

<code>initWithCapacity:(unsigned int)capacity</code>	Initializes a newly allocated mutable data object, giving it enough memory to hold capacity bytes. Sets the length of the data object to 0.
<code>initWithLength:(unsigned int)length</code>	Initializes a newly allocated mutable data object, giving it enough memory to hold length bytes. Fills the object with zeroes up to length.
<code>-(void)increaseLengthBy:(unsigned int)extraLength</code>	Increases the length of a mutable data object by extraLength zero-filled bytes.
<code>-(void *)mutableBytes</code>	Returns a pointer to the bytes in a mutable data object, enabling you to modify the bytes.
<code>-(void)setLength:(unsigned int)length</code>	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.
<code>-(void)appendBytes:(const void *)bytes length:(unsigned int)length</code>	Appends length bytes to a mutable data object from the buffer bytes.
<code>-(void)appendData:(NSData *)other</code>	Appends the contents of the data object other to the receiver.
<code>-(void)replaceBytesInRange:(NSRange)aRange withBytes:(const void *)bytes</code>	Replaces the receiver's bytes located in aRange with bytes. Raises an NSRangeException if aRange is not within the range of the receiver's data.
<code>-(void)resetBytesInRange:(NSRange)aRange</code>	Replaces the receiver's bytes located in aRange with zeros. Raises an NSRangeException if aRange is not within the range of the receiver's data.
<code>-(void)serializeAlignedBytesLength:(unsigned int)length</code>	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 serializeInt:(int)value
atIndex:(unsigned int)index

Serializes the integer value by encoding it as a character representation and replaces the encoded value at the specified 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)index

Serializes numInts count of integers in intBuffer by encoding each integer, starting at the specified index, and replacing each corresponding integer encoded