NSServicesメニューの実装

mindtools@mac.com

概要

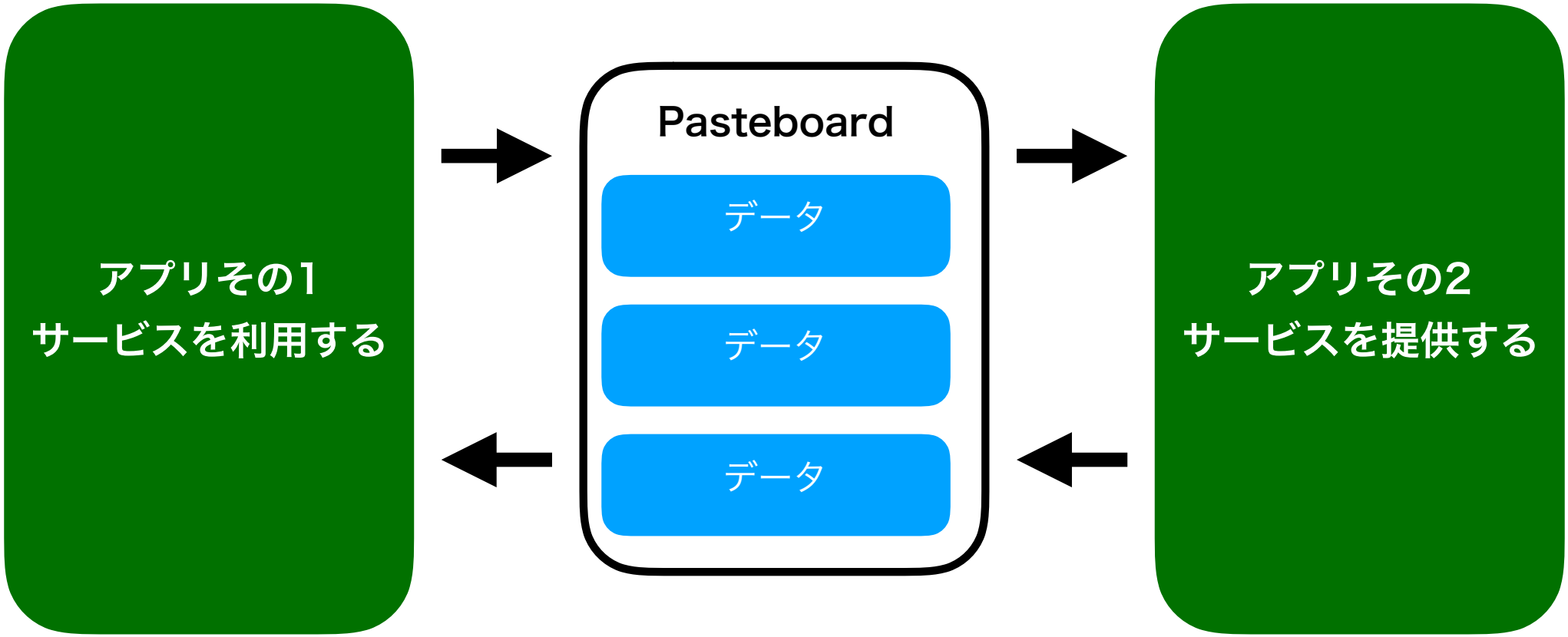
- NSServicesは何か？
- 何が出来るか？

OS内での位置づけ

GUIから使える、Pastebordサーバーを利用した、プロセス間通信。

- Copy & Paste
- Drag & Drop
- Services

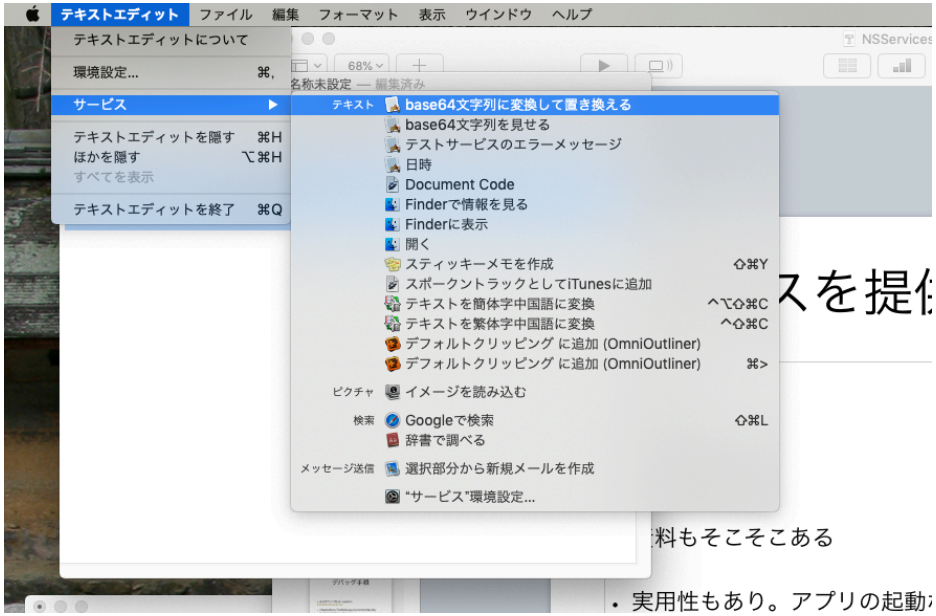
OS内での位置づけ



考えたかは、コピペやD&Dと同じ、GUIと実装が違う。

サービスを提供するアプリ

- 作るのが簡単
- 資料もそこそこある
- 実用性もあり。アプリの起動など。



実装概要

- info.plistの編集
  - NSServicesキーに、各種設定を書いてゆく
- メソッドの実装
  - NSServicesキーに書かれたメソッド名に対応するメソッドを書く

