

# Cocoaはやっぱり! 出張版

2001.7

## 1. Cocoaってどう?

### ■ あいさつ

どうもはじめまして。今回より月に一度Cocoaに関する連載を書くことになりました鶴藺賢吾と申します。Cocoaに関しては、「興味はあるんだけど情報が少なくどう手を付けてよいのやら...」と置いていらっしゃる方も多いのではないのでしょうか。この連載では、多少なりでもそういった方の手助けになればと思っています。

かくいう私自身も勉強中の身でして、この連載を通じてさらに勉強してきたいと思っていますし、色々と読者のみなさんよりの情報やフィードバック等も期待しているところです。よろしく願いいたします。では、早速始めたいと思います。

### ■ Cocoaとは何か

Mac OS Xの構成図を見ると分かりますが、Mac OS Xで動作するアプリケーションには、「Classic、Carbon、Cocoa」の3種類の形式があります（Javaを含めている図もありますので、この場合は4種類となります）。

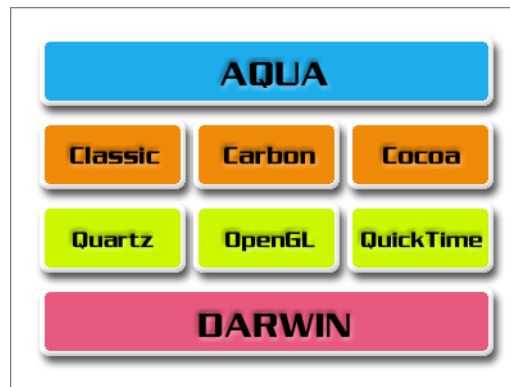


図1. Mac OS Xの構成図

Classicは、従来のMac OS 9までで使用されてきた形式です。Mac OS Xに備わっているClassic環境によって動作します。ただし、Mac OS Xに備わっている新しい機能（プリエンティブマルチタスク、メモリ保護、Quartzによる高機能なグラフィック等々）をフルに利用することはできません。

Carbonは、Mac OS Xの新しい機能を利用するためにClassicから発展して作られた、Mac OS Xネイティブの新しい形式です。CarbonからはMac OS Xに備わっている新しい機能を利用することが出来ます。また、開発者の視点から見ると、CarbonのAPIはClassicのAPIと似ているため、既存のClassicアプリをゼロから作りなおすことなく、Mac OS X対応させることが可能です。

Cocoaは、Carbonと同様にMac OS Xの新しい機能が使える新しいアプリケーションの形式ですが、ベースになっているのはNeXT Computerで生まれた「OPENSTEP（旧NeXTSTEP）」です。そのため開発

方法が今までとは全く異なります。Carbonがあるにもかかわらず、Cocoaというものが用意された理由は、なんといっても「**開発が容易である**」ことです。Mac OSは（つまりClassic APIは）、長年の建て増しの繰り返しのよって継ぎはぎの目立つものになってきました。CarbonもClassicをベースにしているため、整理はされているものの同様の性質を持っています。

Cocoaは、非常にエレガントな作りになっていて、「**10倍簡単に開発ができる**」というような表現を見掛けることがあります。OPENSTEP自体は随分昔に開発されたものですが、基本設計のよさもあり、それを引き継いだCocoaは今でも十分新しく、Mac OSの開発環境としては最も優れたものといえます。これから新規にMac OS X対応のアプリケーションを作る、もしくは、これからMac OS Xのプログラミングの勉強を始めるといふ方には、CarbonよりもCocoaの方が非常に魅力的な解になるでしょう。

## ■ 開発ツール

Cocoaの開発を容易にしている1つ目の理由は、開発用ツールにあります。Project BuilderとInterface Builderが2大開発ツールで、以下のようなものです。

- **Project Builder** : プロジェクト管理、ビルド、デバッグ
- **Interface Builder** : ユーザーインターフェイス等の定義、テスト

特にInterface Builderが特徴的で、2つのオブジェクトをグラフィカルに接続しながらアプリケーションを組み立てていくことができます。**2つのオブジェクトを接続することで、そのオブジェクト間でメッセージを送ることができるようになります。**また、その接続状況を確認する画面もあります。

