

NSDeserializer

Inherits From:	NSObject
Conforms To:	NSObject (NSObject)
Declared In:	Foundation/NSSerialization.h

Class Description

The NSDeserializer class declares methods that convert an abstract representation of a property list (as contained in an NSData object) into a graph of property list objects in memory. The NSDeserializer class object itself provides these methods; you don't create instances of NSDeserializer.

Options to these methods allow you to specify that container objects (arrays or dictionaries) in the resulting graph be mutable or immutable; that deserialization begin at the start of the data or from some position within it; or that deserialization occur lazily, so that a property list is deserialized only if it is actually going to be accessed. See the NSSerializer specification for more information on serialization.

Deserialization Into Property Lists

+ (id) deserializePropertyListFromData: (NSData *) <i>data</i>	
atCursor: (unsigned int*) <i>cursor</i>	Returns a property list object corresponding to the abstract
mutableContainers: (BOOL) <i>mutable</i>	representation in <i>data</i> at the location <i>cursor</i> . If <i>mutable</i> is YES and the object is

a dictionary or an array, the re-composed object is made mutable. Returns **nil** if the object is not a valid one for property lists.

+ (id)**deserializePropertyListFromData:**(NSData *)*data*

mutableContainers:(BOOL)*mutable*

Returns a property list object corresponding to the abstract representation in *data*. If *mutable* is YES and the object is a dictionary or an array, the re-composed object is made mutable. Returns **nil** if the data doesn't represent a property list.

+ (id)**deserializePropertyListLazilyFromData:**(NSData *)*data*

atCursor:(unsigned int*)*cursor*

length:(unsigned int)*length*

mutableContainers:(BOOL)*mutable*

Returns a property list from *data* at location *cursor*.

The deserialization proceeds lazily. That is, if the data at the specified location has a length greater than *length*, a proxy is substituted for the actual property list as long as the constituent objects of that property list are not being accessed. If *mutable* is YES and the object is a dictionary or an array, the re-composed object is made mutable. Returns **nil** if the data doesn't represent a property list.