デバッグ概要

- pbdコマンド(flush, update)
  - /System/Library/CoreServices/pbs -flush
    - Servicesメニューのキャッシュデータをクリア
  - /System/Library/CoreServices/pbs -update
    - Servicesメニューのキャッシュデータをupdate

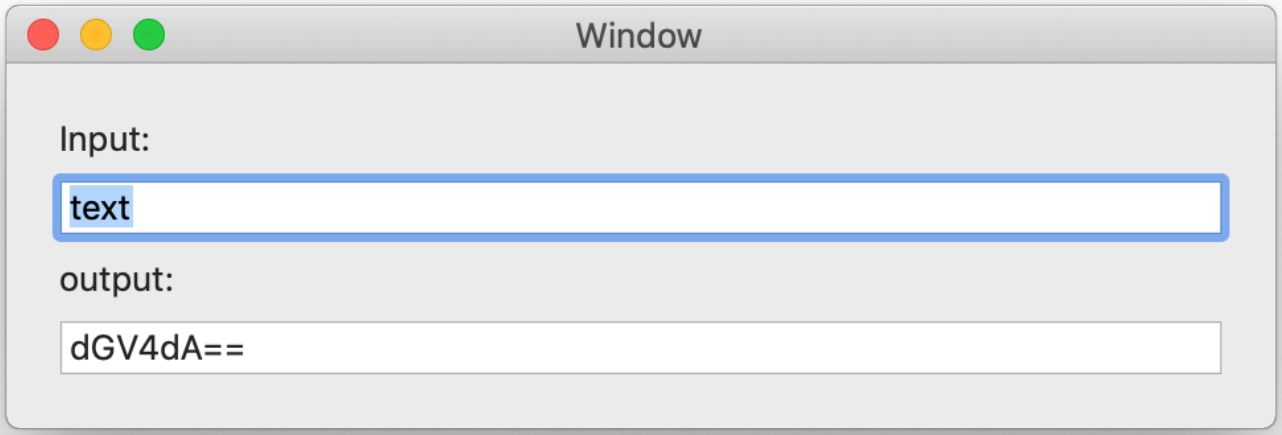
デバッグ概要

- /Applications/TextEdit.app/Contents/MacOS/TextEdit -NSDebugServices com.mindto01s.mac.BASE64
  - TextEditを立ち上げて、Servicesメニュー関連のデバッグメッセージを表示する
  - “com.mindto01s.mac.BASE64”の部分はデバッグするServicesのバンドルID