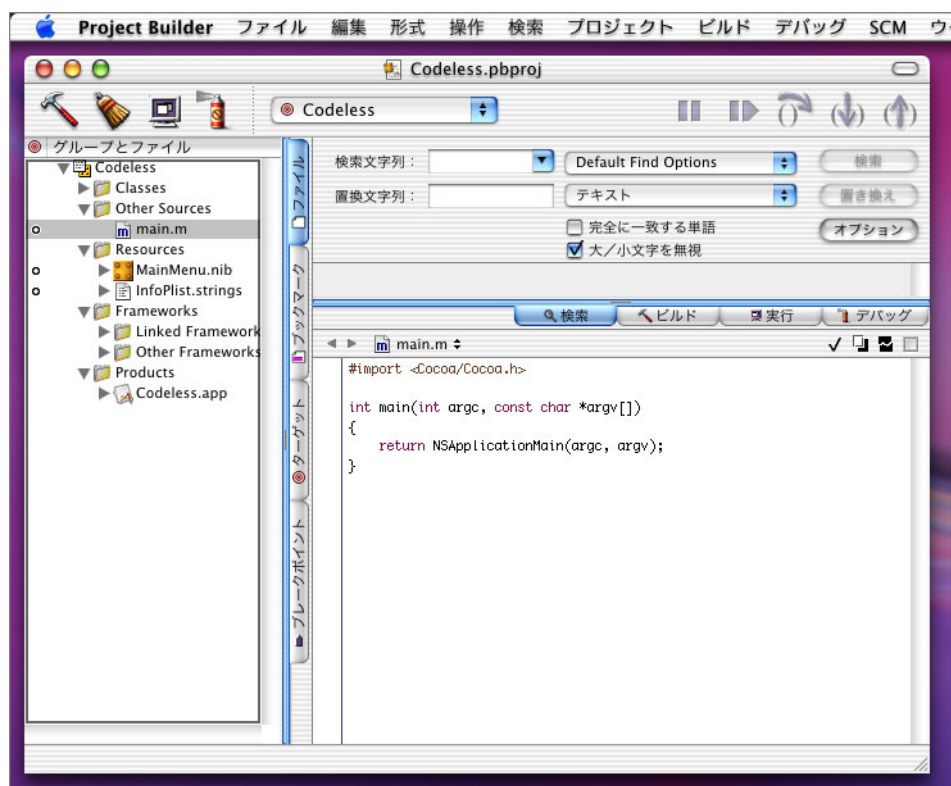


図2. Project Builder

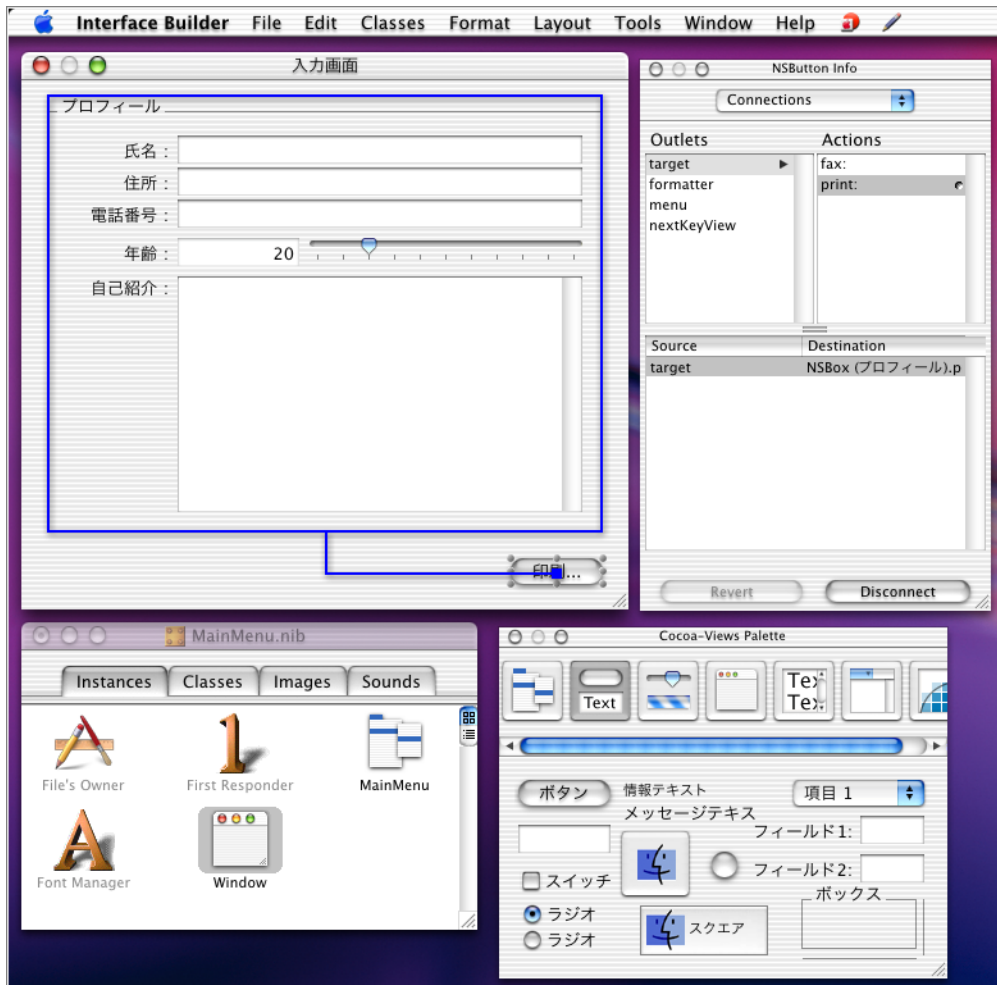


図3. Interface Builderでオブジェクトの接続を行っているところ。  
(印刷ボタンからプロフィールに接続しています)

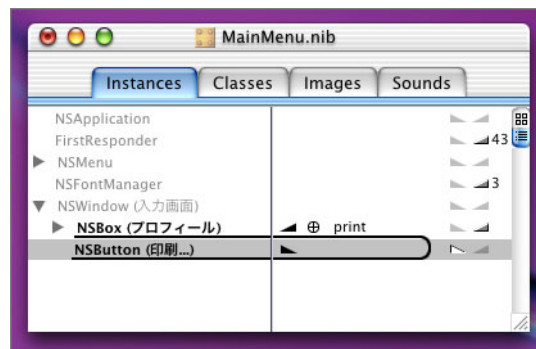


図4. Interface Builderの接続状況確認画面  
(印刷ボタンからプロフィールに「print」メッセージが送信されることが分かります)

## ■ フレームワーク

Cocoaの開発を容易にしているもう1つ目の理由は、フレームワークにあります。FoundationとApplication Kitが2大フレームワークで、以下のようなものです。

- Foundation : 文字列、配列、辞書、ファイル、通信、分散オブジェクト等
- Application Kit : ユーザーインターフェイス、グラフィック、サウンド等

Foundationの方は、ソフトウェアを開発する上で必要となる基礎的なクラスを提供してくれます。Application Kitの方は、ユーザーと対話するためのユーザーインターフェイスやグラフィカルな要素を扱うクラスを多く含みます。どちらも全面的にオブジェクト指向を取り入れたエレガントなものとなっています。

MACWORLD Expoでよくデモされていたような、**画像を回転したり透明度を変えたり**といったことも簡単です。**印刷やPDFへの出力、ムービー再生、スクリーンセーバーの作成**も簡単にできます。また、UNIXのコマンドも呼び出せますので、**UNIXのプログラムにMacらしいユーザーインターフェイスを付加する**ような用途にも使うことも可能です。

## ■ コーディング無しで作れるアプリケーション

さて、Cocoaの開発環境の優秀さの一端を見ていただくために、コーディングをしないでここまでのアプリケーションが作れるという例をお見せしましょう。プロフィール入力を行うアプリケーションです。

