

`(id)encodeArrayOfObjCType:(const char *)types  
count:(unsigned int)count  
at:(const void *)array`

Encodes data of Objective C types listed in types having count elements residing at address array.

`(id)encodeBycopyObject:(id)anObject`

Overridden by subclasses to encode the supplied Objective C object so that a copy rather than a proxy of anObject is created upon decoding. NSCoder's implementation simply invokes encodeObject:.

`(id)encodeConditionalObject:(id)anObject`

Overridden by subclasses to conditionally encode the supplied Objective C object. The object should be encoded only if it is an intrinsic member of the larger data structure. NSCoder's implementation simply invokes encodeObject:.

`(id)encodeDataObject:(NSData *)data`

Encodes the NSData object data.

	invokes encodeObject:.
(id)encodeSize:(NSSize)size	Encodes the supplied size structure.
(id)encodeValueOfObjCType:(const char *)type at:(const void *)address	Encodes data of the specified Objective C type residing at address.
(id)encodeValuesOfObjCTypes:(const char *)types,...	Encodes values corresponding to the Objective C types list.
(id)decodeArrayOfObjCType:(const char *)types count:(unsigned)count at:(void *)address	Decodes data of Objective C types listed in type having count elements residing at address.
(NSData *)decodeDataObject	Decodes and returns an NSData object.
(id)decodeObject	Decodes an Objective C object.
(id)decodePropertyList	Decodes a property list (NSData, NSArray, NSDictionary).
(CGPoint)decodePoint	Decodes a point structure.
(CGRect)decodeRect	Decodes a rectangle structure.
(CGSize)decodeSize	Decodes a size structure.
(id)decodeValueOfObjCType:(const char *)type at:(void *)address	Decodes data of the specified Objective C type residing at address. The caller is responsible for releasing the resulting objects.
(id)decodeValuesOfObjCTypes:(const char *)types,...	Decodes values corresponding to the Objective C types list. You are responsible for releasing the resulting objects.
(NSZone *)objectZone	Returns the memory zone used by decoded objects. For instance, if this is the default memory zone, the one returned by NSDefaultRunLoopMode.
(id)setObjectZone:(NSZone *)zone	Sets the memory zone used by decoded objects. Instances of NSObject subclasses ignore this method.
(int)systemVersion	Returns the system version number as of the time the architecture was created.
(int)versionForClassName:(NSString *)className	Returns the version number of the class className as of the time the architecture was created.