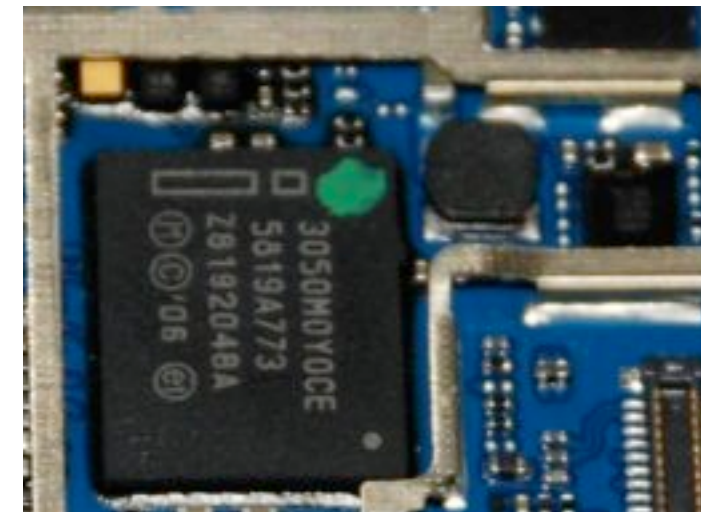


Advanced Mobile Solution AMS

Cours 3 Persistence sur iPhone



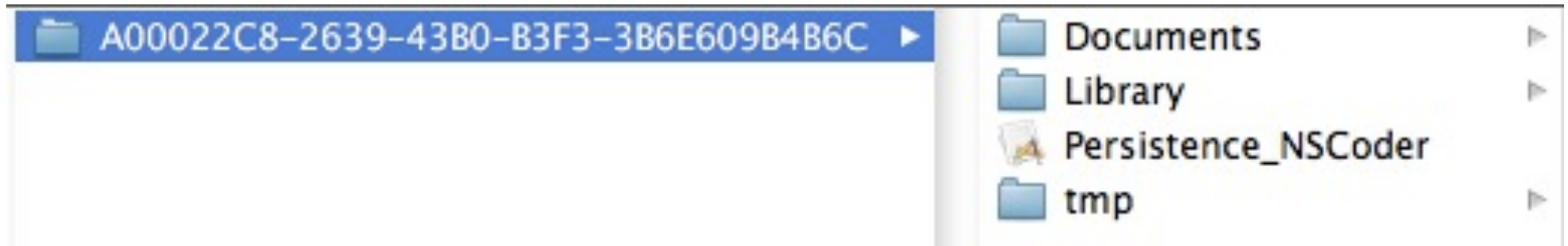
Sommaire:

persitence sur *iPhone*

- **Structure des fichiers**
- **Simple écriture/lecture dans un fichier**
- **Property list (.plist) + Appli. settings & User Pref.**
- **Archivage clé-valeur & NSCoder**
- ~~**SQLite via Core Data**~~
- ~~**SQLite via SQL**~~

Structure des fichiers

/Users/wagen/Library/Application Support/iPhone Simulator/4.1/Applications/
A00022C8-2639-43B0-B3F3-3B6E609B4B6C/Documents/me.txt



/var/mobile/Applications/6117C930-F5AE-4C9C-8AD0-74D098C1EB0F/Documents/me.txt
Chemin_par_défaut /me.txt

```
- (NSString *) documentDirectoryPath
{
    NSArray *paths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
    NSUserDomainMask, YES);

    return [paths objectAtIndex: 0];
}
```

Chemin par défaut



Creating Paths and Locating Directories

Exemple 1

```
/var/mobile/Applications/6117C930-F5AE-4C9C-8AD0-74D098C1EB0F/Documents/MyFile  
Chemin_par_défaut /MyFile
```

```
NSArray *paths = NSSearchPathForDirectoriesInDomains(  
    NSDocumentDirectory,  
    NSUserDomainMask, YES);  
  
NSString *theCompletePath =  
[[paths objectAtIndex: 0] stringByAppendingPathComponent: @"MyFile"];
```