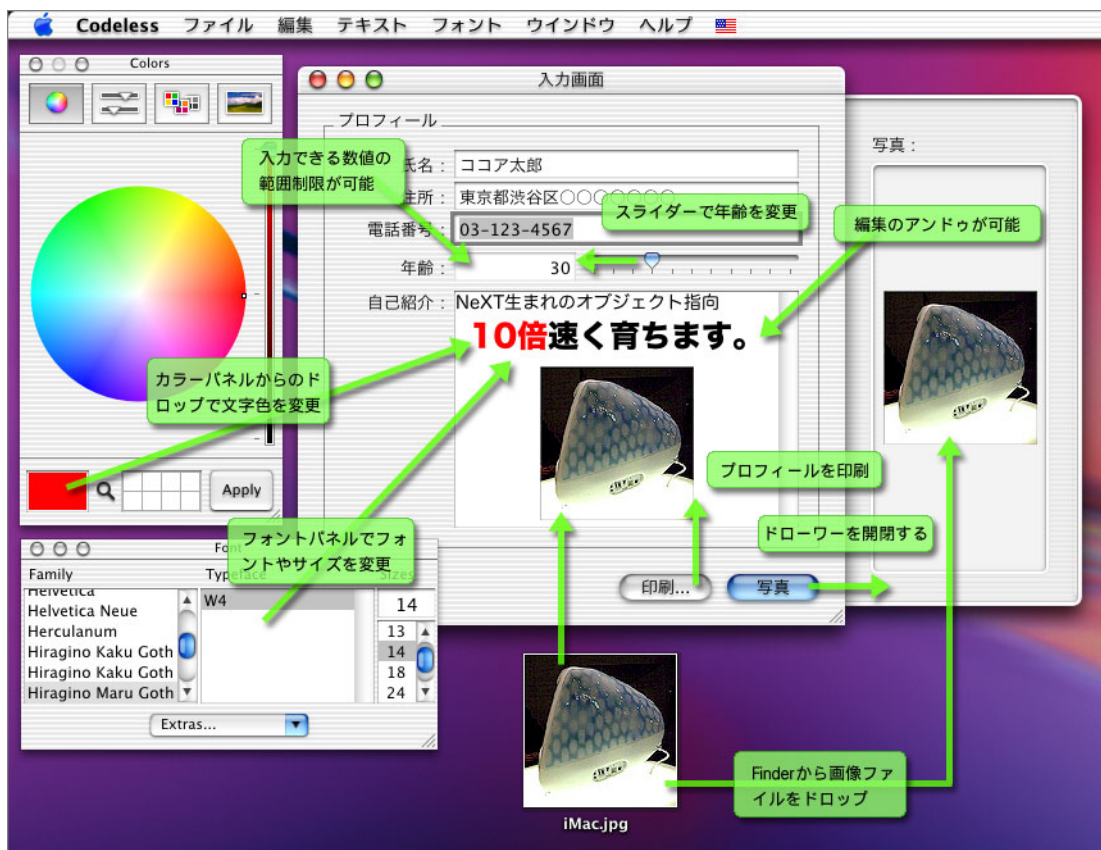


図4. コーディング無しで作ったサンプルアプリケーション

図に書いているように、以下のようなことがコーディング無しで実現できます。

- ・ フォントパネルを使ったフォントや文字サイズの変更。
- ・ カラーパネルを使った文字色の変更。
- ・ テキスト編集のアンドゥが可能。
- ・ 数値入力範囲の制限が可能。
- ・ Finderから画像のドロップによる画像の挿入。
- ・ スライダーによるテキスト入力。

- ・ ボタンによるドローワーの開閉。
- ・ プロフィールの印刷。

印刷は簡易的なものになりますが、印刷プレビューを表示させると以下ようになります。

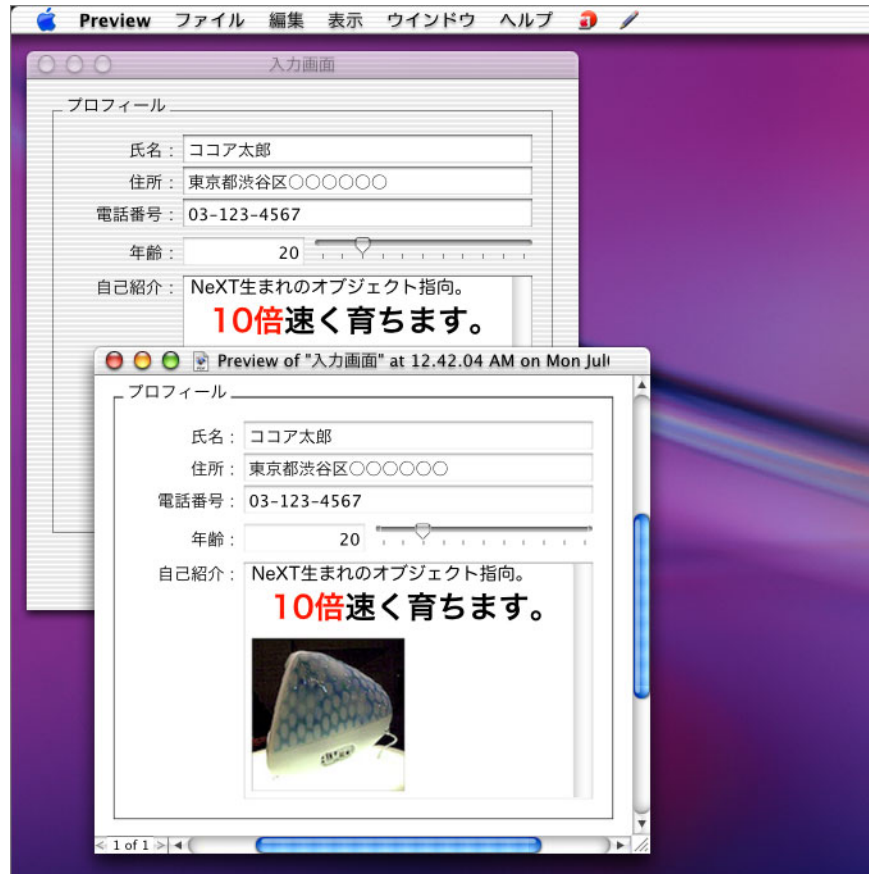


図5. 印刷でプレビューを表示させたところ (手前のウィンドウが印刷プレビュー)

