

encodeArrayOfObjCType:count:at:
 encodeBycopyObject:
 encodeConditionalObject:
 encodeDataObject:
 encodeNXObject:

decodePropertyList:
decodeValueOfObjCType:at:
decodeValuesOfObjCTypes:

Managing Zones objectZone

setObjectZone:

Getting a Version systemVersion

versionForClassName:

(void)decodeArrayOfObjCType:(const char *)type 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.

(Object *)decodeNXObject

Decodes and returns an object that descends from Object.

(id)decodeObject

Decodes an Objective C object.

(id)decodePropertyList

Decodes a property list (NSData, NSArray, NSDictionary, or NSString objects).

(void)decodeValueOfObjCType:(const char *)type at:(void *)address

Decodes data of the specified Objective C type residing at address. You are responsible for releasing the object.

(void)decodeValuesOfObjCTypes:(const char *)types,...

Decodes values corresponding to the Objective C types listed in types argument list. You are responsible for releasing the resulting objects.

(void)encodeBycopyObject:(id)anObject

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

(void)encodeConditionalObject:(id)anObject

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

(void)encodeDataObject:(NSData *)data

Encodes the NSData object data.

(void)encodeNXObject:(Object *)object

Encodes an object that descends from Object.

(void)encodeObject:(id)anObject

Encodes the supplied Objective C object.

(void)encodePropertyList:(id)plist

Encodes the supplied property list (NSData, NSArray, NSDictionary, or NSString objects).

(void)encodeRootObject:(id)rootObject

Overridden by subclasses to start encoding an interconnected group of Objective C objects, starting with rootObject. NSCoder's implementation simply invokes encodeObject:.

(void)encodeValueOfObjCType:(const char *)type at:(const void *)address

Encodes data of the specified Objective C type residing at address.

(void)encodeValuesOfObjCTypes:(const char *)types,...

Encodes values corresponding to the Objective C types listed in types argument list.

Sets the memory zone used by decoded objects. Instances of NSCoder always use the default memory zone returned by `NSDefaultMallocZone()`, and so ignore this method.

`(unsigned int)systemVersion`

Returns the system version number as of the time the archive was created.

`(unsigned int)versionForClassName:(NSString *)className`

Returns the version number of the class `className` as of the time it was archived.