

# Cocoa+Java～Mac OS X 10.2対応への覚書

2002年11月11日 赤松正行

## はじめに

### ◆本稿について

本稿は、「Mac OS Xプログラミング入門」シリーズの書籍「Cocoa+Java～Mac OS X 10.1対応版」（広文社刊、ISBN4-87778-093-9）を元に、Mac OS X 10.2において変更や追加があった事項を説明している。各事項に対応する同書のページ番号を括弧内に記している。

### ◆Mac OS X 10.2について

(p.4) コードネーム「Jaguar」としても知られるMac OS X 10.2は、2002年8月24日（米国時間）に発売され、数多くの新機能の搭載や堅実な改良が行われている。Mac OS X 10.2には、従来に引き続き開発用ツールが付属しており、OSを購入するだけで開発を行うことができる。本稿で使用したおもな開発ツールのバージョンは以下の通りである。

Project Builder バージョン2.0.1 (July 2002 Developer Tools)

Interface Builder バージョン2.3 (v283)

## 第2章 アプリケーション開発の実際

### ◆2.2 Developer Toolsのインストール

(p.19) 「Developer Tools」 CD-ROMの内容は10.1とほぼ同じだが、一部のファイル名が異なっている。

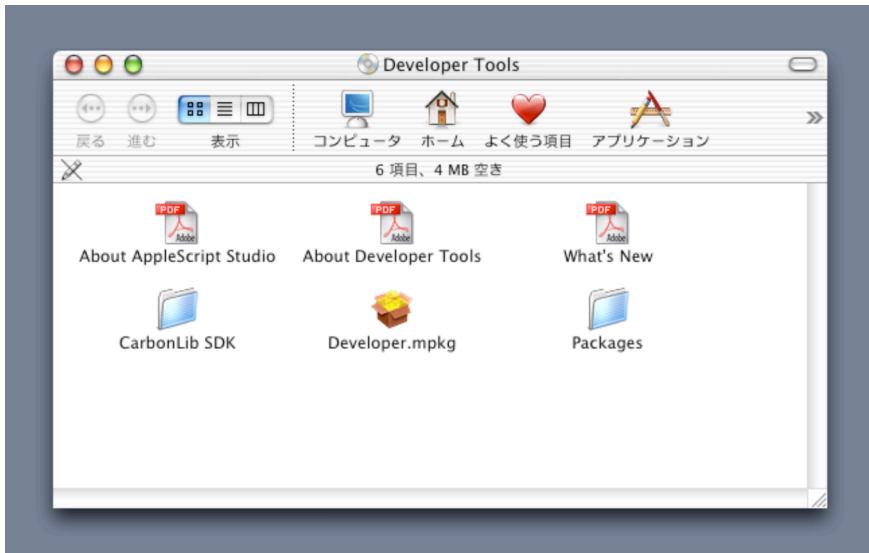


図2-001 「Developer Tools」 CD-ROMの内容

(p.20) インストーラのファイル名は「Developer.mpkg」となり、拡張子が変わった。

(p.21) インストール時には管理者パスワードの入力が求められなくなった。

### ◆2.3 Project Builderの設定

(p.28) Project Builderの設定アシスタントで、エディタウィンドウの設定が変わって、ウィンドウ環境の設定を行うようになった。ウィンドウの取り扱いについて、3種類の方法からいずれかを選ぶ。



図2-002 ウィンドウ環境の設定

(p.28) Project Builderの設定アシスタントで、ファイル管理の設定が加わった。プロジェクトやプロジェクトウィンドウを閉じる時の動作を設定する。



図2-003 ファイル管理の設定

#### ◆2.4 新規プロジェクトの作成

(p.29) 新規プロジェクトのアシスタントで選択できるプロジェクトが増えた。しかし、ここでは同じく「Cocoa-Java Application」を選べば良い。



図2-004 作成するプロジェクトの種類を選択する。

(p.31) プロジェクトウィンドウのツールバーに表示されるアイコンの並びが変わった。また、「Project Builder Release Notes」は表示されない。

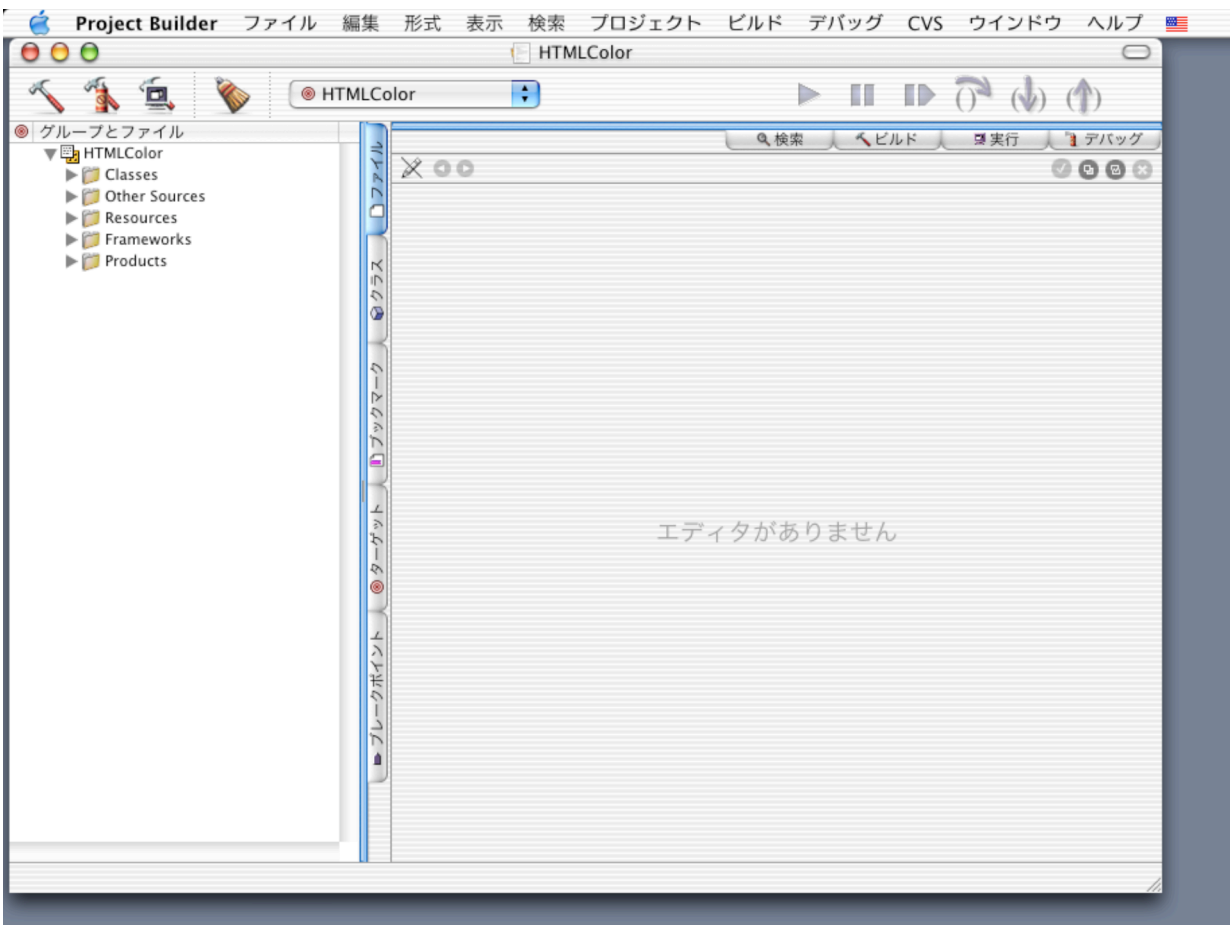


図2-005 プロジェクトウィンドウの初期状態

## ◆2.5 アプリケーションのビルドと実行

(p.33) デフォルトでは、ビルド中の進行状況は「ビルド」タブに表示されないようになった。なお、「ビルド」パネルの下部にあるパネルを区切るバーを上にはドラッグすれば、進行状況を見ることができる。また、プロジェクトウィンドウ下部のステイタスバーにビルドの進行状況が表示され、ビルドが正常に行われると「ビルドは問題なく完了しました」と表示される。

(p.34) ソースコードに間違いがあった場合に、Javaのコンパイルが完了しないというバグは、10.2の Developer Toolsでは解消されている。従って、「注意！」事項は無視して良い。ここで記載した対処方法を設定する必要もない。

## ◆2.6 ビューの配置

(p.35) 「グループとファイル」リストに表示されるファイルの左側にチェックボックスが表示されるようになった。このチェックボックスのチェックを外せば、そのファイルはビルドの対象から外される。例えば、ソースコードを修正した場合でも、チェックを外せば、ソースコードはコンパイルされない。

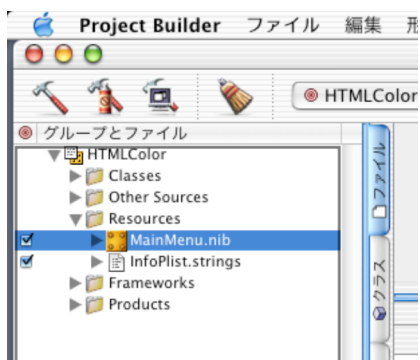


図2-006 「グループとファイル」リストの表示

(p.36) 「Cocoa-Views」パレットの内容が一部変化し、以下の図のようになった。テキストフィールドは右上にある白地の長方形なので、これをドラッグしてウィンドウにドロップする。

