

# NSUnarchiver

**Inherits From:** NSCoder : NSObject

**Conforms To:** NSCoding  
NSObject

**Declared In:** foundation/NSArchiver.h

## Initializing an NSUnarchiver

- (id)**initWithReadingWithData:**(NSData \*)*data*      Initializes an NSUnarchiver object from data object *data*.

## Decoding Objects

+ (id)**unarchiveObjectWithData:**(NSData \*)*data*      Decodes an archived object stored in *data*.

+ (id)**unarchiveObjectWithFile:**(NSString \*)*path*      Decodes an archived object stored in the file *path*.

- (void)**decodeArrayOfObjCType:**(const char \*)*itemType*  
          **count:**(unsigned iny)*count*      Decodes an *array* of *count* data elements of the same  
          **at:**(void \*)*array*      Objective C data *itemType*.

## Managing an NSUnarchiver

- (BOOL)**isAtEnd**      Returns YES if the end of data is reached, NO if more data follows.

- (NSZone \*)**objectZone**      Returns the allocation zone for the unarchiver object.

- (void)**setObjectZone:**(NSZone \*)*zone*      Sets the allocation zone for the unarchiver object to *zone*. If *zone* is **nil**, it sets it to the default zone.

- (unsigned int)**systemVersion**      Returns the system version number for the unarchived data.

## Substituting One Class For Another

+ (NSString \*)**classNameDecodedForArchiveClassName:**(NSString \*)*inArchiveName*

Returns the string object representing the class name used to archive instances of the class (*inArchiveName*). This may not be the original class name but another name encoded with NSArchiver's **encodeClassName:intoClassName**.

- + (void)**decodeClassName:**(NSString \*)*inArchiveName*  
    **asClassName:**(NSString \*)*trueName*      Decodes from the archived data a class name (*inArchiveName*) substituted for the real class name (*trueName*). This method enables easy conversion of unarchived data when there are name changes in classes.
- (NSString \*)**classNameDecodedForArchiveClassName:**(NSString \*)*inArchiveName*  
    Returns the string object representing the class name used to archive instances of the class (*inArchiveName*). This may not be the original class name but another name encoded with NSArchiver's **encodeClassName:intoClassName**.
- (void)**decodeClassName:**(NSString \*)*inArchiveName*  
    **asClassName:**(NSString \*)*trueName*      Decodes from the archived data a class name (*inArchiveName*) substituted for the real class name (*trueName*). This method enables easy conversion of unarchived data when there are name changes in classes.