NSMutableData

Inherits From:	NSData : NSObject
Conforms To:	NSCoding NSCopying NSMutableCopying NSObject
Declared In:	foundation/NSData.h foundation/NSSerialization.h

Allocating and Initializing a Mutable Data Object

+ (id)**allocWithZone:**(NSZone *)*zone*	Creates and returns an uninitialized mutable data object from *zone*.
+ (id)**dataWithCapacity:**(unsigned int)*numBytes*	Creates and returns a mutable data object, initially allocating enough memory to hold *numBytes* bytes.
+ (id)**dataWithLength:**(unsigned int)*length*	Creates and returns a mutable data object, giving it enough memory to hold *length* bytes. Fills the object with zeroes up to *length*.
- (id)**initWithCapacity:**(unsigned int)*capacity*	Initializes a newly allocated mutable data object, giving it enough memory to hold *capacity* bytes. Sets the length of the data object to 0.
- (id)**initWithLength:**(unsigned int)*length*	Initializes a newly allocated mutable data object, giving it enough memory to hold *length* bytes. Fills the object with zeroes up to *length*.

Adjusting Capacity

- (void)**increaseLengthBy:**(unsigned int)*extraLength*	Increases the length of a mutable data object by *extraLength* zero-filled bytes.
- (void)**setLength:**(unsigned int)*length*	Extends or truncates the length of a mutable data object by *length* bytes. If the mutable data object is extended, the additional bytes are zero-filled.
- (void *)**mutableBytes**	Returns a pointer to the bytes in a mutable data object, enabling you to modify the bytes.

Appending Data

- (void)**appendBytes:**(const void *)*bytes* **length:**(unsigned int)*length*	Appends *length* bytes to a mutable data object from the buffer *bytes*.
- (void)**appendData:**(NSData *)*other*	Appends the contents of the data object *other* to the receiver.

Modifying Data

- (void)**replaceBytesInRange:**(NSRange)*aRange* **withBytes:**(const void *)*bytes*	Replaces the receiver's bytes located in *aRange* with *bytes*. *aRange* must specify a range within the receiver's data; otherwise, an NSRangeException error is raised.
- (void)**resetBytesInRange:**(NSRange)*aRange*	Replaces the receiver's bytes located in *aRange* with zeros. If *aRange* isn't within the receiver's range of bytes, an NSRangeException error is raised.

Serializing Data

- (void)**serializeAlignedBytesLength:**(unsigned int)*length*

	Prepares bytes for an **appendBytes:length:** invocation by serializing them. If the *length* of the bytes will cause extension past the page size, this method encodes header information, creating a hole so that all bytes in the data object are aligned on page boundaries.
- (void)**serializeDataAt:**(const void *)*data* **ofObjCType:**(const char *)*type* **context:**(id <NSObjCTypeSerializationCallBack>) *callback*	Serializes whatever data element is referenced by *data*, interpreting it by the Objective C type specifier *type*. If the data element is an object other than an instance of NSDictionary, NSArray, NSString, or NSData, further definition of the object can occur through a callback from object *callback*. All Objective C types are currently supported except **unions** and **void *.** Pointers refer to a single item.
- (void)**serializeInt:**(int)*value*	Serializes the integer *value* by encoding it as a character representation.
- (void)**serializeInt:**(int)*value* **atIndex:**(unsigned int)*location*	Serializes the integer *value* by encoding it as a character representation and replaces the encoded value at the specified *location* in the data.
- (void)**serializeInts:**(int *)*intBuffer* **count:**(unsigned int)*numInts*	Serializes *numInts* count of integers in *intBuffer* by encoding each integer as a character representation.
- (void)**serializeInts:**(int *)*intBuffer* **count:**(unsigned int)*numInts* **atIndex:**(unsigned int)*location*	Serializes *numInts* count of integers in *intBuffer* by encoding each integer, starting at the specified *location*, and replacing each corresponding integer encoding serially.