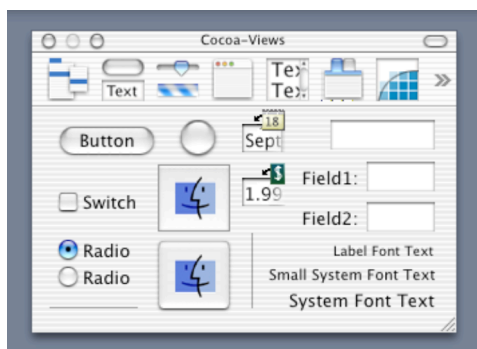


図2-007 「Cocoa-Views」パレット

(p.37) カラーウェルの初期状態が白っぽい青に変更されたので、これに合わせてテキストフィールドに最初に表示するテキストとして「#0e0eff」を設定する。

(p.38) 「Cocoa-Views」パレットのメッセージテキストは、フォントサイズ違いで3種類表示されるようになった。ここでは左下にある「System Font Text」をドラッグし、ウィンドウにドロップする。

## ◆2.7 ウィンドウの設定

(p.40) QuickTime Playerなどで見られるメタル調のウィンドウに設定することもできる。ウィンドウをメタル調にするには、InfoウィンドウのAttributesの「Options」にある「Texture Window」をチェックする。

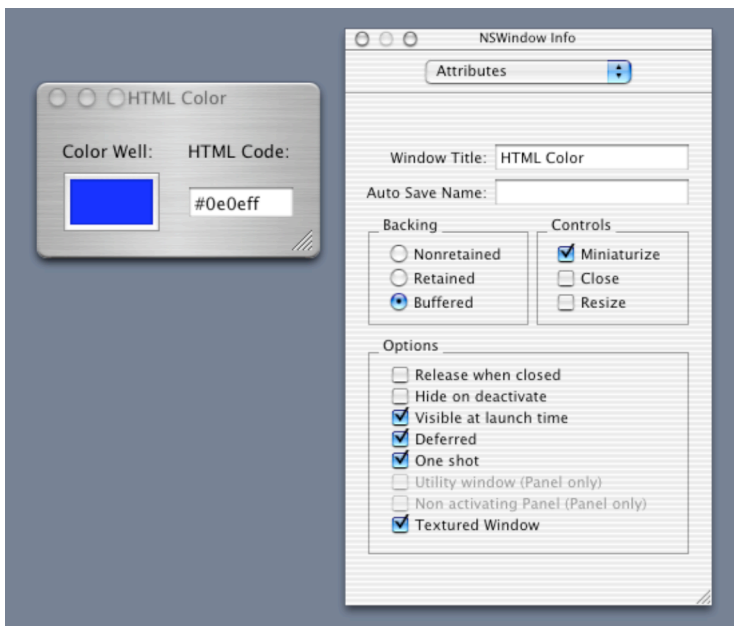


図2-008 メタル調のウィンドウ

## ◆2.8 インターフェースのテスト

(p.41) Colorsパレットのデザインが変わり、クレヨン表示など一部機能が増えている。使用方法としては以前とほぼ同じである。

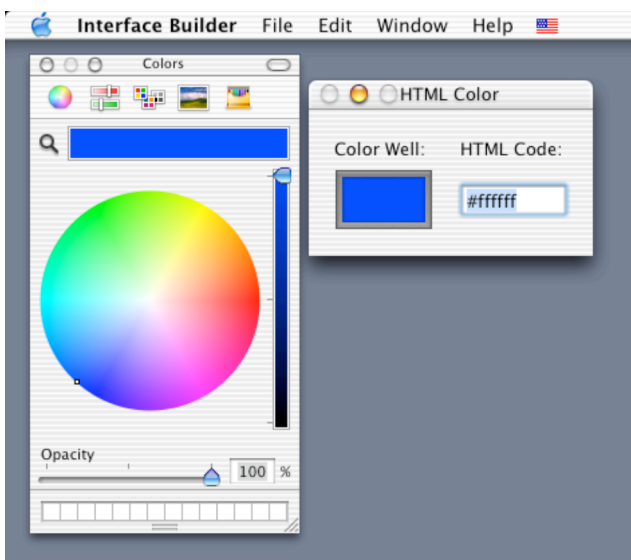


図2-009 Colorsパレット

## ◆2.10 アウトレットの作成

(p.44) InfoウィンドウにおけるクラスのAttributes表示が変更されている。アウトレットとアクションはタブによって表示を切り替える。また、アウトレットやアクションの追加や削除を行う「+」ボタンと「-

」ボタンは、「add」ボタンと「remove」ボタンに変更された。重要な変更点として、アウトレットのタイプをポップアップ・リストから指定できるようになった。ここではcodeTextFieldアウトレットのタイプとして「NSTextField」を指定する。

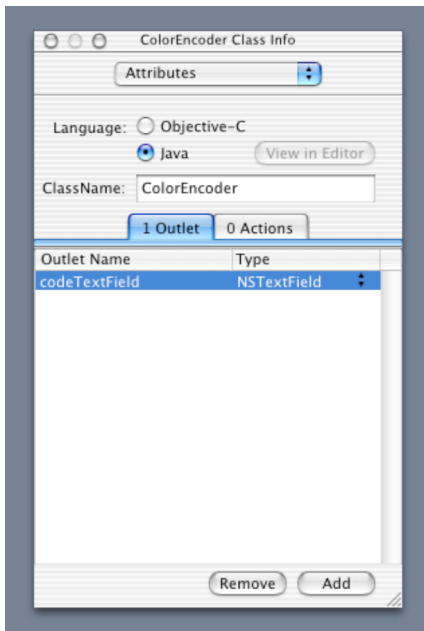


図2-010 InfoウィンドウによるクラスのAttributes表示

#### ◆2.16 メソッドのコーディング

(p.54) Interface Builderでアウトレットのタイプ指定ができるようになったことに伴い、以下のように生成されるソースコードでは、アウトレットに対応するインスタンスフィールドのクラスが自動的に記述されるようになった。従って、インスタンスフィールドをObjectクラスからNSTextFieldクラスに書き換える必要はない。また、インスタンスフィールドは自動的にpublic指定される。本文で記述している「クラスフィールド」は「インスタンスフィールド」に訂正されたい。

```
/* ColorEncoder */  
  
import com.apple.cocoa.foundation.*;  
import com.apple.cocoa.application.*;  
  
public class ColorEncoder {  
  
    public NSTextField codeTextField; /* IBOutlet */  
  
    public void encode(Object sender) { /* IBAction */  
    }  
  
}
```

(p.55) encodeアクションに対応するメソッドのアーギュメントは、本文通りObjectクラスからNSColorWellクラスに書き換えなければならない。従って、クラスを訂正したソースコードは以下のようになる。

```
/* ColorEncoder */
```

```

import com.apple.cocoa.foundation.*;
import com.apple.cocoa.application.*;

public class ColorEncoder {

    public NSTextField codeTextField; /* IBOutlet */

    public void encode(NSColorWell sender) { /* IBAction */
    }

}

```

## ◆2.18 デバッガの利用

(p.62) デフォルトでは、使用するデバッガが「Gdb」に設定されているので、ソースコードエディタでブレークポイントを指定しても、実行時にブレークポイントで一時停止しない。このような場合には、プロジェクトメニューから「アクティブな実行可能ファイルを編集」を選び、デバッガ欄で「Javaデバッガ」を使用するように指定すれば良い。

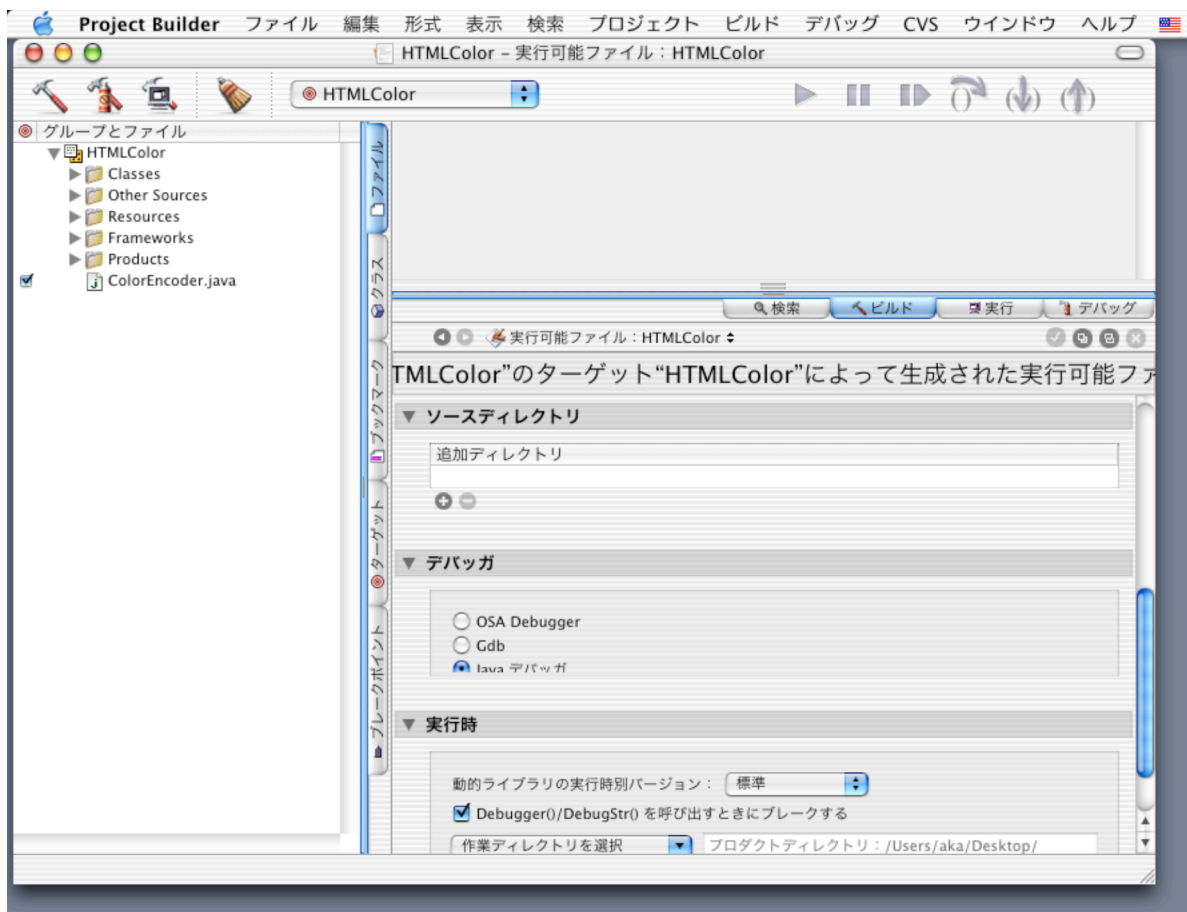


図2-011 使用するデバッガの指定

(p.66) 使用するデバッガは、「ターゲット」タブを開き、「Executables」の「HTMLColor」を選択して表示される実行可能ファイルの編集項目のうち、デバッガ欄で行う。上記のように、プロジェクトメニューから「アクティブな実行可能ファイルを編集」を選んでも良い。ここでは「Gdb」と「Javaデバッガ」に加えて、「OSA Debugger」も指定できるようになった。



