NSCoder

Inherits From: NSObject

Conforms To: NSCoding
NSObject

Declared In: foundation/NSCoder.h
foundation/NSGeometry.h

Encoding Data

- (void)**encodeArrayOfObjCType:**(const char *)*types*
count:(unsigned)*count*
at:(const void *)*array*
Serializes data of Objective C types listed in *types* having *count* elements residing at address *array*.

- (void)**encodeBycopyObject:**(id)*anObject*
Overridden by subclasses to serialize the supplied Objective C object so that a copy rather than a proxy of *anObject* is created upon deserialization. NSCoder's implementation simply invokes **encodeObject:**.

- (void)**encodeConditionalObject:**(id)*anObject*
Overridden by subclasses to conditionally serialize the supplied Objective C object. The object should be serialized only if it is an inherent member of the larger data structure. NSCoder's implementation simply invokes **encodeObject:**.

- (void)**encodeDataObject:**(NSData *)*data*
Serializes the NSData object *data*.

- (void)**encodeObject:**(id)*anObject*
Serializes the supplied Objective C object.

- (void)**encodePropertyList:**(id)*plist*
Serializes the supplied property list (NSData, NSArray, NSDictionary, or

NSString objects).

- (void)**encodePoint:**(NSPoint)*point* Serializes the supplied point structure.
- (void)**encodeRect:**(NSRect)*rect* Serializes the supplied rectangle structure.
- (void)**encodeRootObject:**(id)*rootObject* Overridden by subclasses to start the serialization of an interconnected group of Objective C objects, starting with *rootObject*. NSCoder's implementation simply invokes **encodeObject:**.
- (void)**encodeSize:**(NSSize)*size* Serializes the supplied size structure.
- (void)**encodeValueOfObjCType:**(const char *)*type* **at:**(const void *)*addr*ess Serializes data of Objective C type *type* residing at address *address*.
- (void)**encodeValuesOfObjCTypes:**(const char *)*types,...* Serializes values corresponding to the Objective C types listed in *types* argument list.

Decoding Data

- (void)**decodeArrayOfObjCType:**(const char *)*types* **count:**(unsigned)*count* **at:**(void *)*address* Deserializes data of Objective C types listed in *type* having *count* elements residing at address *address*.
- (NSData *)**decodeDataObject** Deserializes and returns an NSData object.
- (id)**decodeObject** Deserializes an Objective C object.
- (id)**decodePropertyList** Deserializes a property list (NSData, NSArray, NSDictionary, or NSString objects).
- (NSPoint)**decodePoint** Deserializes a point structure.
- (NSRect)**decodeRect** Deserializes a rectangle structure.
- (NSSize)**decodeSize** Deserializes a size structure.
- (void)**decodeValueOfObjCType:**(const char *)*type* **at:**(void *)*address* Deserializes data of Objective C type *type* residing at address *address*. You are responsible for releasing the resulting objects.
- (void)**decodeValuesOfObjCTypes:**(const char *)*types,...* Deserializes values corresponding to the Objective C types listed in *types* argument list. You are responsible for releasing the resulting objects.

Managing Zones

- (NSZone *)**objectZone**
  Returns the memory zone used by deserialized objects. For instances of NSCoder, this is the default memory zone, the one returned by **NSDefaultMallocZone()**.
- (void)**setObjectZone:**(NSZone *)*zone*
  Sets the memory zone used by deserialized objects. Instances of NSCoder always use the default memory zone, the one returned by **NSDefaultMallocZone()**, and so ignore this method.

Getting a Version

- (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.