例0

- サービスのコードを説明するための土台です。
- base64の表示のGUIだけを作ります。
- コードは”BASE64_0”に入っています。
- このアプリは、Servicesのコードは何も含まれていません。

例0:外見



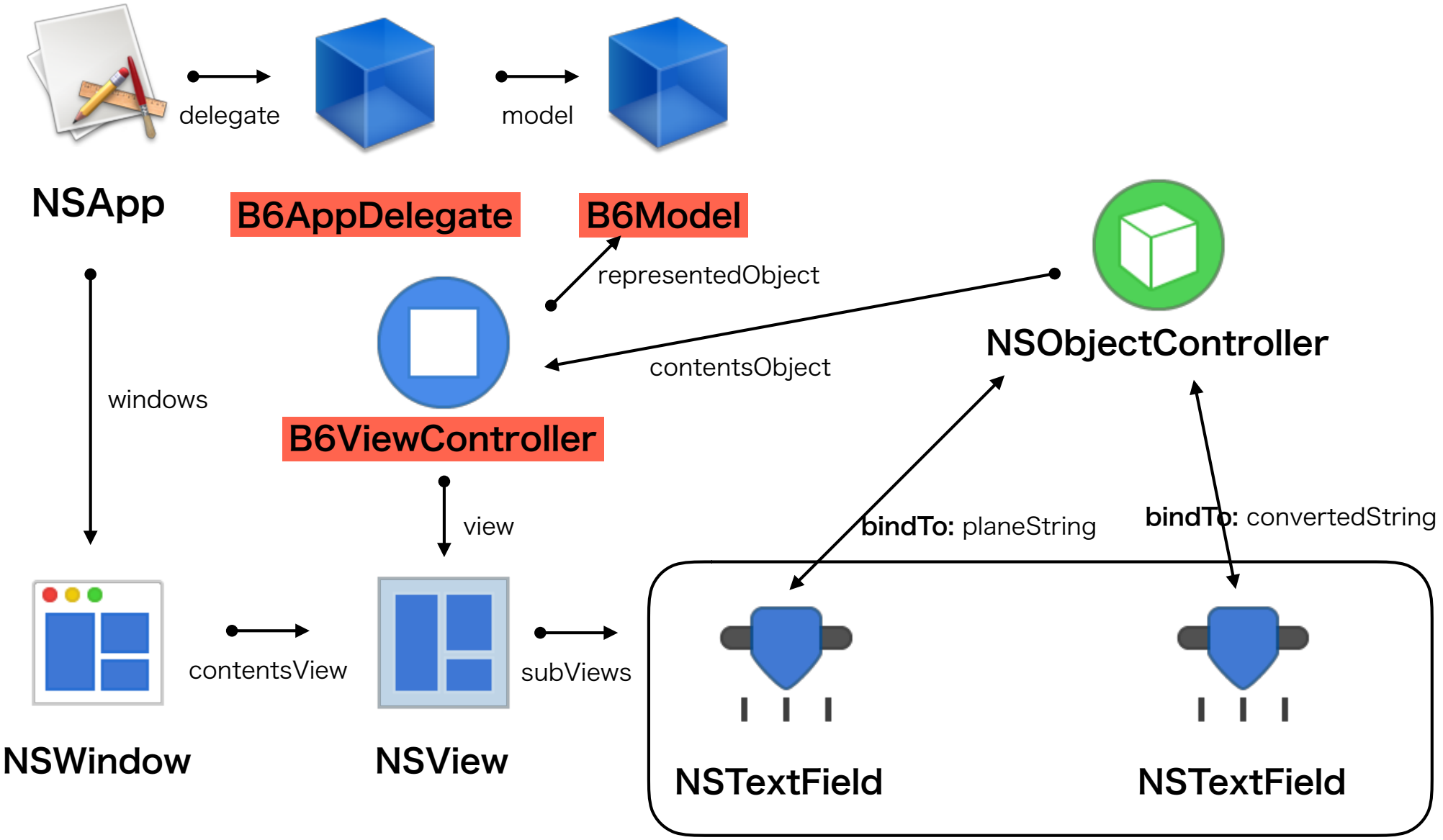
例0:ファイル一覧

```
├── BASE64
│   ├── Assets.xcassets
│   ├── B6AppDelegate.swift
│   ├── B6Model.swift
│   ├── B6ViewController.swift
│   ├── BASE64.entitlements
│   ├── Base.lproj
│   │   └── Main.storyboard
│   └── Info.plist
└── BASE64.xcodeproj
```