## ◆2.20 メニューの処理

(p.68) ColorPanelControllerクラスに追加したsourceColorWellのアウトレットのタイプは、「NSColorWell」を指定する。

(p.73) 生成されたソースコードでは、ColorPanelControllerクラスのインスタンスフィールドsourceColorWellは自動的にNSColorWellクラスになっているので、修正する必要はない。なお、open()メソッドのパラメータであるsenderは、本文通りNSMenuItemクラスに修正する。

## ◆2.24 ヘルプの修正

(p.82) HTML形式のヘルプファイルを指定するには、「ターゲット」タブを開き、「Targets」リストの「HTMLColor」を選択する。そして「Info.plistのエントリ」の「詳細設定ビュー」を選ぶ。そして、「同位置に新規作成」ボタンをクリックすれば、新しいプロパティが追加されるので、その名前を「NSHelpFile」、クラスを「文字列」、値を「help.html」とする。

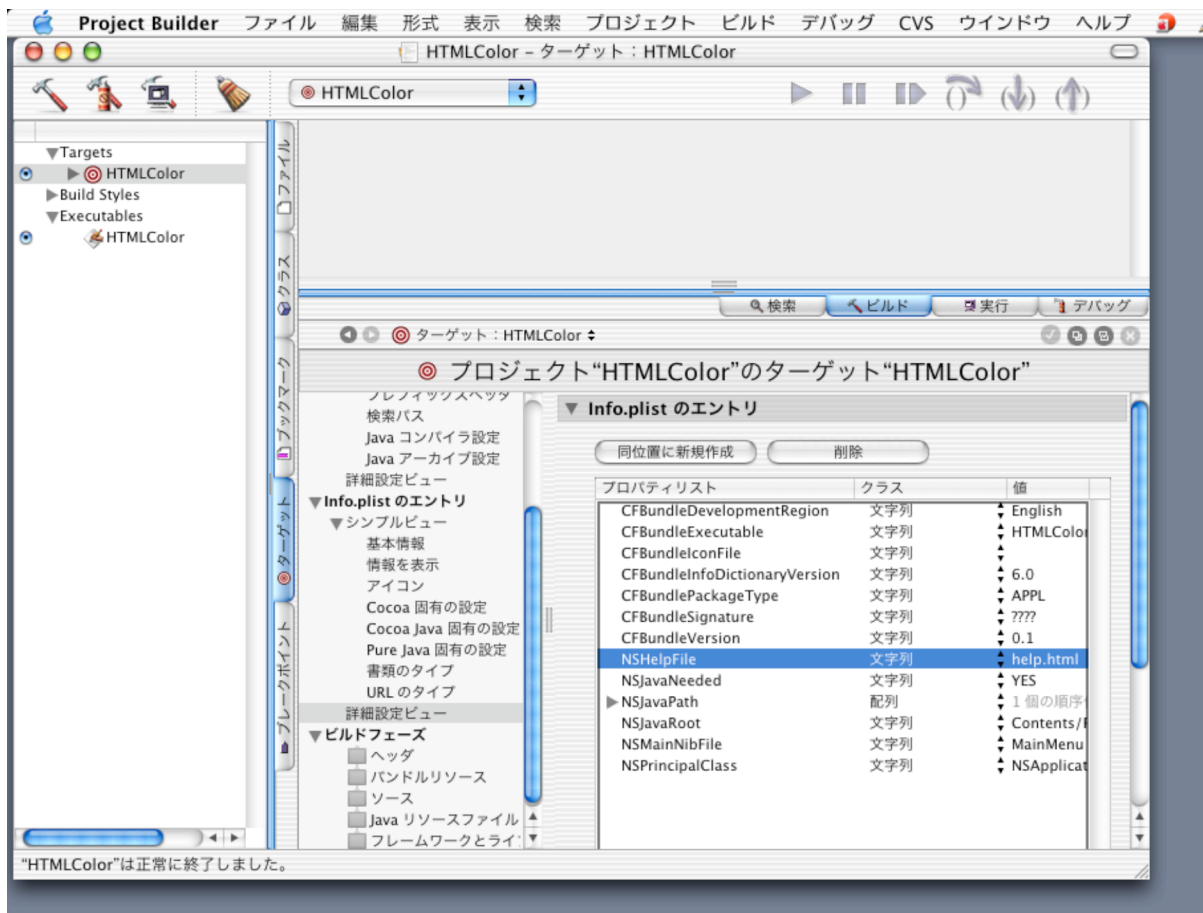


図2-012 ヘルプとして用いるHTMLファイルの指定

## ◆2.25 アイコンの設定

(p.89) アプリケーションのアイコンを指定するには、「ターゲット」タブを開き、「Targets」リストの「HTMLColor」を選択する。そして「Info.plistのエントリ」を選び、アイコンファイル欄に「HTMLColorAppIcon.icns」と入力する。

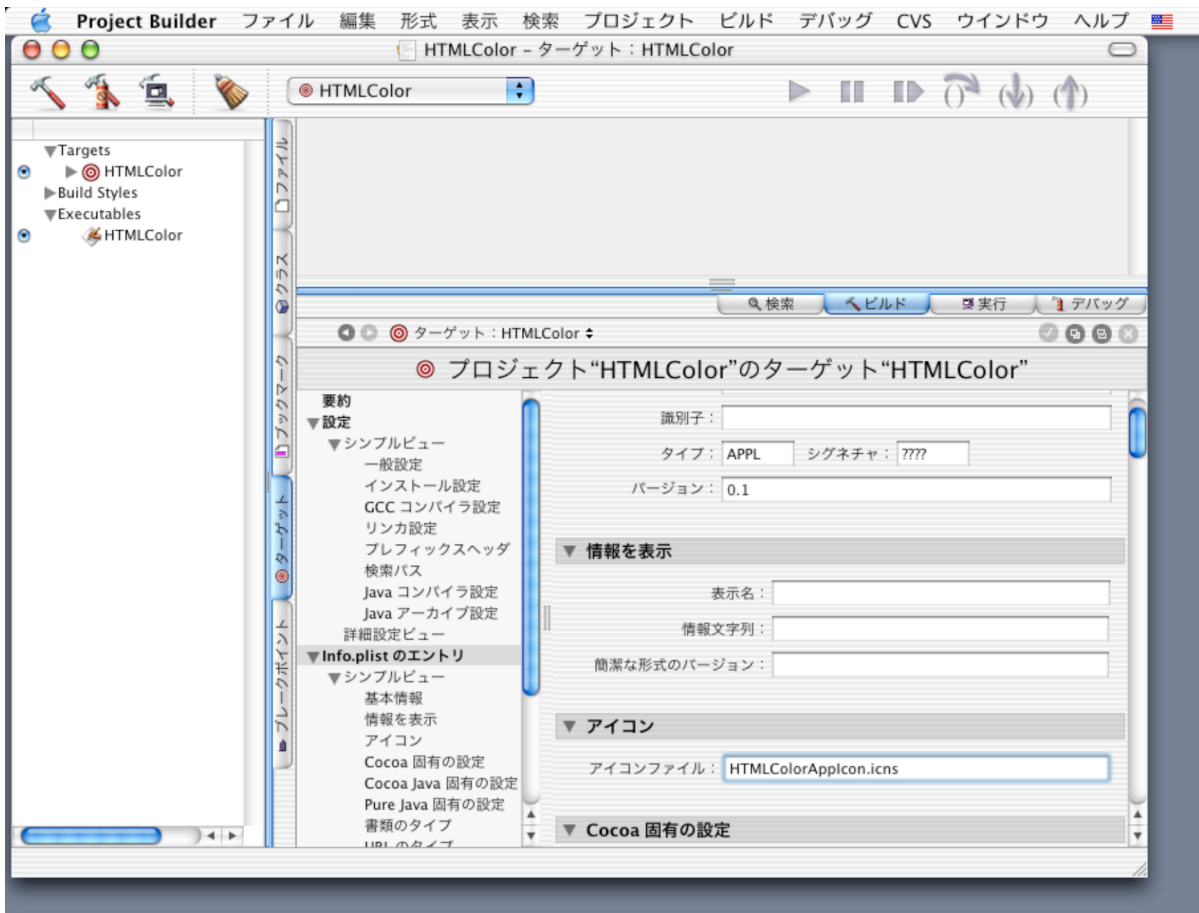


図2-013 アプリケーションのアイコンファイルの指定

## ◆2.26 アプリケーション情報の設定

(p.90) アプリケーション情報を設定するには、「ターゲット」タブを開き、「Targets」リストの「HTMLColor」を選択する。そして「Info.plistのエン트리」を選び、「基本情報」や「情報を表示」の各項目に適切なテキストを入力する。

