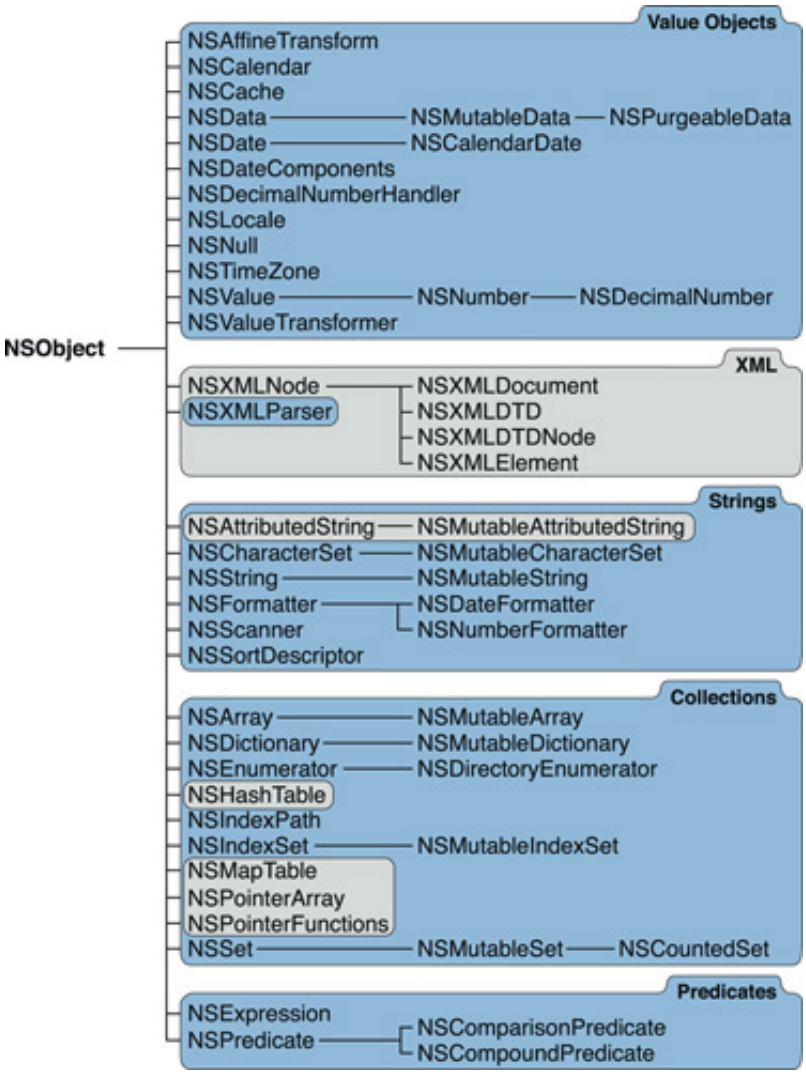
Foundation Kit

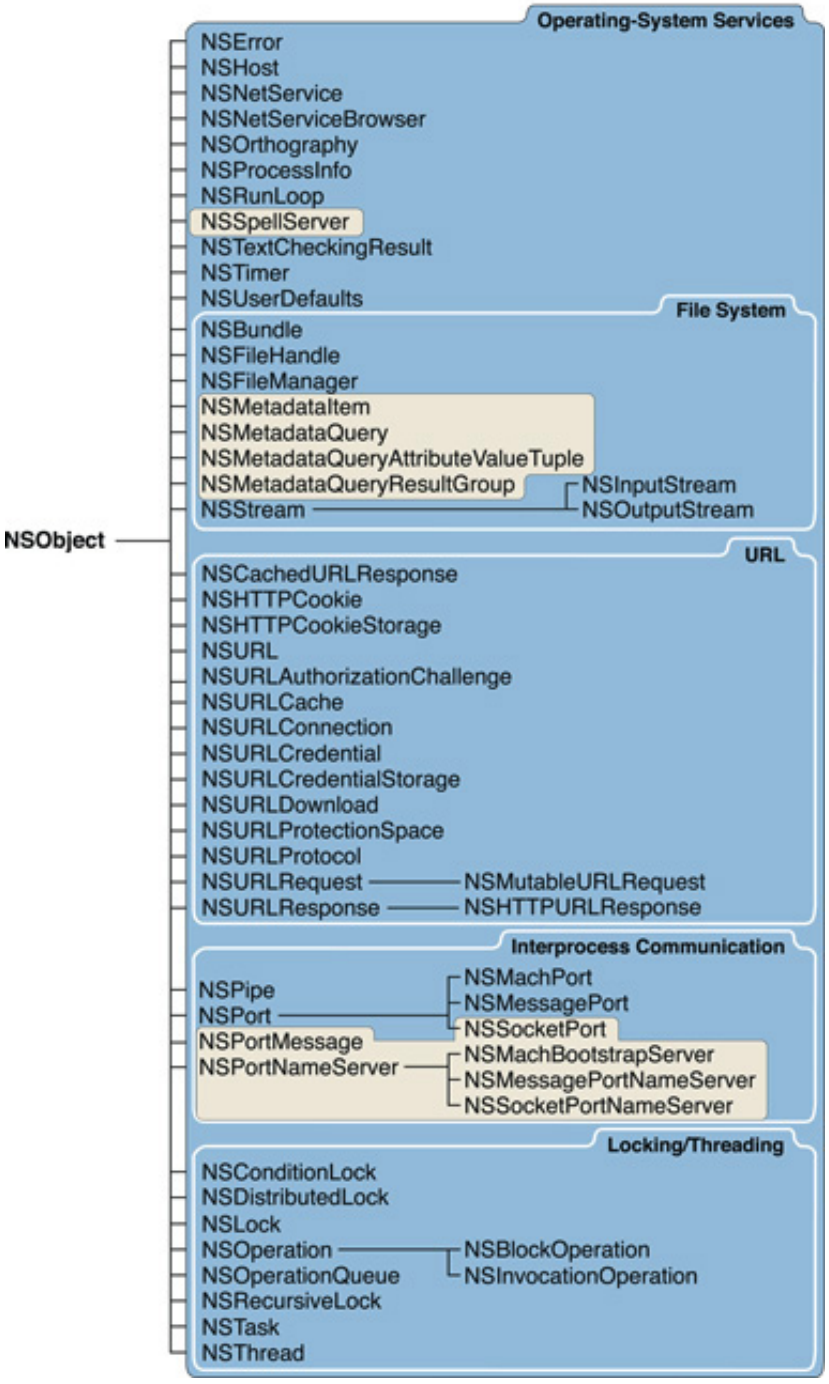
- Foundation是UIKit和AppKit这两类UI框架的基础
- 因为它不包含UI对象，所以，其对象可以在iOS和Mac OS X应用中兼容
- Foundation框架是以另一框架CoreFoundation为基础创建的