Exemple 2

```
NSString *path2 = [NSHomeDirectory()  
    stringByAppendingPathComponent:@"Documents/MyFile2.txt"];
```

writeToFile: / stringWithContentsOfFile:

écriture (`NSString *stringToBeSaved = @"Voilà un texte à sauvegarder !"`)

```
[stringToBeSaved writeToFile: aFilePath atomically: YES encoding::NSUTF8StringEncoding error: NULL];
```

lecture

```
NSString *stringToBeRestored = [NSString stringWithContentsOfFile: aFilePath encoding::NSUTF8StringEncoding error: NULL];
```

Parfait pour stocker des données “simples” (NSString)

```
@interface Person : NSObject  
{
```

```
    NSString *name;  
    NSDate *birthDate;  
    float weight;  
    BOOL married;  
}
```

... mais “inventer” votre propre format

```
NSString *concatenatedPerson = [NSString stringWithFormat: @"%@|%@|%.1f|%.1d",  
name, [[self dateFormatter] stringFromDate: birthDate], weight, married];
```

voir exercice

Persistence_File.xcodeproj

The screenshot shows an iPhone application interface. At the top, the status bar displays "Carrier", a Wi-Fi signal icon, the time "12:52 AM", and a battery level icon. Below the status bar, there is a form with three fields: "Name" with the value "Toto", "Weight" with the value "50.2", and "Married" with a toggle switch set to "ON". Below the form is a date picker showing a list of months and years. The date picker is currently set to "August 7 1992". The date picker has five rows of data: "August 7 1992", "September 8 1993", "October 9 1994", "November 10 1995", and "December 11 1996". At the bottom of the screen, there are two buttons: "Save" and "Restore".

Month	Day	Year
August	7	1992
September	8	1993
October	9	1994
November	10	1995
December	11	1996

Save Restore

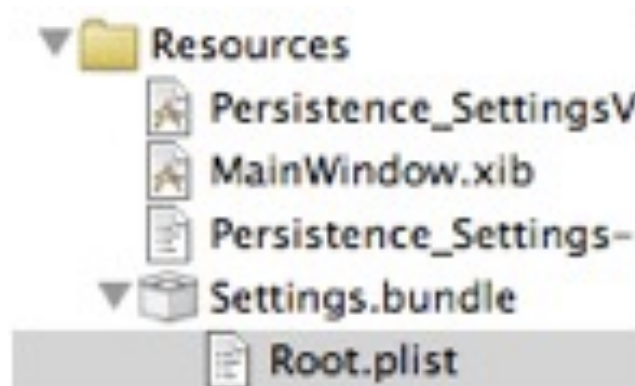
Exercice:

Persistence_File.xcodeproj

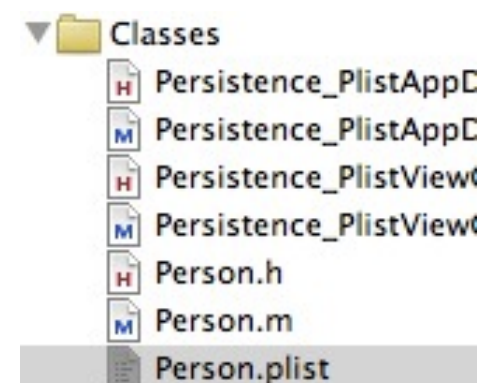
Property list (.plist)

contient des objets correspondants à des clés. Ces fichiers sont au format XML et sont utilisés pour 2 fonctionnalités différentes :