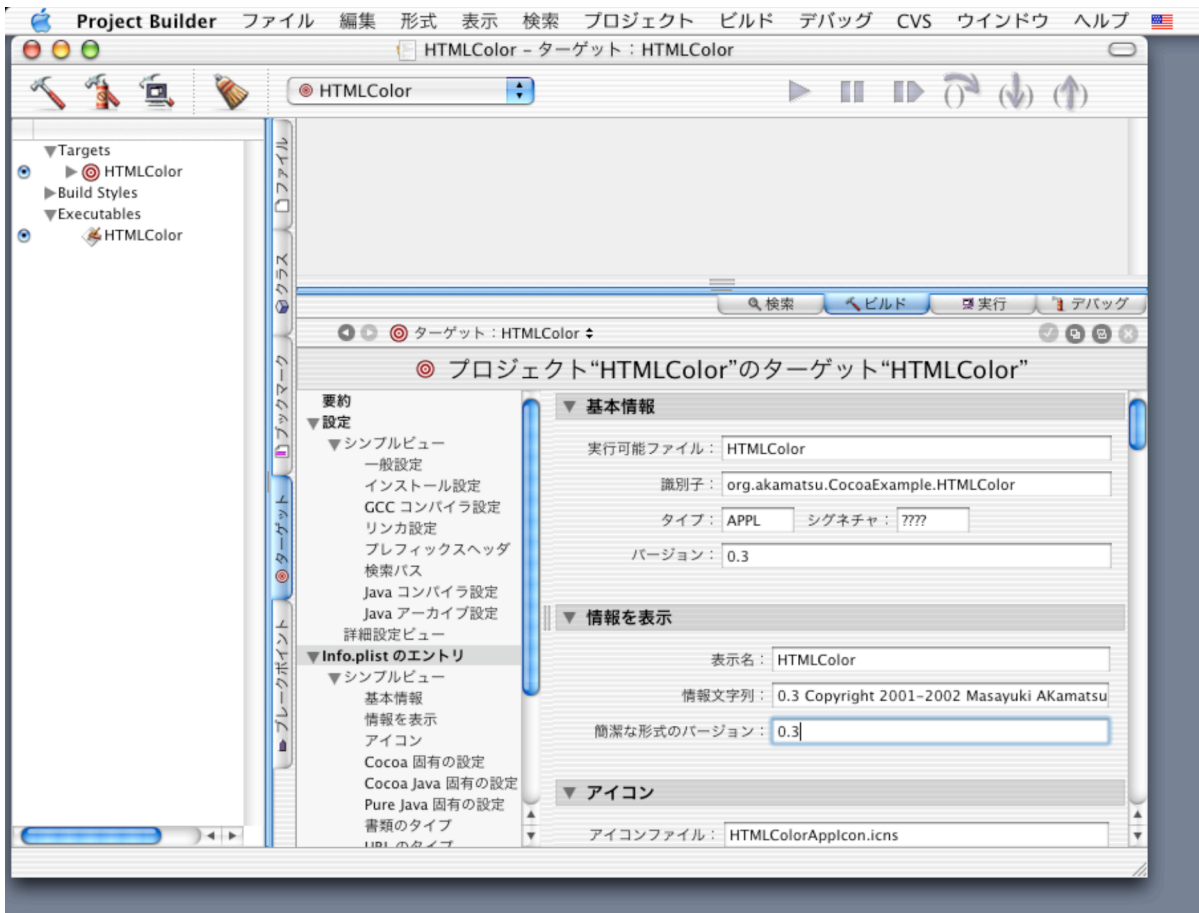


図2-014 アプリケーション情報の設定

(p.91) アプリケーションのビルドに関する設定は、「ターゲット」タブを開き、「Targets」リストの「HTMLColor」を選択する。そして、「設定」を選び、各種項目を設定する。

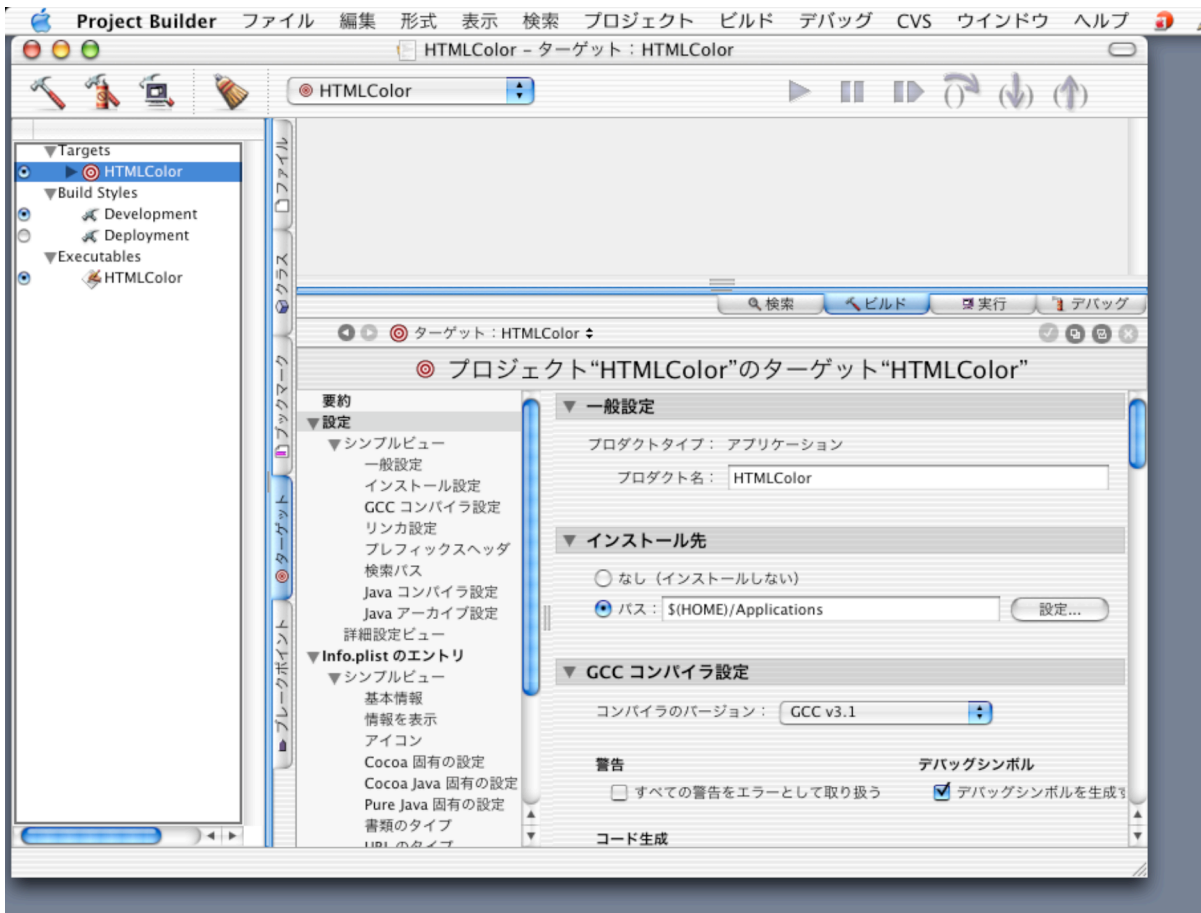


図2-015 ビルドに関する設定

## ◆2.27 ローカライズ

(p.97) 各種のローカライズ作業を行った後、アプリケーションをビルドして実行しても、ローカライズ作業が反映されないことがある。例えば、HTML形式のヘルプファイルをローカライズしても、アプリケーション実行時には英語で表示されるだろう。これは、ローカライズしていないグローバルであるHTMLファイルがビルドファイルの中に残っているためである。このような場合は、ツールバーの帚(ほうき)アイコンのクリーンボタンをクリックして(またはビルドメニューから「クリーニング」を選んで)、ターゲットをクリーニングし、再度ビルドし、実行すれば良い。

## 第3章 さまざまなアプリケーション例

### ◆3.1 ドキュメント型アプリケーション

#### ■ドキュメントのアウトレット設定

(p.106) MyDocumentクラスに追加したtextViewアウトレットのタイプは、「NSTextView」を指定する。

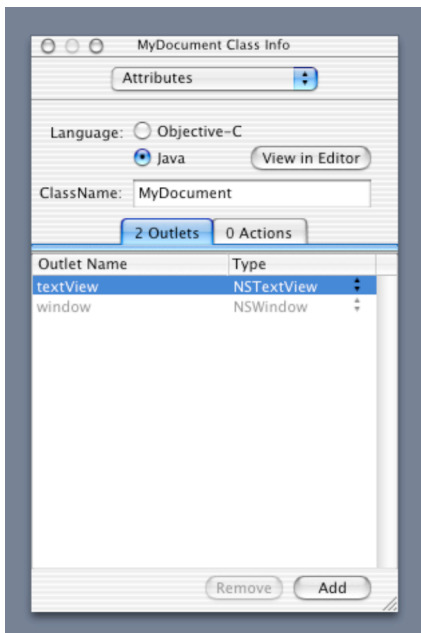


図3-001 「textView」アウトレットのタイプ指定

#### ■ドキュメントのタイプ設定

(p.108) ドキュメントのタイプ設定を行うには、「ターゲット」タブをクリックし、「Targets」リストの「FirstText」を選択する。そして「Info.plistのエントリ」にある「書類のタイプ」で行う。

