

# Introduction

Apple a pris l'habitude d'alterner les versions de son système d'exploitation en s'adressant successivement aux utilisateurs ou aux développeurs. Heureuse nouvelle : les nouveautés d'iOS 5 concernent essentiellement cette dernière catégorie. L'introduction du comptage de références automatique (ARC, *Automatic Reference Counting*) mérite à elle seule la mise à jour de la part des développeurs. Par ce simple ajout, Apple élimine bien des problèmes susceptibles de faire planter vos applications iOS, tout en simplifiant considérablement le développement du code et en améliorant ses performances. Vous avez donc intérêt à adopter ARC sans plus tarder. C'est l'une des plus importantes fonctionnalités d'Objective-C depuis l'apparition du pool d'autorelease.

Mais les nouveautés d'iOS 5 destinées aux développeurs ne s'arrêtent pas là. De l'apparition d'iCloud à la protection automatique des données, le système d'exploitation prend désormais en charge les problèmes les plus complexes, laissant ainsi les développeurs se concentrer sur les fonctionnalités les plus innovantes pour sublimer toutes leurs applications.

Parmi les nouveautés les plus visibles se tient la dernière déclinaison de Xcode. Que vous aimiez ou non les changements, vous avez intérêt à vous familiariser le plus vite possible avec sa nouvelle interface : il s'agit de l'environnement de développement par excellence des applications iOS. Cet ouvrage vous aide à en tirer le meilleur parti.

Vous êtes prêt à embrasser la dernière version du système d'Apple et à repousser les limites de vos applications ? Plongez sans plus tarder dans cet ouvrage.

## À qui s'adresse ce livre ?

Vous ne tenez pas entre les mains un livre qui vous introduit à l'environnement iOS. De nombreux ouvrages vous aiguillent parmi les bases d'Objective-C, en vous prenant par la main dans la découverte d'Interface Builder. Ce n'est pas notre intention. Ce livre part du principe que vous disposez d'ores et déjà d'une petite expérience d'iOS. Que vous soyez autodidacte ou que vous ayez suivi une formation spécialisée, vous avez probablement déjà développé au moins une application,

même si vous ne l'avez pas encore soumise à Apple. Si vous désirez aller plus loin et découvrir les meilleures techniques et les plus grands secrets de la communauté des développeurs, cet ouvrage est fait pour vous : il s'appuie sur une solide expérience, autour du développement de nombreuses applications concrètes.

Ce livre n'est pas non plus une collection de recettes. Vous trouverez de nombreux extraits de code source dans les pages qui suivent, mais nous nous concentrerons essentiellement sur la conception, le développement et la mise à jour d'applications iOS sophistiquées. Vous apprendrez notamment à manipuler les design patterns et à concevoir du code réutilisable dans tous vos projets, tout en découvrant la syntaxe spécifique d'Objective-C ainsi que l'ensemble des nouveaux frameworks.

Tous les exemples s'appuient sur Xcode 4. Si vous n'êtes pas encore familier avec cet environnement de développement, pas de panique : le Chapitre 2 vous aide à vous mettre à la page.

## **Ce que traite cet ouvrage**

La plate-forme iOS va toujours de l'avant et cet ouvrage épouse un tel mouvement. La plupart des exemples nécessitent donc iOS 5 et s'appuient sur le comptage de références automatique (ARC). Sauf dans quelques cas particuliers, nous n'évoquerons pas les solutions assurant la rétrocompatibilité de vos applications. Tout au long de cet ouvrage, nous nous efforcerons de développer les meilleures applications possibles, en tirant parti des dernières nouveautés de la plate-forme et du langage.

Cet ouvrage s'appuie sur l'iPhone 4 et l'iPad 2. La plupart des sujets évoqués s'appliquent également au premier iPad, ainsi qu'à l'iPod Touch, l'iPhone 3GS et Apple TV. À l'heure où nous mettons sous presse, l'iPhone 5 et l'iPad 3 n'ont pas encore dévoilé leurs contours mais tous les enseignements de cet ouvrage devraient s'y appliquer naturellement. Le Chapitre 12 est consacré au développement d'applications multi-plate-forme.

## **Structure de cet ouvrage**

iOS s'articule autour d'une série d'outils extrêmement riches, des frameworks de haut niveau comme UIKit aux outils de bas niveau comme Core Text. Bien souvent, il existe plusieurs solutions pour aboutir à un même objectif. En tant que développeur, comment pouvez-vous choisir la meilleure technique ?