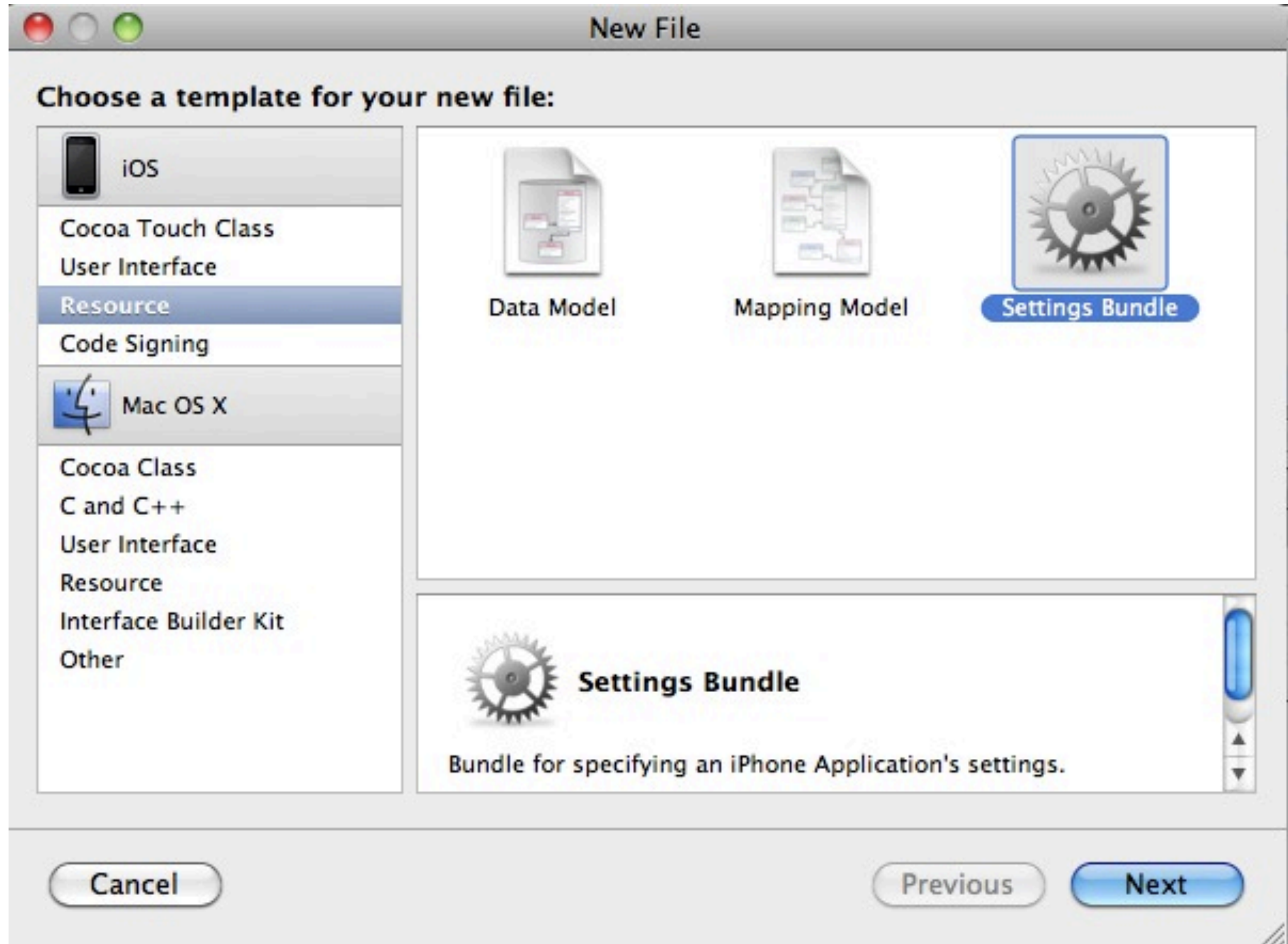
- Gestion des préférences utilisateur (=Application settings, user default, ...)



- Stockage d'information dans son propre fichier .plist



Ajouter un Settings Bundle



Ajouter un Settings.bundle

The screenshot displays the Xcode IDE with the 'Persistence_Settings' project selected. The left sidebar shows the project structure, including the 'Settings.bundle' folder. The main editor area shows the 'Root.plist' file, which is a dictionary containing two items: 'Item 0' and 'Item 1'. 'Item 0' is a dictionary with 'Type' set to 'PSGroupSpecifier' and 'Title' set to 'Group'. 'Item 1' is a dictionary with 'Type' set to 'PSTextFieldSpecifier', 'Title' set to 'Name', 'Key' set to 'name_preference', 'DefaultValue' set to 'Toto', 'IsSecure' set to 'false', 'KeyboardType' set to 'Alphabet', 'AutocapitalizationType' set to 'None', and 'AutocorrectionType' set to 'No'.