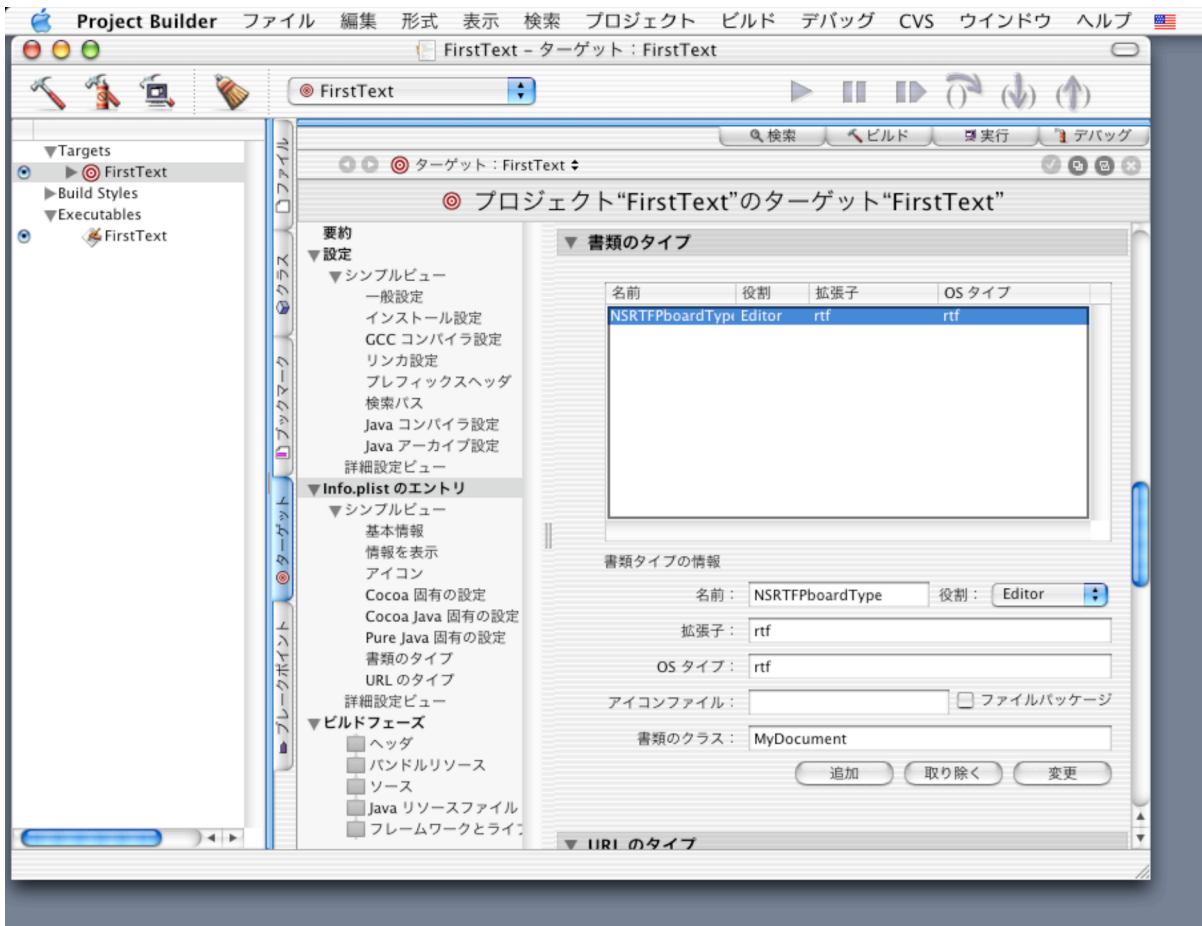


図3-002 ドキュメントタイプの設定

### ◆3.2 タイマとサウンドの利用

#### ■ユーザインターフェースの作成

(p.117) スライダを動かした時に連続的にテンポ表示の数値が変わるようにするためには、スライダの設定で「options:」欄の「Continuous」にチェックを付ける。

(p.118) 「Start」ボタンのアイコン位置として、Infoウィンドウの「Icon Position:」欄で左上のアイコンを選んで、アイコンが無いことを指定する。また、このボタンのテキストは、フォントパネルを開いて13ポイントに指定すると良い。

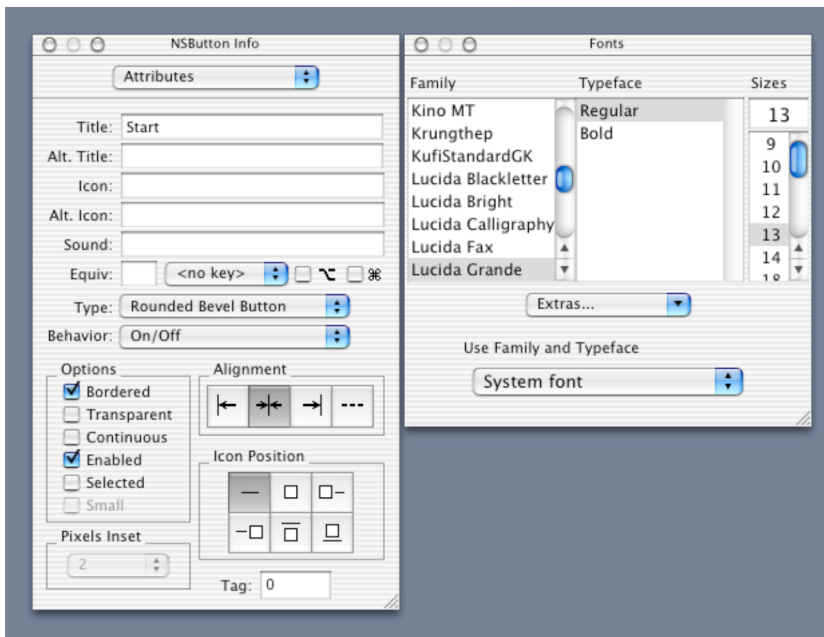


図3-003 「Start」 ボタンのアイコン位置指定とフォント指定

### ■コントローラクラスの作成

(p.119) MetroControllerクラスに追加したアウトレットのタイプとして、beat0からbeat3までとtoggleButtonは「NSButton」を指定する。また、tempoFieldは「NSTextField」、tempoSliderは「NSSlider」を指定する。

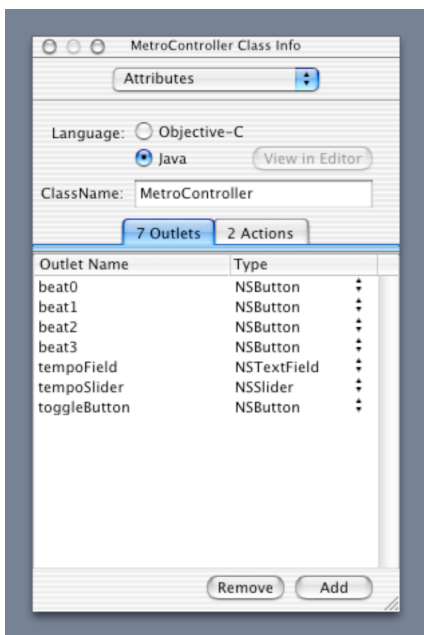


図3-004 MetroControllerクラスに追加したアウトレット

### ■インスタンスフィールドの定義

(p.125) 生成されたソースコードでは、アウトレットに対応するインスタンスフィールドのクラス型が指定されているので、修正する必要はない。なお、「NSTimer metroTimer;」以下のインスタンスフィールドは、本文通り追加する。

### ◆3.3 ベジエ曲線でグラフィックを描く

#### ■カスタムビューの作成

(p.138) カスタムビューのクラス指定は、Infoウィンドのポップアップメニューから「Custom Class」を選んで行う。

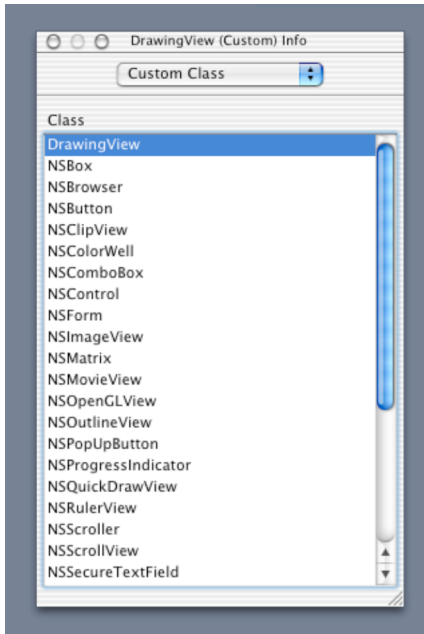


図3-005 カスタムビューのクラス指定

#### ■コントローラクラスの作成

(p.141) DrawingControllerクラスに追加したdrawingViewアウトレットのタイプは「DrawingView」を指定する。

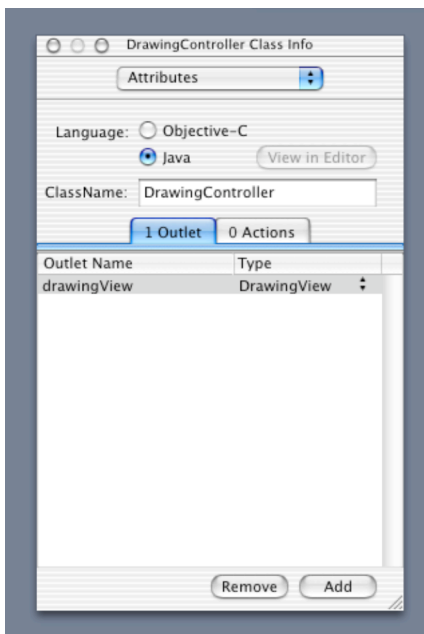


図3-006 DrawingControllerクラスに追加したアウトレット



## ■スライダの処理

(p.152) 生成されたソースコードでは、アウトレットに対応するインスタンスフィールドのクラス型が指定されているので、修正する必要はない。

## 第4章 開発ツールの詳細

### ◆4.1 Project Builder

#### ■Project Builderの構成

(p.161) 図4-2において、プロジェクトウィンドウの下部のバーは「ステータスバー」である。左下に「ツールバー」と記述されているのは誤り。

#### ■ツールバー

(p.161) ツールバーの左側に表示されるボタンの並びが変更になった。以下の図のように、左から順にビルドボタン（金槌アイコン）、ビルドとデバッグボタン（金槌にスプレー缶が重なったアイコン）、ビルドと実行ボタン（金槌にディスプレイが重なったアイコン）、クリーンボタン（帚アイコン）である。