Dans cet ouvrage, on distingue les tâches les plus courantes des opérations les plus spécifiques, en vous aidant à choisir la solution la plus appropriée face à chaque problème. Vous découvrirez le rôle de chaque framework ainsi que les relations qui les lient, tout en apprenant à sélectionner la bibliothèque la mieux adaptée à chaque cas de figure.

---

Cet ouvrage est découpé en quatre parties, des outils les plus courants jusqu'aux plus puissants.

## Partie I : Les nouveautés

Si vous êtes familier avec iOS 4, cette section vous introduit rapidement à toutes les nouvelles fonctionnalités d'iOS 5.

- **Chapitre 1 : Vue d'ensemble des nouveautés.** iOS introduit un grand nombre de fonctionnalités et vous découvrirez dans ce chapitre l'ensemble des nouveautés à votre disposition.
- **Chapitre 2 : Maîtriser Xcode 4.** Apple a repensé toute l'interface de Xcode et il n'est pas évident de revoir ses habitudes. Ce chapitre vous permet d'en tirer le meilleur parti.

## Partie II : Tirer le meilleur parti des outils courants

En tant que développeur iOS, vous disposez d'un large arsenal d'outils, des notifications aux vues de table en passant par les calques d'animation. Exploitez-vous pleinement l'ensemble de ces outils ? Au cours de cette partie, vous apprendrez les meilleures pratiques du développement Cocoa à partir d'exemples pratiques.

- **Chapitre 3 : Les meilleures pratiques en Objective-C.** Vous êtes prêt à progresser considérablement en Objective-C ? Ce chapitre présente tous les outils qu'utilisent les développeurs les plus expérimentés afin d'améliorer le design, la mise à jour et l'ergonomie de leurs applications.
- **Chapitre 4 : Les design patterns avec Cocoa.** Cocoa s'appuie sur un certain nombre de design patterns qui vous feront gagner un temps précieux et que vous pourrez librement réutiliser. Vous apprendrez au cours de ce chapitre comment en tirer le meilleur parti afin de résoudre de nombreux problèmes, à l'instar des propres équipes d'Apple.
- **Chapitre 5 : Gérer les vues de table.** Les vues de table constituent probablement l'un des éléments d'interface les plus complexes et les plus courants d'iOS. Simples et élégantes, elles revêtent bien souvent un aspect très complexe au cours du développement. Apprenez leur fonctionnement précis et découvrez comment les utiliser au mieux. Vous découvrirez ici de nombreuses techniques modernes et efficaces, comme le défilement infini.
- **Chapitre 6 : Mieux dessiner.** Le dessin personnalisé peut intimider de nombreux développeurs, mais il constitue un élément essentiel de la conception d'interfaces utilisateur élégantes et rapides. Vous découvrirez l'ensemble des options à votre disposition, d'UIKit à Core Graphics, en apprenant à les optimiser afin d'accélérer leur traitement sans niveler par le bas la qualité esthétique de vos applications.

- **Chapitre 7 : Des calques qui se superposent : Core Animation.** Les appareils iOS soutiennent des animations très élaborées. Grâce à leur puissant GPU et au framework Core Animation extrêmement optimisé, vous pouvez construire des interfaces attrayantes et intuitives. Dans ce chapitre, vous irez bien au-delà des modèles de base en perçant tous les secrets de l'animation.
- **Chapitre 8 : Gérer les erreurs.** Vous essayez de développer un code parfait, mais parfois vous faites face à des problèmes insurmontables. Anticiper la réaction de votre application face à des soucis inattendus l'érige au rang de programme extraordinaire. Vous découvrirez tout au long de ce chapitre les patterns les plus courants pour traiter les erreurs et apprendrez à améliorer votre code pour faire face aux erreurs les plus insoupçonnées.

### Partie III : Maîtriser les outils spécialisés

Si certains outils sont utilisés dans la quasi-totalité des applications, d'autres répondent à des besoins précis et on ne les exploite que dans des cas de figure spécifiques. Au cours de cette partie, vous découvrirez l'ensemble de ces techniques, qui devraient vous épauler dans les tâches les plus avancées.

