

Functions

Class Functions

Create a new instance of a class

id	class_createInstance (Class <i>aClass</i> , unsigned int <i>indexedIvarBytes</i>)
id	class_createInstanceFromZone (Class <i>aClass</i> , unsigned int <i>indexedIvarBytes</i> , NXZone <i>*zone</i>)

Get the class template for an instance variable

Ivar	class_getInstanceVariable (Class <i>aClass</i> , const char <i>*variableName</i>)
------	---

Get, add, and remove methods

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

Pose as the superclass

Class	class_poseAs (Class <i>theImposter</i> , Class <i>theSuperclass</i>)
-------	--

Set and get the class version

void	class_setVersion (Class <i>aClass</i> , int <i>versionNumber</i>)
int	class_getVersion (Class <i>aClass</i>)

System Functions

Manage run-time structures

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

Dynamically load and unload classes

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

Send messages at run time

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

Make the run-time system thread safe

void	objc_setMultithreaded (BOOL <i>flag</i>)
------	--

Object Functions

Manage object memory

id	object_dispose (Object * <i>anObject</i>)
id	object_copy (Object * <i>anObject</i> , unsigned int <i>indexedIvarBytes</i>)
id	object_copyFromZone (Object * <i>anObject</i> , unsigned int <i>indexedIvarBytes</i> , NXZone * <i>zone</i>)
id	object_realloc (Object * <i>anObject</i> , unsigned int <i>numBytes</i>)
id	object_reallocFromZone (Object * <i>anObject</i> , unsigned int <i>numBytes</i> , NXZone * <i>zone</i>)

Return the class name

const char *	object_getClassName (id <i>anObject</i>)
--------------	--

Set and get instance variables

Ivar	object_setInstanceVariable (id <i>anObject</i> , const char * <i>variableName</i> , void * <i>value</i>)
Ivar	object_getInstanceVariable (id <i>anObject</i> , const char * <i>variableName</i> , void ** <i>value</i>)

Return a pointer to an object's extra memory

void *	object_getIndexedIvars (id <i>anObject</i>)
--------	---

Method Functions and Macros

Get information about a method

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

Examine and alter method argument values

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

Selector Functions

Match method names with method selectors

SEL	<code>sel_getUid(const char *<i>aName</i>)</code>
const char *	<code>sel_getName(SEL <i>aSelector</i>)</code>

Determine whether a selector is valid

BOOL	<code>sel_isMapped(SEL <i>aSelector</i>)</code>
------	---

Register a method name

SEL	<code>sel_registerName(const char *<i>aName</i>)</code>
-----	---