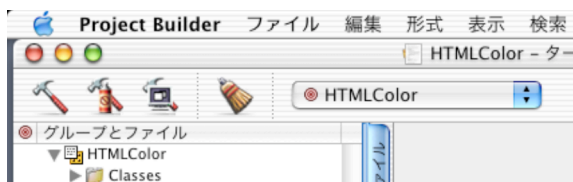


図4-001 ツールバー左側のボタンとポップアップメニュー

(p.162) ツールバーの右側に表示されるデバッグ関係のボタンの最も左側に、再起動ボタンが追加された。

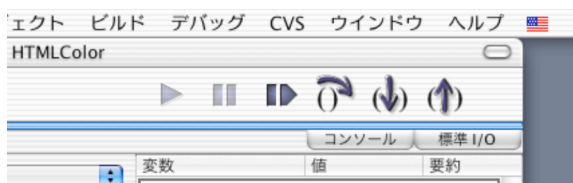


図4-002 ツールバー右側のデバッグ関係のボタン

(p.163) ツールバーをカスタマイズするには、表示メニューから「ツールバーをカスタマイズ...」を選ぶ。ツールバーに表示できるボタンの種類が増え、ボタンを小アイコンとして表示できるようになった。

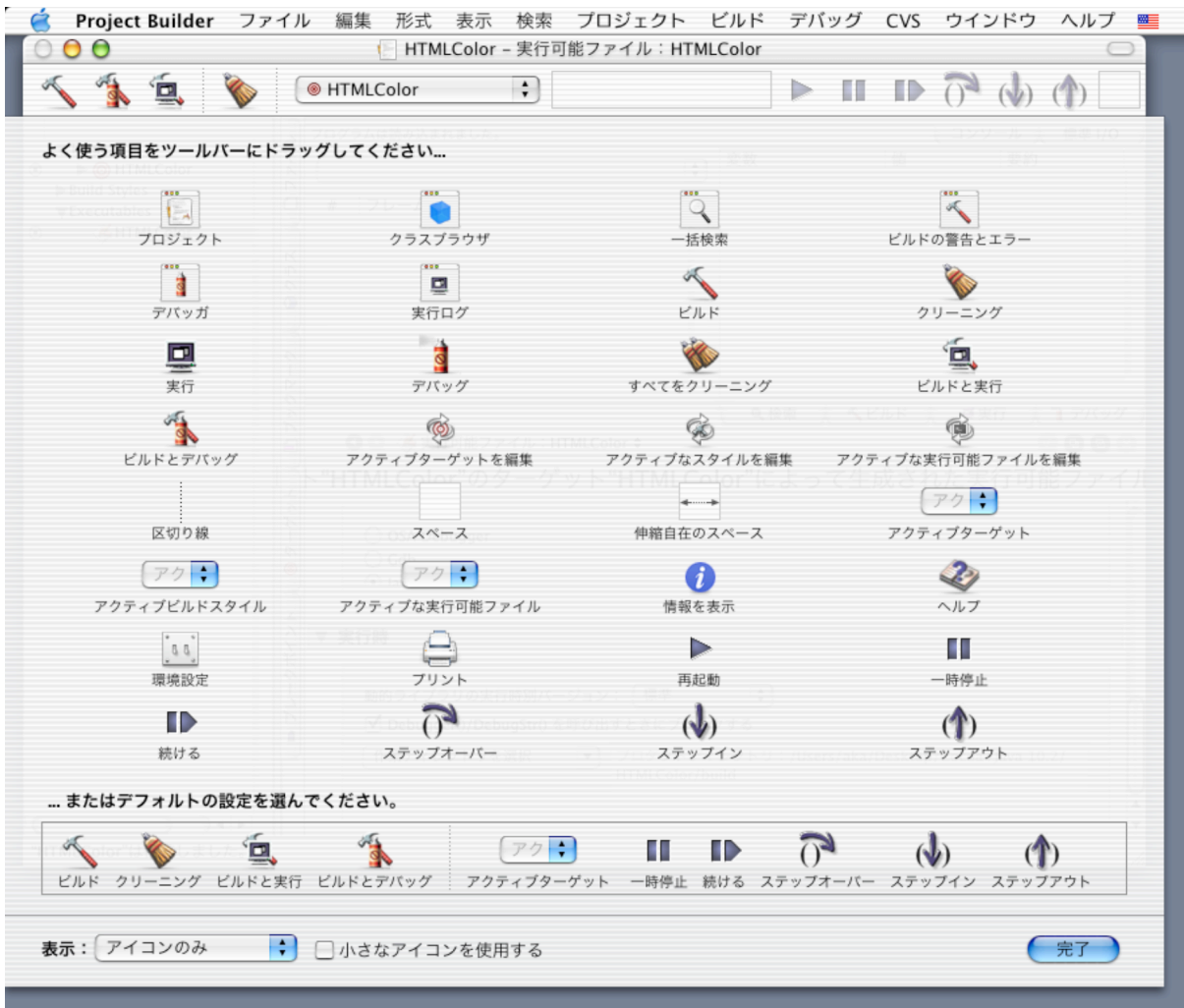


図4-003 ツールバーのカスタマイズ

## ■コンテンツリスト

(p.164) 「グループとファイル」リストにおいて、ファイルの左側にチェックボックスが表示されるようになった。このチェックボックスのチェックを外すと、ビルド処理の対象外となる。

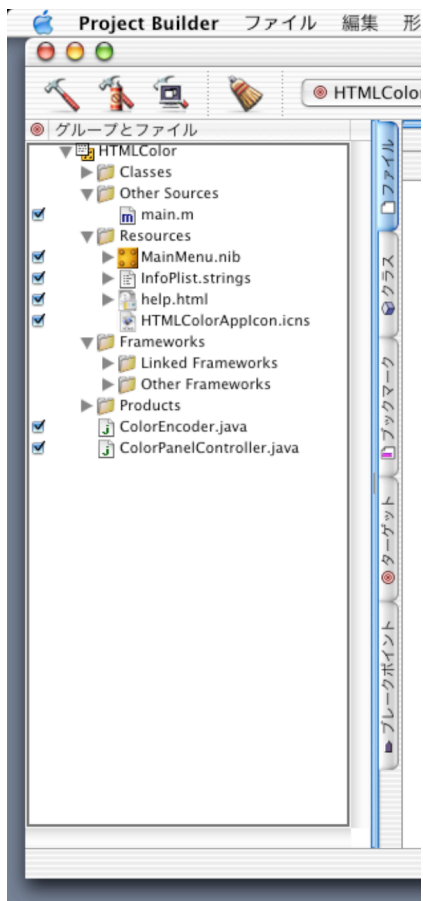


図4-004 「グループとファイル」リスト

(p.166) 「ターゲット」リストの「Targets」で選択したターゲットの設定は、新しい形式のエディタで行うようになった。エディタの左側のリストに設定項目が表示され、リストで選んだ項目を右側のエリアで設定する。