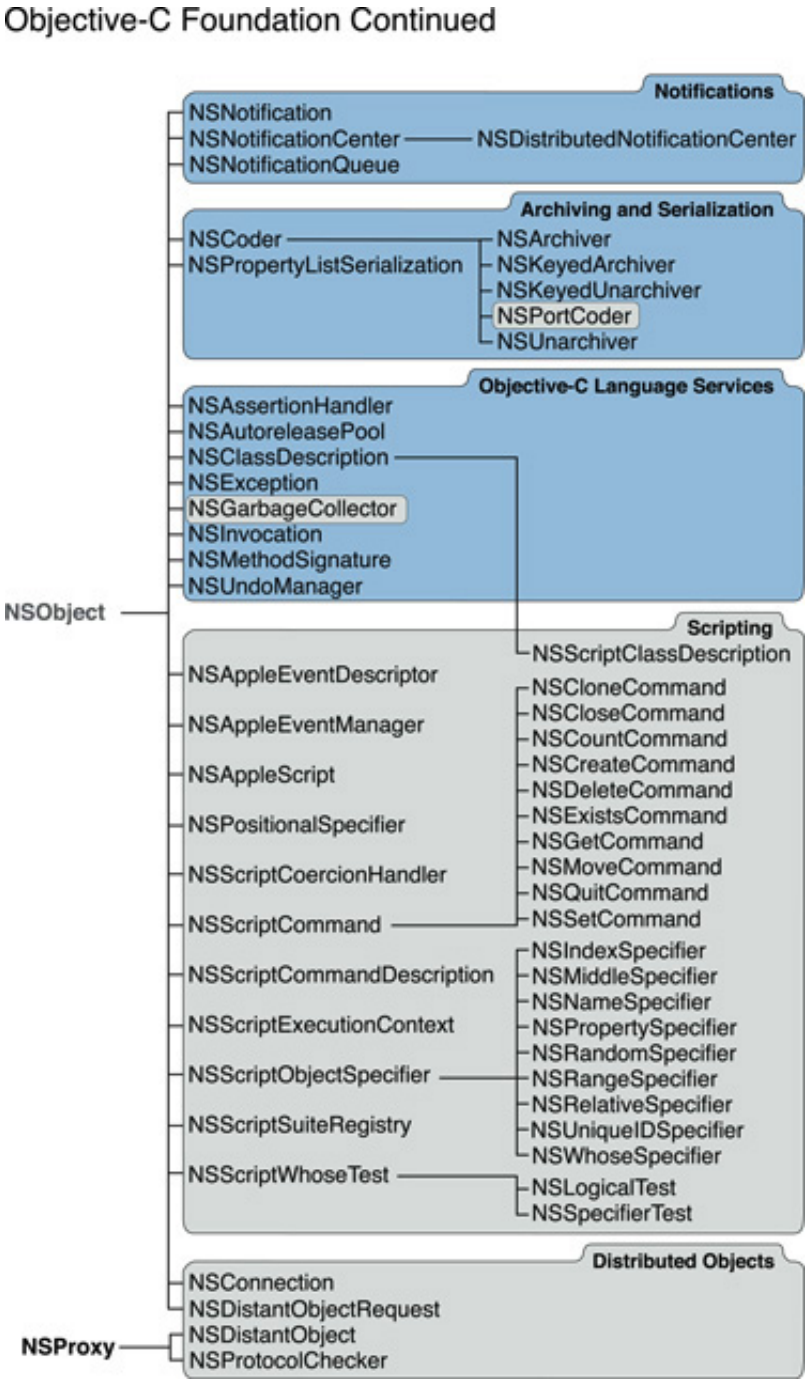
Foundation框架图

灰色的是iOS不支持的，是OS X系统的



Objective-C Foundation Continued





Foudation对象主要分类

- 值对象
- 集合
- 操作系统服务 包括三个方面: `文件系统、URL、进程间通讯`。
- 通知
- 归档和序列化
- 表达式和条件判断

- Objective-C语言服务

一些有用的数据类型

- 范围（NSRange）

```
typedef struct _NSRange
{
    NSUInteger location;
    NSUinteger length;
} NSRange;
```

创建NSRange的三种方式：

```
1. 直接给字段赋值
   NSRange range;
   range.location = 17;
   range.length = 4;
2. C语言的聚合结构赋值机制（花括号）
   NSRange range = {17, 4};
3. cocoa提供的NSMakeRange()函数
   NSRange range = NSMakeRange(17, 4);
```

- 几何数据类型 (这个属于CoreGraphics框架)
    - CGPoint
    - CGSize
    - CGRect

Foundation的一些常用类

1. 字符串
    - NSString

    创建字符串：

    ```
     NSString *str = @"Hello,world";
     NSString *height = [NSString stringWithFormat:@"Your height is %d
    feet, %d
                         inches", 5, 11];
    ```

    实例方法：

    字符串长度:

```
- (NSUInteger) length;
```

字符串比较:

```
- (BOOL) isEqualToString:(NSString *)aString;
- (NSComparisonResult)compare:(NSString *)string;
- (NSComparisonResult)compare:(NSString *)string options:
    (NSStringCompareOptions)mask;
```

2. 集合大家族
    - NSArray、NSMutableArray
    - NSEnumerator
    - NSDictionary、NSMutableDictionary

创建字典:

```
1. + (id) dictionaryWithObjectsAndKeys: (id) firstObject, ...;
2. @{key:value,...};
