encodeRemotelyFor:freeAfterEncoding:isBycopy:
encodeUsing:
decodeUsing:

initWithSize:
initWithData:size:dealloc:
free

Getting the object's data data

Getting the data's size size

Copying the object copyFromZone:

copyFromZone:(NXZone *)zone

Returns a newly allocated NXData instance containing a copy of the receiver's data. The new object's data will be deallocated when the new object gets freed.

(void *)data

Returns a pointer to the data contained in the object.

encodeRemotelyFor: (NXConnection *)conn
freeAfterEncoding:(BOOL *)flagPointer
isBycopy:(BOOL)isBycopy

Returns self to indicate that a copy of the NXData object (and not a proxy to it) is to be copied across a connection any time the object is vended to a remote object. The data for the remote copy will be freed when the copy is freed. If you want the local NXData to be freed after being sent across the connection, you will need to override this method to set the boolean indicated by flagPointer to YES.

initWithData:(void *)data
size:(unsigned int)size
dealloc:(BOOL)flag

Initializes the receiver, a new NXData object, with data, which must be at most size bytes long. If will be deallocated when the NXData object is freed. data could have been allocated with vm_allo variant. Returns self.

initWithSize:, free

initWithSize:(unsigned int)size

Initializes the receiver, a new NXData object, so that it can contain at most size bytes of data. The allocated directly from the virtual memory system if it is one page or greater in size (though applic where the memory came from) otherwise the data will be allocated from the object's zone. The da the NXData object is freed. Returns self.

initWithData:size:dealloc:, free

(unsigned int)size

Returns the size of the data that the object holds.