

# Object (MiscRuntimeExtensions)

**Declared In:** <misckit/MiscRuntimeExtensions.h>

## Category Description

This category methods to the Object class which extend the runtime. Use **;MiscRuntimeExtensions.rtf;+instanceSize;¬+instanceSize** to determine the size of an instance of a particular class. Use **;MiscRuntimeExtensions.rtf;becomeClass;¬±becomeClass**: to change the class of an object. Use **±compare**: to compare two different objects, as you would with **±isEqual**.

## Method Types

- Determining the size of an object **;MiscRuntimeExtensions.rtf;+instanceSize;¬+instanceSize**  
**instanceSize**  
**;MiscRuntimeExtensions.rtf;instanceSize;¬± instanceSize**  
**;MiscRuntimeExtensions.rtf;sizeOf;¬± sizeOf**
- Changing an object's class **;MiscRuntimeExtensions.rtf;becomeClass;¬- becomeClass**:

Comparing objects

`;MiscRuntimeExtensions.rtf;compare;` compare:

## Class Methods

`+instanceSize;` `¬instanceSize`

+ (int)`instanceSize`

Returns the size of an instance of the receiving class. Note that an Objective-C class is represented by the compiler as a C structure. This method returns the size of that structure, but does *not* return any information about data that instance variables may point to, which means that memory allocated by this object is *not* taken into consideration.

**See also:** `;MiscRuntimeExtensions.rtf;instanceSize;` `¬instanceSize` and `;MiscRuntimeExtensions.rtf;sizeOf;` `¬sizeOf`

## Instance Methods

`becomeClass;` `¬becomeClass`

± `becomeClass:aClass`

Changes the receiver into an object of class `aClass`, where `aClass` is an Objective-C Class object. Returns **self** if successful or *nil* if unsuccessful. Note that you cannot change a class into a class of larger size, though you can change a class into a class of a *smaller* size. (You can compare sizes using the `;MiscRuntimeExtensions.rtf;instanceSize;` `¬instanceSize` method.) One known bug is that after changing to a class of a smaller size you cannot change back to the original class. This happens because the old size information is thrown out. If you *know* that you can move to a larger class size safely, then you can call the `±_becomeClass:` version instead to bypass range checking. **Warning:** This method is extremely dangerous. If used without knowing what it does it may likely cause crashes. Despite all this warning, this method can be invaluable and useful in some situations. The user simply must take all responsibility for the outcome of this

method if they use it, as the author disclaims any and all responsibility!

**See also:** `± class` (Object)

### **compare;¬compare**

`± (int)compare:anObject`

Attempts to compare two objects. Returns zero (0) if they compare as <sup>a</sup>equal<sup>o</sup>, one (1) if the receiver is <sup>a</sup>greater<sup>o</sup> than *anObject* or a negative one (-1) if the receiver is <sup>a</sup>less<sup>o</sup> than *anObject*. The default implementation attempts to compare string values (as obtained with `±stringValue`) of two objects. Subclasses should override this if there is a more useful comparison possible.

**See also:** `± isEqual:` (Object), `strcmp()`

### **instanceSize;¬instanceSize**

`± (int)instanceSize`

Returns the size of an instance of the receiving class. Note that an Objective±C class is represented by the compiler as a C structure. This method returns the size of that structure, but does *not* return any information about data that instance variables may point to, which means that memory allocated by this object is *not* taken into consideration. This method should not be overridden. If the implementation is changed, methods which depend upon this behavior will break, such as `;MiscRuntimeExtensions.rtf;becomeClass;¬±becomeClass.` If a method which gives complete size information is desired, use `;MiscRuntimeExtensions.rtf;sizeOf;¬±sizeOf.`

**See also:** `;MiscRuntimeExtensions.rtf;+instanceSize;¬+ instanceSize` and `;MiscRuntimeExtensions.rtf;sizeOf;¬± sizeOf`

### **sizeOf;¬sizeOf**

`± (int)sizeof`

Returns the size of an instance. This may be overridden to take into account the space taken up by allocated memory such as string pointers and objects pointed to (in composite objects).

**See also:** `;MiscRuntimeExtensions.rtf;+instanceSize;¬+ instanceSize` and `;MiscRuntimeExtensions.rtf;instanceSize;¬± instanceSize`