```

访问字典中的对象:

```
1. - (id) objectForKey: (id) aKey;
2. dict[key];
```

创建可变字典:

```
+(id)dictionaryWithCapacity: (NSUInteger) numItems;
```

添加元素:

```
- (void)setObject:(id)anObject forKey:(id)aKey;
```

删除元素:

```
- (void)removeObjectForKey: (id) aKey;
```

3. 其他数值对象
    - NSNumber

装箱：

创建NSNumber对象(将基本数据类型封装成NSNumber对象)

```
1.
   + (NSNumber *) numberWithChar: (char) value;
   + (NSNumber *) numberWithInt: (int) value;
   + (NSNumber *) numberWithFloat: (float) value;
   + (NSNumber *) numberWithBool: (BOOL) value;
2.
   NSNumber *number;
   number = @'X';     //字符型
   number = @12345;   //整型
   number = @12345ul; //无符号整型
   number = @123.45f; //浮点型
   number = @YES;     //布尔值
```

开箱：

从NSNumber对象中取出基本数据类型

```
- (char) charValue;
- (int) intValue;
- (float) floatValue;
- (BOOL) boolValue;
- (NSString *)stringValue;
```

- NSValue

创建NSValue对象

```
 + (NSValue *) valueWithBytes: (const void *)value objType:(const char
*)type;
```

提取NSValue中的数值

```
 - (void)getValue:(void *)buffer;
```

Cocoa提供的将常用struct型数据与NSValue互相转换的便捷方法

```
+ (NSValue *)valueWithPoint:(NSPoint)aPoint;
+ (NSValue *)valueWithSize:(NSSize)size;
+ (NSValue *)valueWithRect:(NSRect)rect;

- (NSPoint)pointValue;
- (NSSize)sizeValue;
- (NSRect)rectValue;
```

- NSNull

创建NSNull对象

```
+ (NSNull *) null;
```