Functions

Class Functions

Create a new instance of a class

id **class_createInstance(**Class *aClass*, unsigned int *indexedIvarBytes***)**
id **class_createInstanceFromZone(**Class *aClass*, unsigned int *indexedIvarBytes*, NXZone **zone***)**

Get the class template for an instance variable

Ivar **class_getInstanceVariable(**Class *aClass*, const char **variableName***)**

Get, add, and remove methods

Method **class_getInstanceMethod(**Class *aClass*, SEL *aSelector***)**
Method **class_getClassMethod(**Class *aClass*, SEL *aSelector***)**
void **class_addMethods(**Class *aClass*, struct objc_method_list **methodList***)**
void **class_removeMethods(**Class *aClass*, struct objc_method_list **methodList***)**

Pose as the superclass

Class **class_poseAs(**Class *theImposter*, Class *theSuperclass***)**

Set and get the class version

void **class_setVersion(**Class *aClass*, int *versionNumber***)**
int **class_getVersion(**Class *aClass***)**

System Functions

Manage run-time structures

id **objc_getClass(**const char **aClassName***)**
id **objc_lookUpClass(**const char **aClassName***)**
id **objc_getMetaClass(**const char **aClassName***)**
NXHashTable * **objc_getClasses(**void**)**
void **objc_addClass(**Class *aClass***)**
Module * **objc_getModules(**void**)**

Dynamically load and unload classes

long **objc_loadModules(**char **files*[], NXStream **stream*, void (**callback*)(Class, Category), struct mach_header ***header*, char ***debugFilename***)**
long **objc_unloadModules(**NXStream **stream*, void (**callback*)(Class, Category)**)**

Send messages at run time

id **objc_msgSend(**id *theReceiver*, SEL *theSelector*, ...**)**
id **objc_msgSendSuper(**struct objc_super **superContext*, SEL *theSelector*, ...**)**
id **objc_msgSendv(**id *theReceiver*, SEL *theSelector*, unsigned int *argSize*, marg_list *argFrame***)**

Make the run-time system thread safe

void **objc_setMultithreaded(**BOOL *flag***)**

Object Functions

Manage object memory

id **object_dispose(**Object **anObject***)**
id **object_copy(**Object **anObject*, unsigned int *indexedIvarBytes***)**
id **object_copyFromZone(**Object **anObject*, unsigned int *indexedIvarBytes*, NXZone **zone***)**
id **object_realloc(**Object **anObject*, unsigned int *numBytes***)**
id **object_reallocFromeZone(**Object **anObject*, unsigned int *numBytes*, NXZone **zone***)**

Return the class name

const char * **object_getClassName(**id *anObject***)**

Set and get instance variables

Ivar **object_setInstanceVariable(**id *anObject*, const char **variableName*, void **value***)**
Ivar **object_getInstanceVariable(**id *anObject*, const char **variableName*, void ***value***)**

Return a pointer to an object's extra memory

void * **object_getIndexedIvars(**id *anObject***)**

Method Functions and Macros

Get information about a method

unsigned int **method_getNumberOfArguments(**Method *aMethod***)**
unsigned int **method_getSizeOfArguments(**Method *aMethod***)**
unsigned int **method_getArgumentInfo(**Method *aMethod*, int *index*, const char ***type*, int **offset***)**

Examine and alter method argument values

type-name **marg_getValue(**marg_list *argFrame*, int *offset*, *type-name***)**
type-name * **marg_getRef(**marg_list *argFrame*, int *offset*, *type-name***)**
void **marg_setValue(**marg_list *argFrame*, int *offset*, *type-name*, *type-name value***)**

Selector Functions

Match method names with method selectors

SEL **sel_getUid(**const char ***aName*)**
const char * **sel_getName(**SEL *aSelector*)**

Determine whether a selector is valid

BOOL **sel_isMapped(**SEL *aSelector*)**

Register a method name

SEL **sel_registerName(**const char ***aName*)**