これはちょっと極端な例かもしれませんが、Cocoaでは、これくらいのことはコーディングをしなくてもできてしまうのです。

## ■ 開発言語

Cocoaの開発で使用する言語は、「Objective-C」もしくは「Java」です。Cocoaの2大フレームワークはObjective-Cで記述されていることもあり、この連載では、Objective-Cを使います。

Objective-Cは、馴染みのない方が多いかと思いますが、Cocoaという新しいものに加えて、新しい言語も覚えなければならないのかと思われるかもしれません。でも、**Objective-Cは、ANSI C互換のC言語**で、そこに**シンプルな形でオブジェクト指向の機能拡張**を行っています。ですので、C言語が分かる方であればさほど修得には時間はかかりません。

C言語ベースにオブジェクト指向の拡張を行った言語としてはC++の方が一般的ですが、かなり多機能で修得には時間がかかります。しかし、C++の複雑さに挫折した方でも、Objective-Cならなんとかなるかもしれません。それくらいシンプルな言語です。でも、かなりダイナミックで強力な機能を備えています。プログラム言語の善し悪しを論じるのは難しいですが、かなり面白い言語であることは間違いないと思います。

Javaに通じている方は、JavaでCocoaの開発を行うことも可能です。Javaを使う場合は、Java Bridgeを経由してObjective-Cで書かれたフレームワークを使うことになり、パフォーマンス的には多少不利なところもあります。

## ■ 開発環境の入手方法

もしかすると、Cocoaで開発をする最大のメリットは、開発環境が安価に手に入ることもかもしれません。14,800円のMac OS Xのパッケージには、開発ツールが標準でついています。Macでは、こんなに安価な開発環境は他にはありません。

また、Appleの開発者向けのサービスである「Apple Developer Connection (ADC)」に入会することで、Appleのサイトより無料でダウンロードすることができます (ファイルは巨大ですので常時接続環境がないと辛いですが)。ADCのにはいくつかのコースがありますが、「Onlineプログラム」は無料ですので入会して損は無いでしょう。

### Apple Developer Connection プログラム

<http://developer.apple.com/ja/membership/index.html>

## ■ Cocoaの欠点

いいこと尽くめのようなCocoaにも欠点があります。ここでは、その欠点についても触れておきましょう。

### ◎ CocoaアプリはMac OS X専用

Cocoaで開発する最大のデメリットは「CocoaアプリがMac OS X専用であること」でしょうか。Carbonアプリは、Mac OS 8.6以降で動くように開発が出来ますが、Cocoaではそれはできません。古いマシンや古いOSでも動くようにという要求がある場合は、Cocoaでの開発はマッチしないといえます。

### ◎ 大規模なClassicアプリからの移行は大変

Classic/CarbonとCocoaの間にはAPIの互換性がないので、既存のClassicアプリをCocoaへ移植する場合、Carbon化するよりも一般に工数がかかります。小規模なものであれば、開発の容易さもありますので、ゼロから作りなおすという手もあるかもしれませんが、大規模なものには向かないといえます。

### ◎ Cocoaは発展途上

Mac OS X自体が発展途上であるともいえるのですが、CocoaはOPENSTEPから発展してきたために、従来のMac OSの作法を完全には身に付けていないところがいくつか見受けられます。例えば、

- ・ファイルタイプやクリエイータのサポートが弱い。
- ・エイリアスは扱えないのでファイルパスを使う。
- ・リソースフォークのサポートが弱い。

というような点です。CocoaアプリからはCarbon APIを呼び出すことができるため、以上の問題は解決できない問題ではありませんが、CocoaとCarbonの両方を知らなければならないということには変わりありません。これは、今後のCocoaのフレームワークの改善に期待するしかありませんが、現状としては、Mac OSらしいアプリを作るには多少面倒な手続きが必要になることがあります。

## ■ 情報源

最後に、現在手に入るCocoaの開発情報源を紹介します。情報は豊富にあるとはまだまだ言い難いですが、徐々に増えつつあります。また、Cocoaで作成されたフリーウェアは、ソースコードが公開されているものが比較的多く参考になります。日本語による情報はもう少し増えてほしいところですが..