例0:クラス一覧

クラス名	目的
B6AppDelegate	いつものAppDelegate
B6Model	文字列をbase64文字列に変換するだけのモデルクラス
B6ViewController	GUIの都合で作った。initalFirstResponderの指定と、modelオブジェクトへのショートカットを指定している。

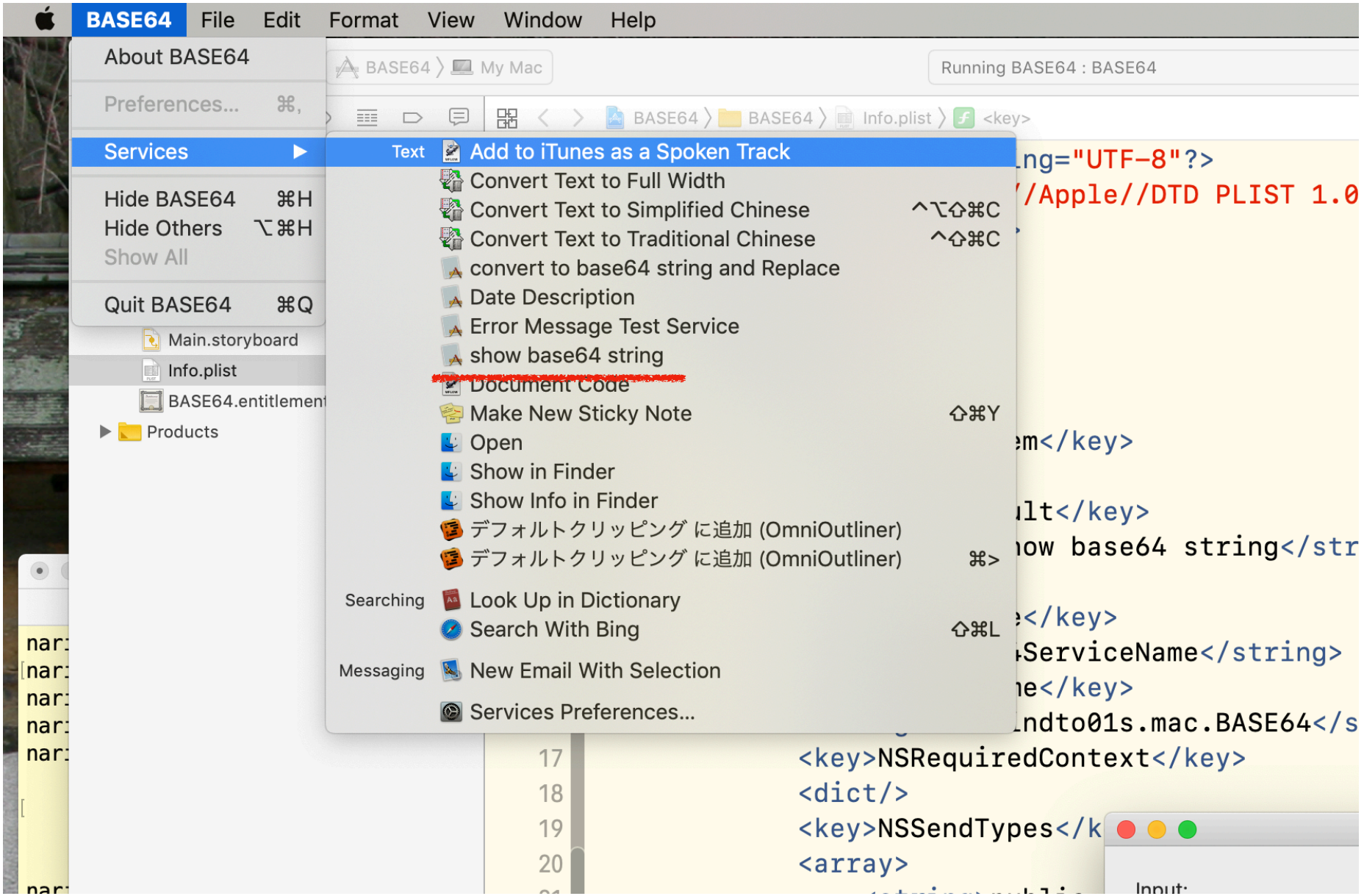
例0:オブジェクトモデル



例1

- TextEditなどで、選択された文字列を取得して、BASE64アプリで表示するだけのサービスメニュー

例1:外見


BASE64
File
Edit
Format
View
Window
Help
About BASE64
Preferences... ⌘,
Services
Hide BASE64 ⌘H
Hide Others ⌥⌘H
Show All
Quit BASE64 ⌘Q
Main.storyboard
Info.plist
Products
Text
Add to iTunes as a Spoken Track
Convert Text to Full Width
Convert Text to Simplified Chinese ^⌥⇧⌘C
Convert Text to Traditional Chinese ^⇧⌘C
convert to base64 string and Replace
Date Description
Error Message Test Service
show base64 string
Document Code
Make New Sticky Note ⇧⌘Y
Open
Show in Finder
Show Info in Finder
デフォルトクリッピング に追加 (OmniOutliner)
デフォルトクリッピング に追加 (OmniOutliner) ⌘>
Searching
Look Up in Dictionary
Search With Bing ⇧⌘L
Messaging
New Email With Selection
Services Preferences...
Running BASE64 : BASE64
<key>NSRequiredContext</key>
<dict/>
<key>NSSendTypes</k
<array>
Input:

以下の3つは何も変わらない

ファイル一覧

クラス一覧

オブジェクトモデル

追加された物

起動時の登録

```
func applicationDidFinishLaunching(_ aNotification: Notification)
{
    // サービスのメソッドを実装するオブジェクトの登録
    NSApp.servicesProvider = self

    // OSへサービスメニューの内容のUpdateを依頼する
    NSUpdateDynamicServices()
}
```

B6AppDelegateにメソッド

```
@objc func base64ServiceName(_ pasteboard: NSPasteboard,
                             userData: String?,
                             error: NSErrorPointer)
{
    // ドット表記だとKVOされなかったので、
    // setValueForKeyを使いKVOでUIに反映させる
    self.model?.setValue(pasteboard.string(forType: .string),
                         forKey: "planeString")
}
```

plistにNSServices

```
<key>NSServices</key>
<array>
    <dict>
        <key>NSMenuItem</key>
        <dict>
            <key>default</key>
            <string>show base64 string</string>
        </dict>
        <key>NSMessage</key>
        <string>base64ServiceName</string>
        <key>NSPortName</key>
        <string>com.mindto01s.mac.BASE64</string>
        <key>NSRequiredContext</key>
        <dict/>
        <key>NSSendTypes</key>
        <array>
            <string>public.plain-text</string>
            <string>public.rtf</string>
            <string>com.apple.flat-rtfd</string>
        </array>
    </dict>
</array>
```

