)initWithBytes:(const void *)bytes length:(unsigned int)length	Initializes a newly allocated NSData object by putting in it length bytes of data copied from the buffer bytes.
)initWithBytesNoCopy:(void *)bytes length:(unsigned int)length	Initializes a newly allocated NSData object by putting in it length bytes of data from the buffer bytes.
)initWithContentsOfFile:(NSString *)path	Initializes a newly allocated NSData object by reading into it the data from the file specified by path.
)initWithContentsOfMappedFile:(NSString *)path	Initializes a newly allocated NSData object to contain the data residing in the mapped file path, assuming mapped files are available on the underlying operating system. If mapped files are not available, this method is identical to initWithContentsOfFile:.
)initWithData:(NSData *)data	Initializes a newly allocated NSData object by placing in it the contents of another NSData object, data.

nst void *)bytes	Returns a pointer to the object's contents. This method returns read-only access to the data.
SString *)description	Returns an NSString object that contains a hexadecimal representation of the the receiver's contents.
id)getBytes:(void *)buffer	Copies the receiver's contents into buffer.
id)getBytes:(void *)buffer length:(unsigned int)length	Copies length bytes of the receiver's contents into buffer.
id)getBytes:(void *)buffer	Copies into buffer the portion of the receiver's contents

	contents of the receiver, this method returns YES. If no
signed int)length	Returns the number of bytes contained in the receiver.
OOL)writeToFile:(NSString *)path atomically:(BOOL)useAuxiliaryFile	Writes the bytes in the receiving object to the file specifie by path. If useAuxiliaryFile is YES, the data is written then, assuming no errors occur, the backup file is renar intended file name.
signed int)deserializeAlignedBytesLengthAtCursor:(unsigned int*)cursor	Returns the length of the serialized bytes at the location re the bytes have been page-aligned, it also obtains the re information and adjusts the cursor. An invocation of th corresponding serializeAlignedBytesLength: invocatio
id)deserializeBytes:(void *)buffer length:(unsigned int)bytes atCursor:(unsigned int*)cursor	Deserializes bytes number of bytes in the buffer pointed at by buffer, places them internally starting at cursor, and advances the cursor.
id)deserializeDataAt:(void *)data ofObjCType:(const char *)type atCursor:(unsigned int*)cursor context:(id <NSObjCTypeSerializationCallBack>) callback	Deserializes the data pointed at by cursor, interpreting it by the Objective C type specifier type and writing it to the memory location referenced by data. If the data element is an object other than an instance of NSDictionary, NSArray, NSString, or NSData, a call callback can provide further definition of the object. A currently supported except union and void *. Pointers r
t)deserializeIntAtCursor:(unsigned int*)cursor	Deserializes and returns the integer encoded at cursor. Als
t)deserializeIntAtIndex:(unsigned int)index	Deserializes and returns the integer encoded at offset inde cursor.
id)deserializeInts:(int *)intBuffer count:(unsigned int)numInts atCursor:(unsigned int*)cursor	Deserializes numInts integers encoded at the location referenced by cursor and puts them in the buffer intBuffer. Also advances the cursor.
id)deserializeInts:(int *)intBuffer count:(unsigned int)numInts atIndex:(unsigned int)index	Deserializes numInts integers encoded at offset index and puts them in the buffer intBuffer. Does not advanc the cursor.