Key	Type	Value
Root	Dictionary	(2 items)
StringsTable	String	Root
PreferenceSpecifiers	Array	(2 items)
Item 0	Dictionary	(2 items)
Type	String	PSGroupSpecifier
Title	String	Group
Item 1	Dictionary	(8 items)
Type	String	PSTextFieldSpecifier
Title	String	Name
Key	String	name_preference
DefaultValue	String	Toto
IsSecure	Boolean	<input type="checkbox"/>
KeyboardType	String	Alphabet
AutocapitalizationType	String	None
AutocorrectionType	String	No

On the right, a preview of the settings app is shown. It features a 'Settings' button and a section titled 'Persistence_Settings'. Under the 'Group' heading, there is a text field labeled 'Name' with the value 'Toto' entered.

Ecriture/lecture Settings

- Ref: **Property List Programming Guide**

écriture



A defaults database is created automatically for each user by `standardUserDefaults`.

```
[[NSUserDefaults standardUserDefaults]  
    setValue: @"Tototest" forKey: @"Name_preference"];
```

```
[[NSUserDefaults standardUserDefaults] synchronize];
```

lecture

```
[[NSUserDefaults standardUserDefaults] stringForKey: @"Name"]
```

```

- (void) manageUserDefaults
{
    NSString *pathStr = [[NSBundle mainBundle] bundlePath];
    NSString *settingsBundlePath = [pathStr stringByAppendingPathComponent:
@"Settings.bundle"];
    NSString *finalPath = [settingsBundlePath stringByAppendingPathComponent:
@"Root.plist"];

    NSDictionary *settingsDict = [NSDictionary dictionaryWithContentsOfFile: finalPath];
    NSArray *prefSpecifierArray = [settingsDict objectForKey: @"PreferenceSpecifiers"];

    NSMutableDictionary *appDefaults = [NSMutableDictionary dictionary];
    NSString *nameDefaultValue;

    NSDictionary *prefItem;

    for (prefItem in prefSpecifierArray)
    {
        NSString *keyValueStr = [prefItem objectForKey: @"Key"];
        id defaultValue = [prefItem objectForKey: @"DefaultValue"];

        // Retrieve the default value for name_preference (only at first launch)
        if ([keyValueStr isEqualToString: @"name_preference"] &&
            ([[NSUserDefaults standardUserDefaults] stringForKey: @"name_preference"] == nil))
        {
            nameDefaultValue = defaultValue;
            [appDefaults setObject: nameDefaultValue forKey: @"name_preference"];
        }
    }

    // Set the settings in the UserDefaults and synchronize
    [[NSUserDefaults standardUserDefaults] registerDefaults: appDefaults];
    [[NSUserDefaults standardUserDefaults] synchronize];
}

```

Persistence pour des objets

- Archivage “Model Objects”, ou “NSCoding”
ou “clé-valeur”

Archivage clé-valeur & NSCoder

- Pour archiver *NSKeyedArchiver*

```
[NSKeyedArchiver archiveRootObject: objetAArchiver toFile: [self filePath]];
```

- Pour restaurer *NSKeyedUnarchiver*

```
objetARestorer = (NSObject *) [NSKeyedUnarchiver unarchiveObjectWithFile: [self filePath]];
```

Seule contrainte :

**objetAArchiver et objetARestorer
doivent être conforme à NSCoder**

```
@interface objetAArchiver : NSObject <NSCoding>
```

Déclaration de l'objet Person (fichier Person.h)

```
// Person.h

#import <Foundation/Foundation.h>

@interface Person : NSObject <NSCoding>
{
    NSString *name;
    NSDate *birthDate;
    float weight;
    BOOL married;
}

@property (nonatomic, retain) NSString *name;
@property (nonatomic, retain) NSDate *birthDate;
@property (nonatomic) float weight;
@property (nonatomic) BOOL married;

- (id) initWithName: (NSString *)aName
    birthDate: (NSDate *)aBirthDate
    weight: (float)aWeight
    isMarried: (BOOL)aStatus;

@end
```