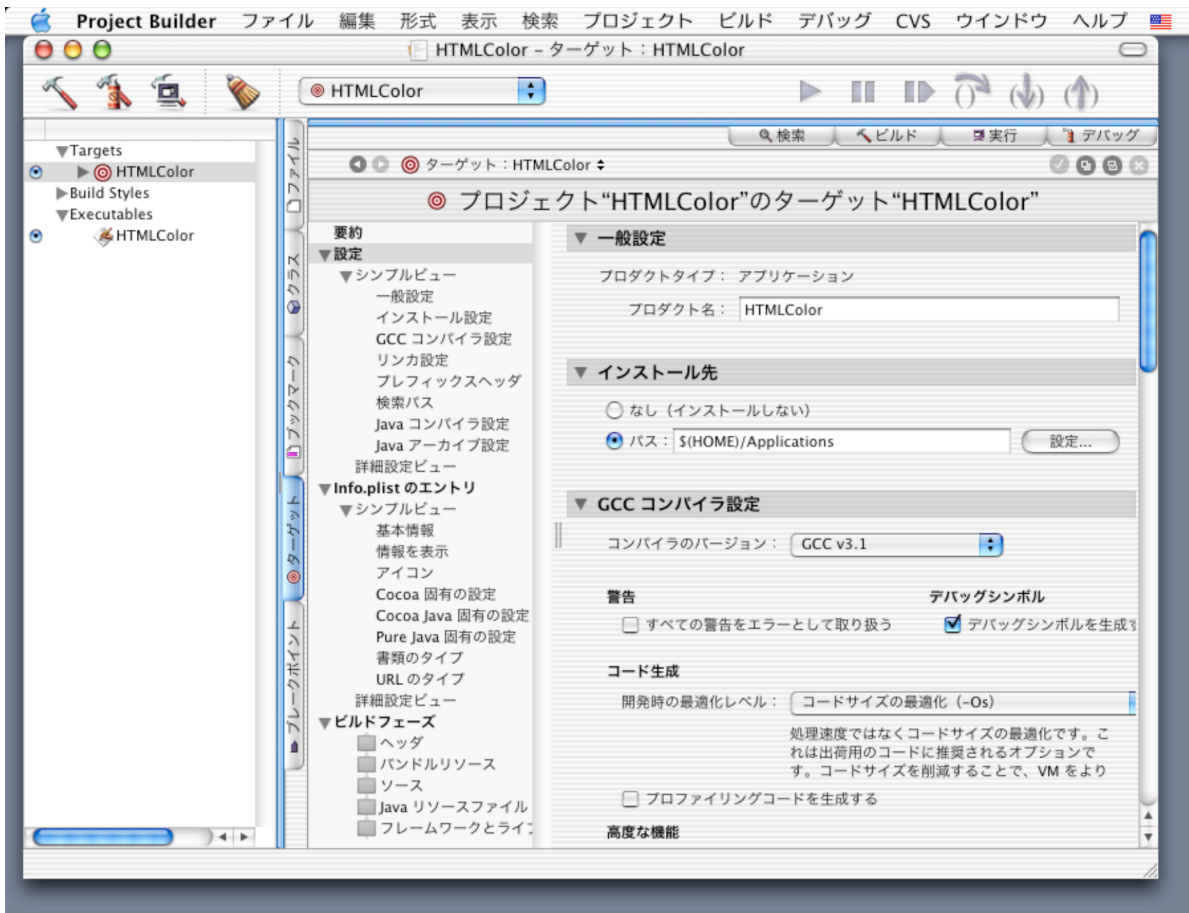


図4-005 ターゲットの設定

(p.167) 「ターゲット」リストの「Targets」で選択した実行可能ファイルの設定は、新しい形式のエディタで行うようになった。

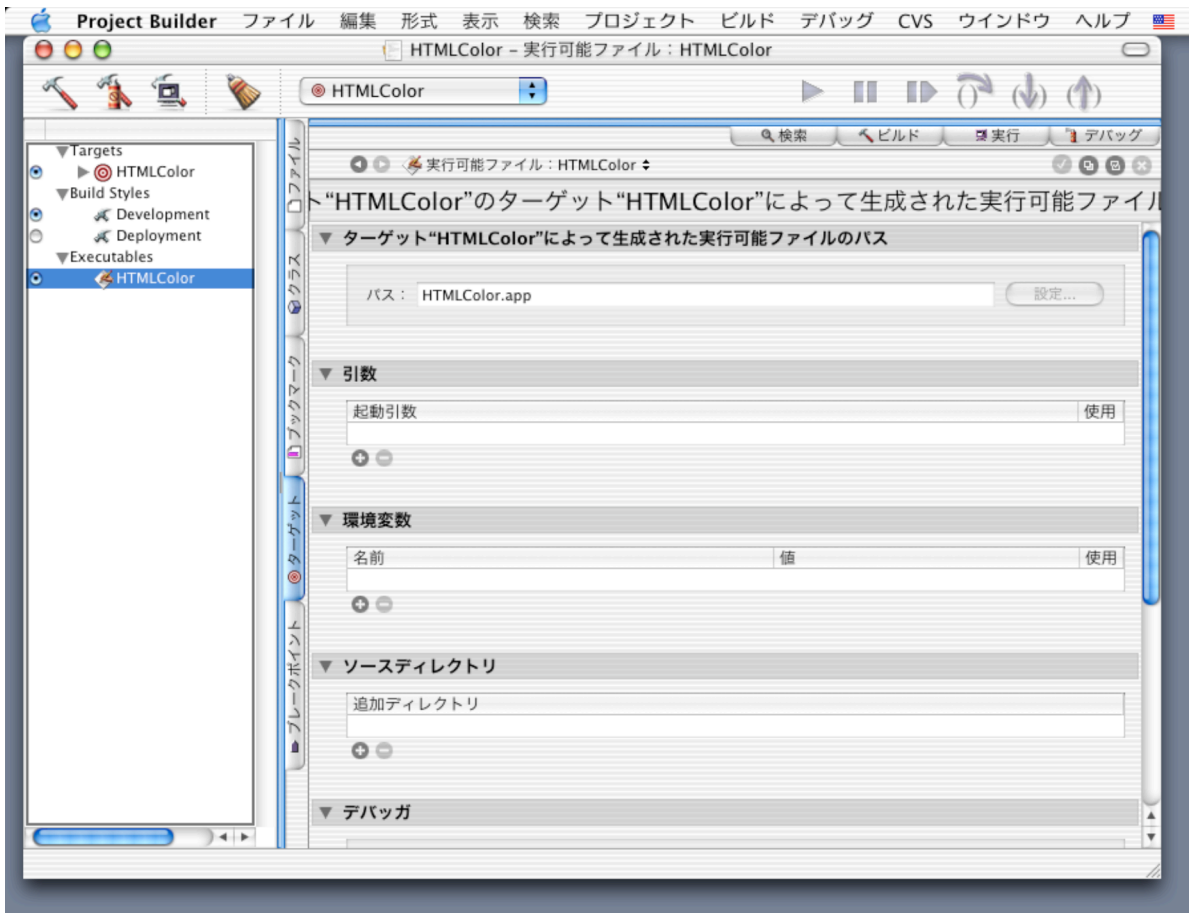


図4-006 実行可能ファイルの設定

## ■エディタ

(p.171) ソースコードエディタの上部にあるナビゲーションバーの右側に表示されるボックスのアイコンが変更された。左から順に、構文チェック、対応するものに移動、エディタを分割、分割しているエディタを統合、の役割を担う。

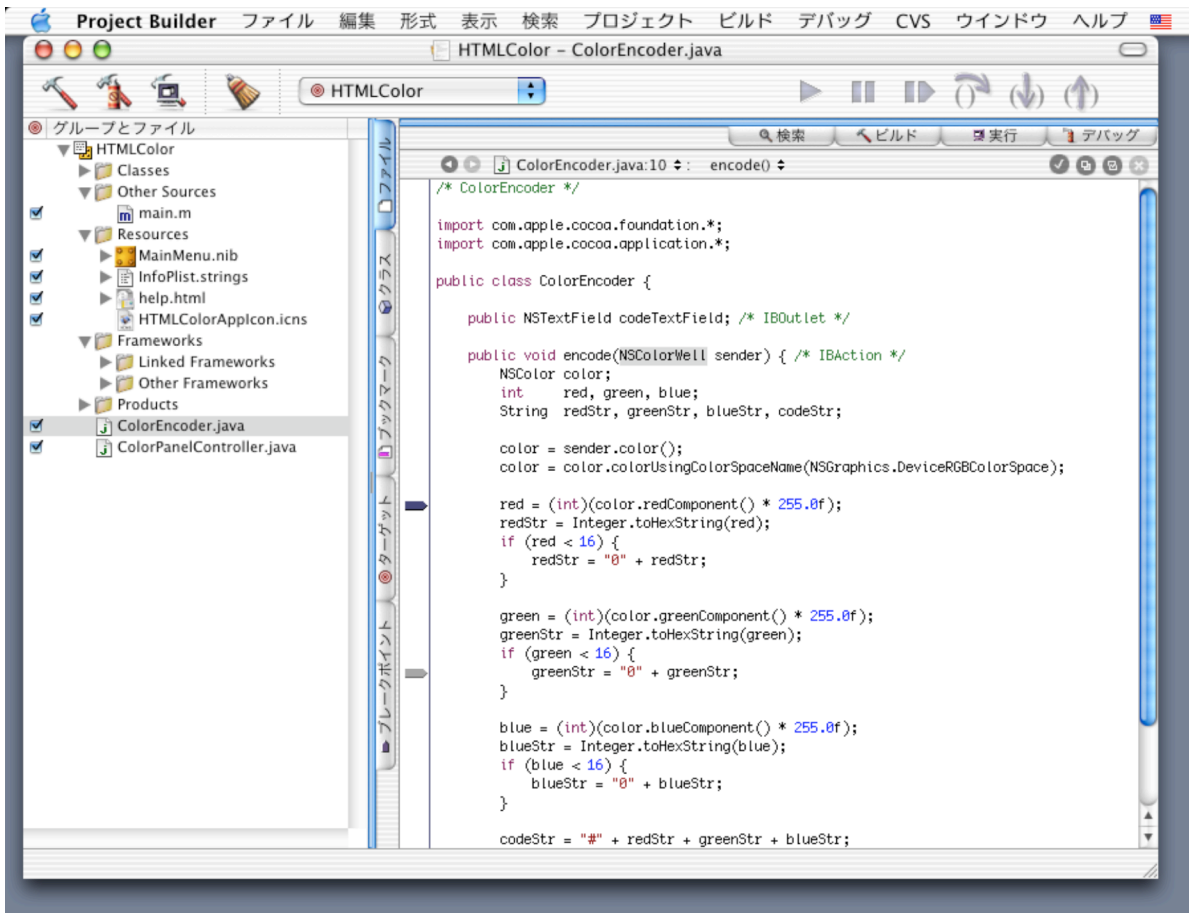


図4-007 ソースコードエディタ

(p.171) ソースコードエディタなどに表示されている単語をoptionキーを押しながらダブルクリックすれば、その単語が検索パネルの検索文字列に取り込まれ、自動的に検索が開始される。また、その単語がクラス名やメソッド名である場合は、そのドキュメントが表示される。

(p.172) 従来の「ビルド設定」パネルや「アプリケーション設定」パネル、「詳細設定」パネル、「実行可能ファイル」パネルは廃止され、ターゲットの設定エディタと実行可能ファイルの設定エディタに統合された。