メニューに表示される

メソッド名

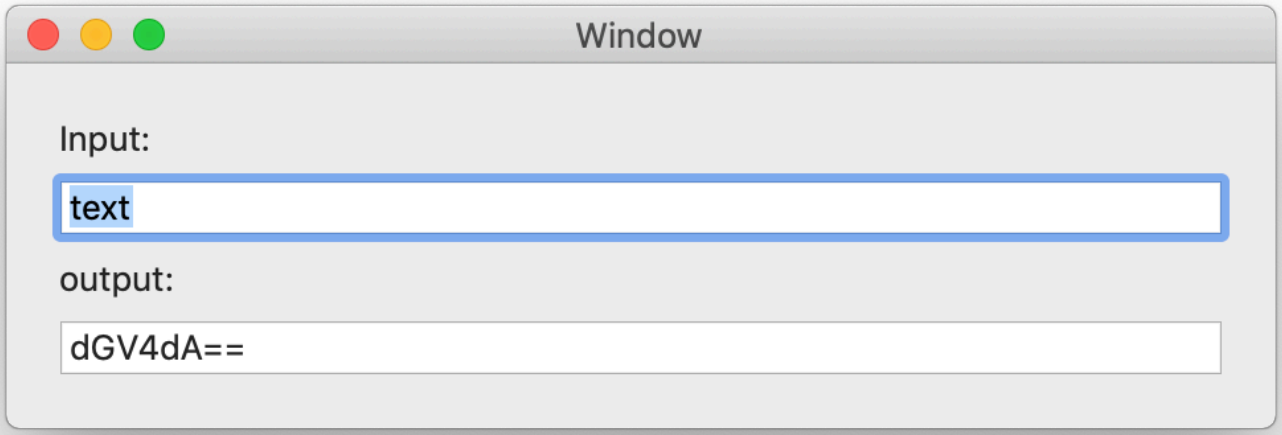
アプリのBundleID

必要な条件、XCodeだけとか

送られてくるデータのUTI

注意事項：

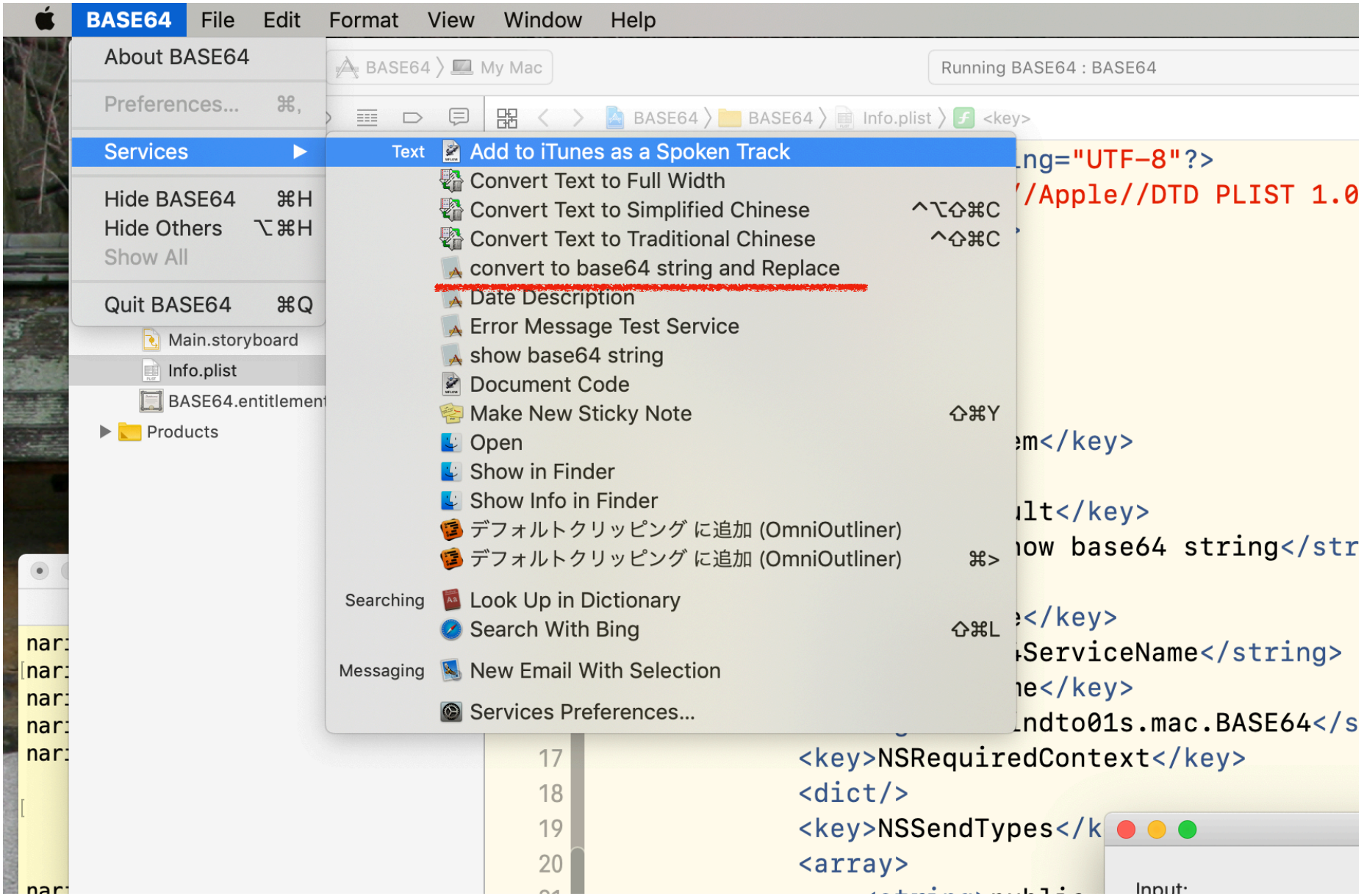
サービスを提供するアプリが前面にくる。



例2

- TextEditなどで、選択された文字列を取得して、BASE64で置き換えるだけのサービスメニュー

例2:外見


BASE64
File
Edit
Format
View
Window
Help
About BASE64
Preferences... ⌘,
Services
Hide BASE64 ⌘H
Hide Others ⌥⌘H
Show All
Quit BASE64 ⌘Q
Main.storyboard
Info.plist
Products
Text
Add to iTunes as a Spoken Track
Convert Text to Full Width
Convert Text to Simplified Chinese ^⌥⇧⌘C
Convert Text to Traditional Chinese ^⇧⌘C
convert to base64 string and Replace
Date Description
Error Message Test Service
show base64 string
Document Code
Make New Sticky Note ⇧⌘Y
Open
Show in Finder
Show Info in Finder
デフォルトクリッピング に追加 (OmniOutliner)
デフォルトクリッピング に追加 (OmniOutliner) ⌘>
Searching
Look Up in Dictionary
Search With Bing ⇧⌘L
Messaging
New Email With Selection
Services Preferences...
Running BASE64 : BASE64
<key>NSRequiredContext</key>
<dict/>
<key>NSSendTypes</k
<array>
Input:

以下の3つは何も変わらない

ファイル一覧

クラス一覧

オブジェクトモデル

追加された物

B6AppDelegateにメソッド

```
@objc func base64RetuenServiceName(_ pasteboard: NSPasteboard,
                                   userData: String?,
                                   error: NSErrorPointer)
{
    self.model?.setValue(pasteboard.string(forType: .string),
                         forKey: "planeString")

    // さらにペーストボードに値を設定する
    if let theReturnString = self.model?.convertedString
    {
        pasteboard.clearContents()
        pasteboard.setString(theReturnString as String, forType: .string)
    }
}
```

plistにNSServices

```
<dict>
    <key>NSMenuItem</key>
    <dict>
        <key>default</key>
        <string>convert to base64 string and Replace</string>
    </dict>
    <key>NSMessage</key>
    <string>base64RetuenServiceName</string>
    <key>NSPortName</key>
    <string>com.mindto01s.mac.BASE64</string>
    <key>NSRequiredContext</key>
    <dict/>
    <key>NSSendTypes</key>
    <array>
        <string>public.plain-text</string>
        <string>public.rtf</string>
        <string>com.apple.flat-rtfd</string>
    </array>
    <key>NSReturnTypes</key>
    <array>
        <string>NSStringPboardType</string>
    </array>
</dict>
```

