buffer:client:
getDMAAlignment:
maxTransfer
releaseTarget:lun:forOwner:
reserveTarget:lun:forOwner:
resetSCSIBus
returnFromScStatus:

initFromDeviceDescription:

Reserving target/lun pairs numReserved

Getting and setting parameters getIntValues:forParameter:count:
setIntValues:forParameter:count:

Collecting statistics maxQueueLength
numQueueSamples
sumQueueLengths
resetStats

(IOReturn)getIntValues:(unsigned int *)parameterArray
forParameter:(IOParameterName)parameterName
count:(unsigned int *)count

Handles the two parameters specific to SCSI controllersÐIO_SCSI_CONTROLLER_STATS and IO_IS_A_SCSI_CONTROLLERÐand forwards the handling of all other parameters to super. The returned for IO_SCSI_CONTROLLER_STATS is set to the numbers returned by maxQueueLengt and sumQueueLengths. No array is returned for IO_IS_A_SCSI_CONTROLLER only IO_R_SUC indicate that this IODevice is indeed a SCSI controller.

setIntValues:forParameter:count:

initFromDeviceDescription:deviceDescription

Initializes a new IOSCSIController instance. After invoking IODirectDevice's version of initFrom this method starts an I/O thread (with startIOThread) and sets its unit, name, and device kind. Each its own unit number. The first instance's unit is 0, the second is 1, and so on. The name is the conc the unit (for example, ªsc0º) , and the device kind is ªscº.

This method also determines the alignment restrictions for the hardware, using the getDMAAlignm alignment restrictions are used by the method allocateBufferOfLength:actualStart:actualLength:.

This method returns nil and frees the instance if initialization failed otherwise, it returns self.

You should implement this method to invoke IOSCSIController's version and to then perform any initialization, including initializing the hardware and (at the very end) invoking registerDevice.

Returns zero. Subclasses that support standard statistics should implement this method so that it re
times the instance has collected information about its queue of I/O requests. This number should b
instance is initialized and when resetStats is called. See the class description for more information
statistics.

(unsigned int)numReserved

Returns the number of target/lun pairs that are reserved. Each pair corresponds to an active device
this instance controls.

reserveTarget:lun:forOwner: and releaseTarget:lun:forOwner: (IOSCSIControllerExported protoc

(void)resetStats

Does nothing. Subclasses that support standard statistics should implement this method so that it re
numbers that are returned by maxQueueLength, numQueueSamples, and sumQueueLengths. See t
more information on supporting standard statistics.

(IOReturn)setIntValues:(unsigned int *)parameterArray
forParameter:(IOParameterName)parameterName
count:(unsigned int)count

Handles the IO_SCSI_CONTROLLER_STATS parameter, forwarding the handling of all other pa
IO_SCSI_CONTROLLER_STATS parameter resets (using resetStats) the standard statistical data

getIntValues:forParameter:count:

(unsigned int)sumQueueLengths

Returns zero. Subclasses that support standard statistics should implement this method so that it re
of requests detected in its queue of I/O requests. This number should be reset to 0 when this instan
when resetStats is called. See the class description for more information on supporting standard sta