#### ■操作メニュー

(p.185) 従来の操作メニューは表示メニューに変更された。

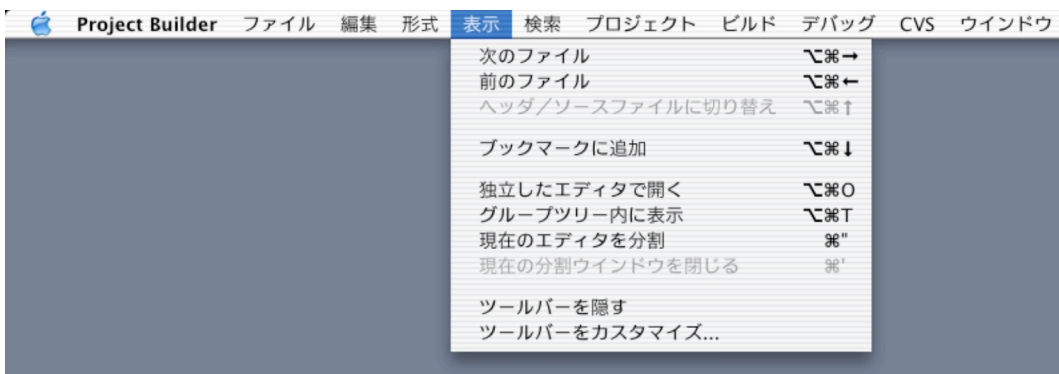


図4-008 表示メニュー

## ■SCMメニュー

(p.192) 従来のSCMメニューはCVSメニューに変更された。



図4-009 CVSメニュー

## ■プリファレンス

(p.195) プリファレンスウィンドウが拡張され、「タスクテンプレート」、「インデント設定」、「ファイルタイプ」、「Documentation Viewing」の設定が追加された。また、従来からの設定も項目の追加変更が行われている。

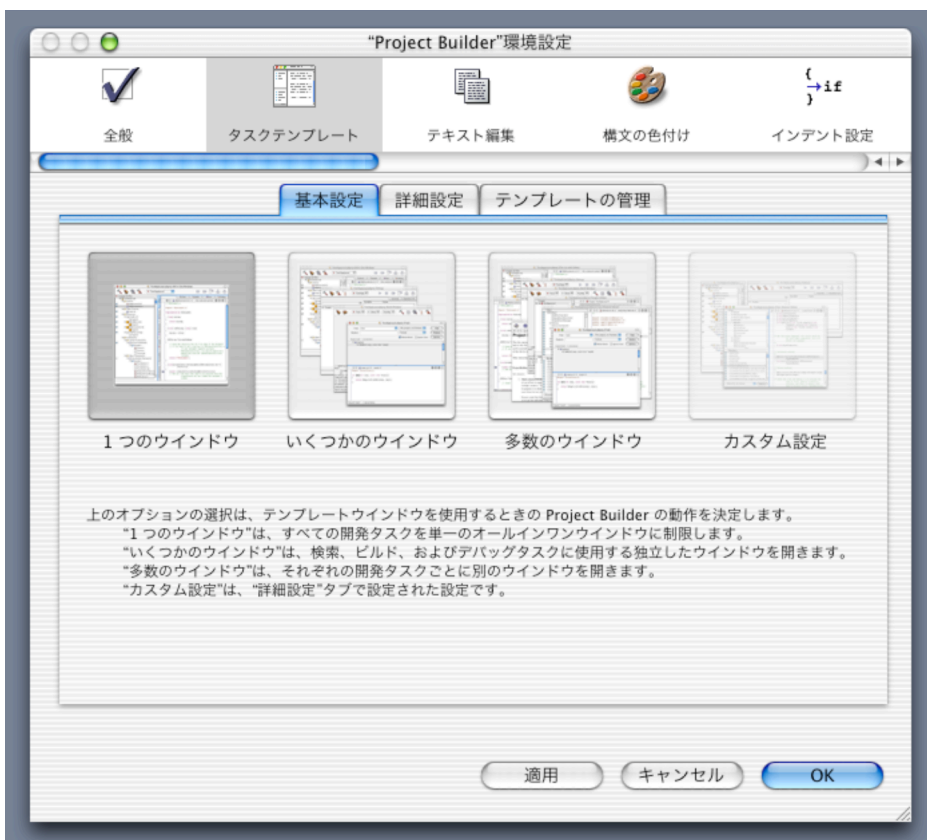


図4-010 プリファレンスウィンドウ

## ◆4.2 Interface Builder

### ■パレット



(p.212) 「Views」パレットに表示される固定テキストが3種類になった。これらはフォントサイズが異なるだけで、同じビューである。

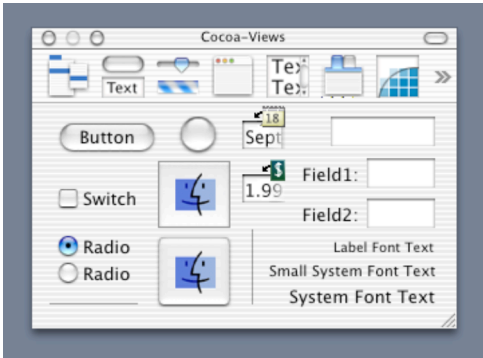


図4-011 「Views」パレット

(p.212) 「Other」パレットの下部中央に回転式のプログレス・インジケータが加わった。

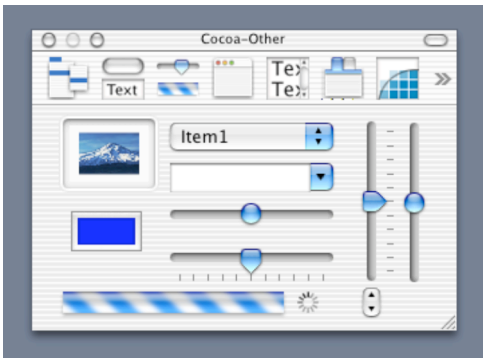


図4-012 「Other」パレット

(p.214) 「AppleScript」パレットが加わった。青い立方体アイコンで表される「ASKDataSource」オブジェクトをインスタンスとして用いることができる。

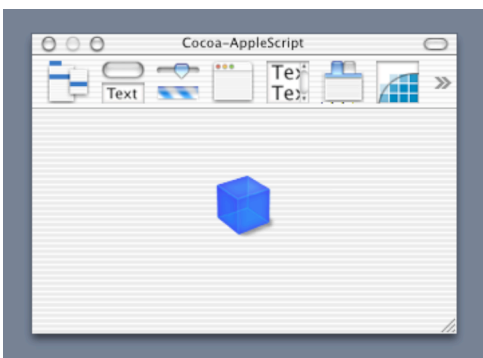


図4-013 「AppleScript」パレット

## ■プリファレンス

(p.234) プリファレンスウィンドウに「Editing」パネルが追加された。このパネルでは編集に関する設定を行う。

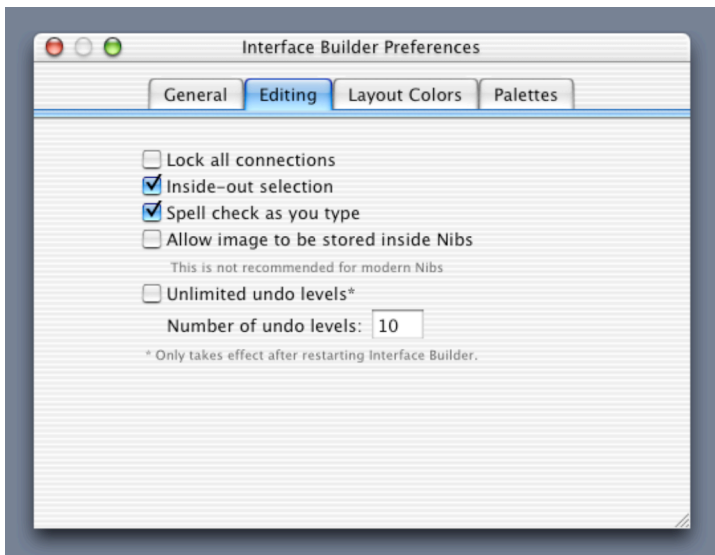


図4-014 「Editing」 パネル

## 第5章 Cocoaフレームワーク

### ◆5.1 Application Kitフレームワーク

#### ■Application Kitクラス

Application Kitに追加されたクラスは以下の通り。

NSGlyphInfo	文字列のGlyph情報を扱うクラス。
NSTabViewItem	タブによって切り替えられる内容を扱うクラス。

なお、p.249の下から5行目はNSStatusItemとなっているが、NSStepperCellが正しい。

NSStepperCell	ステッパーの表示やマウス処理を扱うクラス。
---------------	-----------------------

#### ■Application Kitインターフェース

Application Kitに追加されたインターフェースは以下の通り。

NSValidatedUserInterfaceItem	コントロールを自動的に切り換えるユーザインターフェースのためのインターフェース。
------------------------------	--

### ◆5.2 Foundationフレームワーク

#### ■Foundationクラス

Foundationに追加されたクラスは以下の通り。

NSAppleEventDescriptor	AppleEventを記述するためのクラス。
NSAppleScript	AppleScriptのロード、コンパイル、実行を行うクラス。
NSKeyedArchiver	オブジェクトをキーを持ったアーカイブにエンコードするクラス。
NSKeyedUnarchiver	キーを持ったアーカイブからオブジェクトをデコードするクラス。
NSNameSpecifier	集合から名前によってオブジェクトを指定するクラス。
NSUniqueIDSpecifier	集合から重複しないIDによってオブジェクトを指定するクラス。

#### ■Foundationインターフェース

Foundationに追加されたインターフェースは以下の通り。

NSCoding	クラス定義のコーディングのためのインターフェース。
----------	---------------------------

(以上)