- **Chapitre 9 : Contrôler le multitâches.** Le multitâches constitue un aspect important des applications les plus modernes. Vous apprendrez au cours de ce chapitre à effectuer plusieurs traitements à la fois, lorsque votre application s'exécute ou tourne à l'arrière-plan.
- **Chapitre 10 : Les échanges REST.** Les services REST constituent une solution moderne pour acheminer des informations aux applications les plus réactives. Vous découvrirez comment les implémenter avec iOS.
- **Chapitre 11 : Renforcer la protection grâce aux services de sécurité.** La sécurité et le respect de la vie privée des utilisateurs doivent être l'une de vos préoccupations majeures. Vous apprendrez ici à immuniser votre application et les données de l'utilisateur contre les pirates à l'aide de chaînes de clé, de certificats et de techniques de chiffrement.
- **Chapitre 12 : Exécuter des applications sur plusieurs plates-formes et appareils iOS.** Le panorama d'iOS s'élargit d'année en année, avec l'apparition de l'iPod Touch, de l'iPhone, de l'iPad, de l'Apple TV et de constantes nouveautés. Il n'est pas suffisant d'assurer une compatibilité avec l'ensemble de ces appareils ; vous devez faire en sorte que vos applications tirent le meilleur parti de chaque plate-forme. Vous apprendrez ici à adapter vos applications aux composants spécifiques à chaque lignée d'appareil, en tirant leur meilleur parti.
- **Chapitre 13 : Internationalisation et localisation.** Même si vous vous concentrez sur un seul marché aujourd'hui, anticipez dès à présent l'évolution de vos applications et envisagez de les adapter à un marché international. Vous vous épargnerez bien des soucis en prenant en compte cet aspect dès le début de vos projets.

- **Chapitre 14 : Rentabiliser son application grâce aux achats In App.** Les achats In App constituent une source de revenu complémentaire et un business model inédit pour tous les développeurs. Les utilisateurs plébiscitent ce type de services à la demande, qui améliorent l'ergonomie et les fonctionnalités d'une application. Vous apprendrez ici à implémenter ce type de services au sein de vos projets.

## Partie IV : Repousser les limites

Tout au long de cette section, vous découvrirez les techniques les plus avancées des développeurs iOS confirmés. Repoussez les limites de toutes vos applications en adoptant les meilleurs outils disponibles. Vous découvrirez au passage les mécanismes internes les plus complexes d'iOS.

- **Chapitre 15 : Le plus grand atout de Cocoa : le codage et l'observation clé-valeur.** La plupart des frameworks les plus puissants d'Apple s'appuient sur KVO pour offrir une telle performance et souplesse. Vous apprendrez au cours de ce chapitre à tirer le meilleur parti de KVO afin d'améliorer l'ensemble de vos techniques de développement.
- **Chapitre 16 : Penser différemment : les blocs et la programmation fonctionnelle.** De nombreux développeurs demeurent hermétiques à l'apparition de blocs au sein d'Objective-C. Il s'agit pourtant de l'un des plus précieux outils pour interagir avec les frameworks d'Apple, mais aussi pour renouveler entièrement votre manière de concevoir un programme. Embrassez dès à présent de nouvelles techniques de développement afin de mettre en place des projets ultramodernes.
- **Chapitre 17 : Travailler hors connexion.** La programmation réseau est un sujet complexe, mais il est encore plus difficile de délivrer à vos utilisateurs une expérience hors connexion sans heurt. Apprenez à mettre en cache vos données et à intégrer un tel système au sein des échanges avec votre serveur.
- **Chapitre 18 : Mettre en forme le texte.** – D'UIKit à Core Text, iOS met en place de nombreuses solutions pour afficher du texte. Comme il n'y a pas de technique idéale pour afficher du texte riche avec iOS, vous devez vous familiariser avec l'ensemble des frameworks à votre disposition pour choisir une solution à bon escient.
- **Chapitre 19 : Core Foundation.** Parmi les frameworks les plus puissants d'iOS, exploitez le plein potentiel des frameworks Core comme Core Graphics, Core Animation et Core Text. Ils s'appuient tous sur Core Foundation. Vous découvrirez au cours de ce chapitre le fonctionnement des types de données de Core Foundation afin de faire face à tous les cas de figure.
- **Chapitre 20 : Objective-C avancé.** Vous désirez connaître les rouages internes d'Objective-C et son fonctionnement le plus secret ? Ce chapitre vous ouvre les portes de son environnement d'exécution, en vous apprenant à modifier

dynamiquement les classes et les méthodes. Vous découvrirez également comment les appels de méthodes Objective-C se redirigent vers des appels de fonctions C, en prenant le contrôle de ce système afin d'étendre vos applications au-delà de toute limite.

