

# NSValue

<b>Inherits From:</b>	NSObject
<b>Conforms To:</b>	NSCopying NSMutableCopying
<b>Declared In:</b>	foundation/NSValue.h

## Class Description

NSValue objects provides an object-oriented wrapper for the data types defined in standard C and Objective C.

The NSValue class declares the programmatic interface to an object that contains a C data type. It provides methods for creating value objects that contain values of a specified data type, pointers, and other objects.

The NSValue class is often used to put Objective C and standard C data types into collections that require objects, such as NSArray. When a value object is instantiated, it is encoded with the specified data type.

## Instance Variables

None declared in this class.

## Adopted Protocols

NSCopying	- copy - copyWithZone:
NSMutableCopying	- mutableCopy - mutableCopyWithZone:

## Method Types

Allocating and initializing	+ value:withObjCType: + valueWithNonretainedObject: + valueWithPointer:
Accessing data	- getValue - nonretainedObjectValue - objCType - pointerValue

## Class Methods

### **value:withObjCType:**

+ (NSValue \*)**value:(const void\*)value withObjCType:(const char \*)type**

Creates and returns a value object containing *value* of the Objective C type *type* in the specified zone.

This excerpt creates and initializes a value object *valObj* and encodes its data to be of the type NSRange.

```
NSRange myRange = {4, 10};
NSValue *valObj = [NSValue value:&myRange
                  withObjCType:@encode(NSRange)];
```

### **valueWithNonretainedObject:**

+ (NSValue \*)**valueWithNonretainedObject:anObject**

Creates and returns a value object containing the object *anObject* without retaining *anObject*. The statement **[NSValue valueWithNonretainedObject:obj]** is equivalent to the statement **[NSValue value:&obj withObjCType:@encode(void \*)]**.

### **valueWithPointer:**

+ (NSValue \*)**valueWithPointer:**(void \*)*pointer*

Creates and returns a value object that contains a pointer. The statement **[NSValue valueWithPointer:pointer]** is equivalent to the statement **[NSValue value:&pointer withObjectType:@encode(void \*)]**.

**See also:** - **data** (NSData)

## **Instance Methods**

### **getValue:**

- (void)**getValue:**(void \*)*value*

Copies the data in a value object into the caller's variable *value*.

This excerpt creates a value object *valObj* and then uses **getValue:** to copy a pointer to *valObj*'s data into *valPtr*.

```
NSValue* valObj;  
int theInt = 123;  
const void *myVal = &theInt;  
void *valPtr;  
valObj = [NSValue  
    value:myVal  
    withObjectType:@encode(int*)];  
[valObj getValue:&valPtr];
```

**See also:** - **bytes** (NSData), - **getBytes:length:** (NSData), - **getBytes:range:** (NSData)

### **nonretainedObjectValue**

- **nonretainedObjectValue**

Returns a non-retained object from a value object.

**See also:** - **bytes** (NSData), - **getBytes:length:** (NSData), - **getBytes:range:** (NSData)

### **objCType**

- (const char \*)**objCType**

Returns the Objective C type of the data contained in a value object.

This excerpt assigns to *type* a const char\* representation of the Objective C type for the data contained in *valObj*

```
const char* type;
NSValue* valObj;
...
type = [valObj objCType];
```

**See also:** - **bytes** (NSData), - **getBytes:length:** (NSData), - **getBytes:range:** (NSData)

### **pointerValue**

- (void \*)**pointerValue**

Returns the value pointed to by a pointer contained in an value object.

**See also:** - **initWithBytesNoCopy:length:** (NSData)