### ◎ WEBサイト

#### • Cocoa (英語)

<http://developer.apple.com/cocoa/>

AppleのオフィシャルなCocoaのサイトです。Cocoaに関するドキュメントやCocoaのメーリングリスト(英語)があります。

#### • Cocoa Sample Code (英語)

[http://developer.apple.com/samplecode/Sample\\_Code/Cocoa.htm](http://developer.apple.com/samplecode/Sample_Code/Cocoa.htm)

Appleが配付しているCocoaのサンプルコードが置いてあるサイトです。

#### • Discovering OPENSTEP (PDF)

<http://developer.apple.com/ja/techpubs/pdf/DiscoveringOPENSTEP-J.pdf>

アップルが日本語で配付している資料は少ないのですが、最も本格的なものといえば、この「Discovering OPENSTEP」です。名前から分かるようにOPENSTEPの解説なのですが、大半は今でも通用しますのでダウンロードしておいて損はありません。オブジェクト指向についての解説もあります。

#### • Nito's Page

<http://www.fsinet.or.jp/~nito/index-j.html>

Cocoa倶楽部というメモリ管理や例外処理についてかかれた記事や、OPENSTEPの詳細な解説があります。

#### • HAPPY Macintosh Developing TIME!

<http://www05.u-page.so-net.ne.jp/xd5/mkino/HMDT/>

Cocoaについての調査結果をアクティブに発表しているサイトです。サンプルも多く参考になります。

#### • Java de Cocoa

<http://underwater.lab.nig.ac.jp/~takaho/computer/osx/index.html>

JavaでのCocoaの開発情報は比較的少ないのですが、日本語のサイトではこちらが一番充実しています。

#### • Cocoa Dev Central (英語)

<http://www.cocoadevcentral.com/>

テーマ毎に短かめのサンプルを用いて解説を行っていて分かりやすいです。

#### • Vermont Recipes (英語)

<http://www.stepwise.com/Articles/VermontRecipes/>

丁寧なチュートリアルや充実したリンク集があってお薦めのサイトです。

- **Native OS X Application : Cocoa Source Code (英語)**

<http://osx.hyperjeff.net/Apps/CocoaSource.html>

Cocoaの公開されているソースコードへのリンク集です。大量のリンクがあり、生きた教材であるソースコードを沢山見ることが出来ます。

- **Omni Group (英語)**

<http://www.omnigroup.com/>

NeXT時代からのデベロッパー。OmniWeb等でMac OS Xのアプリも多数リリースしています。メーリングリストや高機能なフレームワークも配付しています。

- **Cocoaはやっぱり!**

<http://www.big.or.jp/~crane/cocoa/>

手前味噌になってしまうかもしれませんが、私が書いているサイトです。順に読んでいくことでCocoaを理解できるように書いています。サンプルコードも沢山あります。

## ◎ 書籍

- **Mac OS Xプログラミング入門 Objective-C**

萩原剛志著 広文社 ISBN4-87778-068-8

残念ながら日本語のCocoa解説書は現在この1冊だけです。タイトルどおりObjective-Cに絞った解説ですので、やや地味な内容ですが、Cocoaに関する知識の地固めには欠かせない情報が豊富にあります。アプリケーションを作るために不可欠なApplication Kitについての解説は少なく、そういった情報に関しては、現在はWeb上から得ることになります。

- **Learning Cocoa (英語)**

Apple Computer, Inc. O'REILLY ISBN0-596-00160-6

現在、唯一のCocoaアプリケーション開発の解説本です。ただし英語です。また、内容的にはAppleのサイトからダウンロードできるPDFの内容が中心です。ですので紙で読みたい方には悪くありませんが、画面で読めればいいという方は、PDFをダウンロードしましょう。

## ■ 終わりに

いかがだったでしょうか。Cocoaの開発環境は、高機能で効率的に開発が出来ることがちょっとは分かっていただけではないかと思います。もし、Mac OS Xのパッケージをお持ちであれば、今すぐにでも始められます。週末にでもチャレンジしてみたいかでしょうか。

今後毎月一度のペースで、サンプルを交えたすぐに役に立つCocoaのワンポイントレッスンのようなものを書いていきたいと思っています。