Implémentation de l'objet Person (fichier Person.m)

```
// Person.m

#import "Person.h"

@implementation Person

@synthesize name, birthDate, weight, married;

- (void) encodeWithCoder: (NSCoder *)encoder
{
    [encoder encodeObject: name forKey: @"clename"];
    [encoder encodeObject: birthDate forKey: @"clebirthDate"];
    [encoder encodeFloat: weight forKey: @"cleweight"];
    [encoder encodeBool: married forKey: @"clemarried"];
}

- (id) initWithCoder: (NSCoder *)decoder
{
    self = [super init];

    [self setName: [decoder decodeObjectForKey: @"clename"]];
    [self setBirthDate: [decoder decodeObjectForKey: @"clebirthDate"]];
    [self setWeight: [decoder decodeFloatForKey: @"cleweight"]];
    [self setMarried: [decoder decodeBoolForKey: @"clemarried"]];

    return self;
}

// Constructeur
- (id) initWithName: (NSString *)aName birthDate: (NSDate *)aBirthDate weight: (float)aWeight isMarried: (BOOL)aStatus
{
    self = [super init];

    [self setName: aName];
    [self setBirthDate: aBirthDate];
    [self setWeight: aWeight];
    [self setMarried: aStatus];

    return self;
}

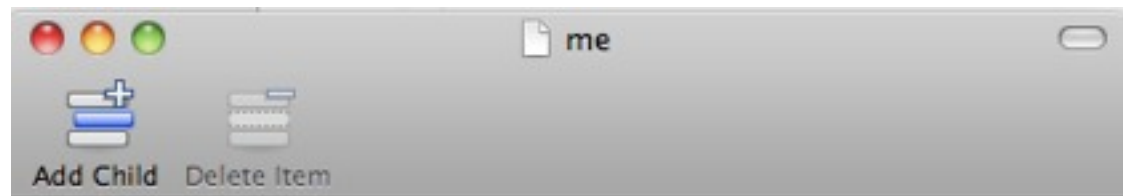
@end
```

Exercices

fichier *output.plist*

écriture

```
[array writeToFile:@"a_valid_path/output.plist" atomically:YES];
```



The screenshot shows a window titled "me" with a toolbar containing "Add Child" and "Delete Item" buttons. Below the toolbar is a table representing a plist structure.

Key	Type	Value
▼ Root	Dictionary (4 items)	
\$archiver	String	NSKeyedArchiver
▼ Subjects	Array (6 items)	
Item 0	String	\$null
▼ Item 1	Dictionary (2 items)	
clemarried	Boolean	<input type="checkbox"/>
cleweight	Number	54.67
Item 2	String	toto
▼ Item 3	Dictionary (1 item)	
NS.time	Number	-328013902
▼ Item 4	Dictionary (2 items)	
▼ \$classes	Array (2 items)	
Item 0	String	NSDate
Item 1	String	NSObject
\$classname	String	NSDate
▼ Item 5	Dictionary (2 items)	
▼ \$classes	Array (2 items)	
Item 0	String	Person
Item 1	String	NSObject
\$classname	String	Person
▼ Stop	Dictionary (0 items)	
\$version	Number	100000

Groups & Files

Persistence_Plist

Classes

Persistence_PlistAppD

Persistence_PlistAppD

Persistence_PlistViewC

Persistence_PlistViewC

Person.h

Person.m

Person.plist

MySettings.bundle

Root.plist

en.lproj

Root.strings

TestDataModel.xcdata

Other Sources

Resources

Frameworks

Products

Targets

Persistence_Plist

Copy Bundle Resource

Compile Sources (5)

Link Binary With Libra

Executables

Find Results

Bookmarks

SCM

Project Symbols

Implementation Files

NIB Files

File Name

TestDataModel.xcdatamodel

Threads

TestDataModel.xcdatamodel

Person

Entity	Abs	Class
Person	<input type="checkbox"/>	Person

Property	Kind	Type or Destination
birthDate	Attribute	Undefined
married	Attribute	Undefined
name	Attribute	Undefined
weight	Attribute	Float

Person

Attributes

birthDate

married

name

weight

Relationships