Vous pouvez librement parcourir cet ouvrage dans l'ordre de votre choix, en vous concentrant sur les fonctionnalités qui vous intéressent le plus. Chaque chapitre est autonome, excepté ceux qui s'appuient sur des objets de Core Foundation (en particulier Core Graphics, Core Animation et Core Text). Ces sujets vous renvoient vers le Chapitre 19 pour de plus amples détails sur le fonctionnement de Core Foundation.

## Ce dont vous avez besoin pour utiliser ce livre

Tous les exemples de cet ouvrage ont été développés avec Xcode 4.2 sur Mac OS X 10.7 et iOS 5. Vous avez besoin d'un compte de développeur Apple pour accéder à la plupart des outils et de la documentation, ainsi que d'une licence de développeur pour exécuter les applications sur vos propres appareils iOS. Reportez-vous à l'adresse <http://developer.apple.com/programs/> pour vous inscrire.

La plupart des exemples de cet ouvrage peuvent s'exécuter sur le Simulateur iOS qui accompagne Xcode 4.2. Vous pouvez pleinement l'utiliser sans licence de développeur Apple.

On recense quelques disparités entre Xcode 4.2 sur Mac OS X 10.6 et 10.7 ; tous les exemples ont été testés sous 10.6.

## Parcourir la documentation Apple

Apple propose une colossale documentation sur son site web et au sein de Xcode. Les URL changent fréquemment et sont souvent très longues. Cet ouvrage se réfère essentiellement aux documentations officielles par leur titre et non leur adresse. Pour rechercher des documents dans Xcode, appuyez sur Cmd-Option-? ou cliquez sur Help > Documentation and API Reference. Dans l'organisateur de documentation, cliquez sur l'icône de recherche, saisissez le nom du document et sélectionnez le résultat correspondant.

Pour rechercher des documents sur le site des développeurs Apple, rendez-vous à l'adresse [developer.apple.com](http://developer.apple.com), cliquez sur Member Center et authentifiez-vous. Sélectionnez ensuite la section iOS Dev Center et saisissez le titre d'un document dans le champ de recherche.

La documentation en ligne est généralement la même que celle qui accompagne Xcode. Vous trouverez bien souvent deux versions d'un même document, pour iOS et pour Mac. Choisissez la première version, sous peine de voir des méthodes qui vous seront indisponibles.

## Code source

En parcourant les exemples de cet ouvrage, vous avez la possibilité de les saisir manuellement dans Xcode ou de télécharger tous les fichiers correspondants sur le site web officiel de Pearson (<http://www.pearson.fr>), en suivant le lien **Codes sources** sur la page dédiée à ce livre dans la librairie en ligne. Par exemple, le code suivant est disponible dans le dossier Chapitre 18 de cette archive à télécharger, dans le projet *SimpleLayout* et le fichier *CoreTextLabel.m* :

```
CoreTextLabel.m (SimpleLayout)
- (id)initWithFrame:(CGRect)frame {
    if ((self = [super initWithFrame:frame])) {
        CGAffineTransform
            transform = CGAffineTransformMakeScale(1, -1);
        CGAffineTransformTranslate(transform,
            0, -self.bounds.size.height);
        self.transform = transform;
        self.backgroundColor = [UIColor whiteColor];
    }
    return self;
}
```

Parfois, nous ne publions que des extraits de code source dans cet ouvrage ; le cas échéant, reportez-vous aux fichiers à télécharger afin de parcourir l'ensemble du code correspondant.

## Votre avis

L'avis de nos lecteurs étant toujours le bienvenu, n'hésitez pas à nous faire part de vos commentaires. Pour cela, rendez-vous sur la page dédiée à cet ouvrage sur le site web Pearson (<http://www.pearson.fr>) dans la librairie en ligne et cliquez sur le lien Réagir.

## Conventions typographiques

Les conventions typographiques sont les suivantes.

### Gras et gris

Représente les URL.

### Italique

Indique des termes nouveaux, des noms de fichiers, des répertoires et du texte mis en exergue.

### Chasse fixe

Cette police est utilisée pour les parties de code, le contenu des fichiers, les commandes et le résultat des commandes.

**Chasse fixe gras**

Signifie à l'utilisateur de saisir littéralement des commandes ou du texte, et met en avant, à l'occasion, des parties du code.

*Chasse fixe italique*

Montre que du texte doit être remplacé par des valeurs saisies par l'utilisateur et souligne certains commentaires dans le code.