NSMutableCopying

Adopted By: <<*various*>>

Declared In: foundation/NSObject.h

Protocol Description

A class that defines an “immutable vs. mutable” distinction adopts this protocol to allow mutable copies of its instances to be made. A mutable copy of an object is usually a *shallow copy* (as opposed to the *deep copy* defined in the NSCopying protocol specification). The original and its copy share references to the same instance variables, so that if a component of the copy is changed, for example, that change is reflected in the original.

A class that doesn’t define an “immutable vs. mutable” distinction but that needs to offer both deep and shallow copying shouldn’t adopt this protocol. The NSCopying methods should by default be assumed to produce deep copies; the class can then also implement methods to produce shallow copies.

The copied instance returned by the methods in this protocol are not autoreleased.

Instance Methods

mutableCopy

– **mutableCopy**

Invokes **mutableCopyWithZone:** with the same memory zone as the receiver. Subclasses should implement their own versions of **mutableCopyWithZone:**, not **mutableCopy**, to define class-specific copying.

See also: – **copy** (NSCopying protocol)

mutableCopyWithZone:

– **mutableCopyWithZone:**(NXZone *)*zone*

Returns a new instance that's an exact copy of the receiver. Memory for the new instance is allocated from *zone*. The copy returned is mutable if the consideration "immutable vs. mutable" applies to the receiving object. The returned copy is not autoreleased.

This method usually creates only one new object, performing a shallow copy of the receiver. If the receiver has instance variables that point to other objects, the instance variables in the copy will point to the same objects. The values of the instance variables are copied, but the objects they point to are not.

See also: – **copyWithZone:** (NSMutableCopying protocol)