メニューに表示される

メソッド名

アプリのBundleID

必要な条件

送られてくるデータのUTI

返すデータのUTIのはずだが、PboardTypeでないと動作しない

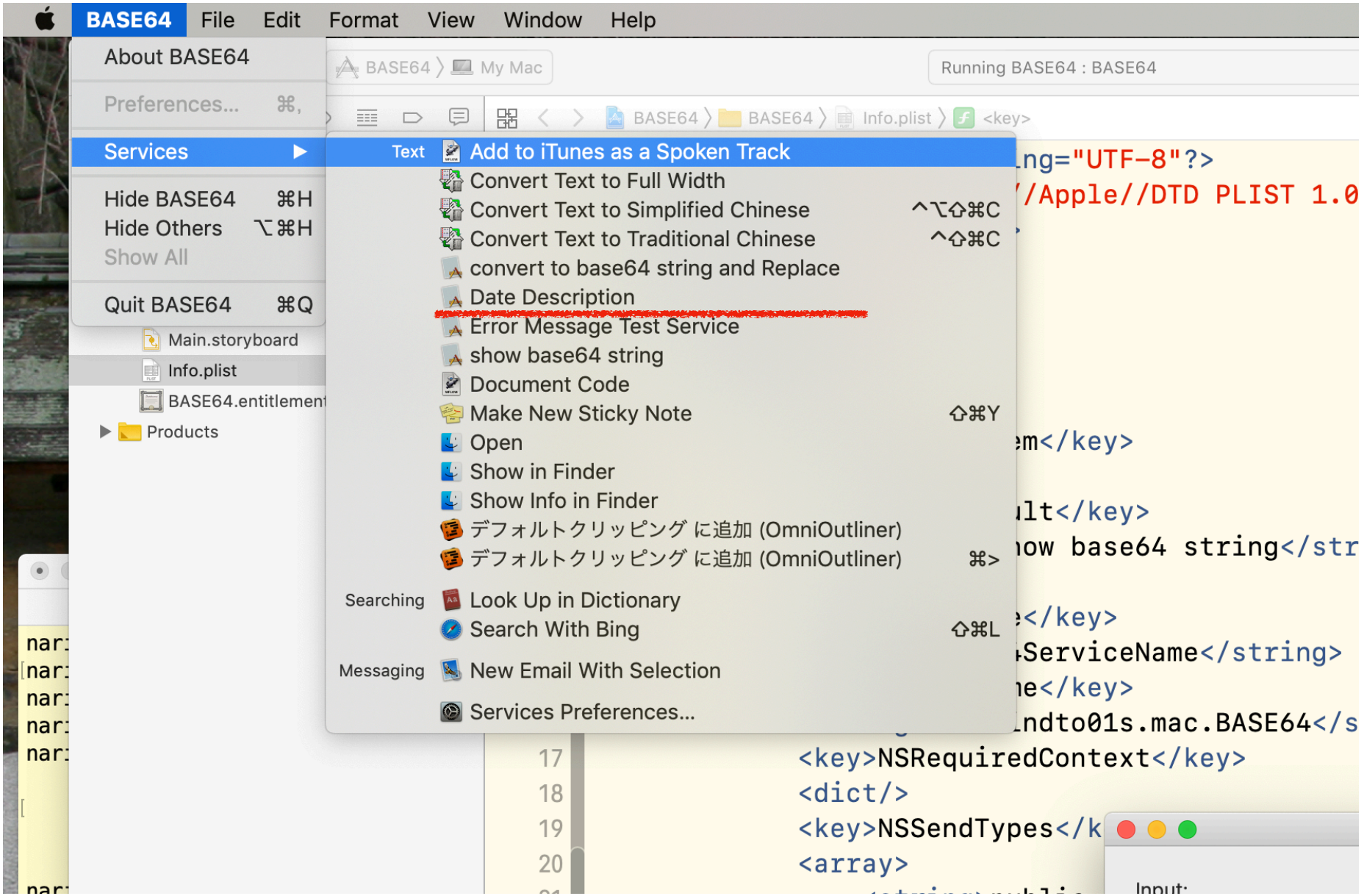
注意事項：

サービスを提供するアプリが前面にこない

例3

- TextEditなどで、選択された位置へ、現在時刻を挿入するだけのサービスメニューを挿入

例3:外見


BASE64
File
Edit
Format
View
Window
Help
About BASE64
Preferences... ⌘,
Services
Hide BASE64 ⌘H
Hide Others ⌥⌘H
Show All
Quit BASE64 ⌘Q
Main.storyboard
Info.plist
Products
Text
Add to iTunes as a Spoken Track
Convert Text to Full Width
Convert Text to Simplified Chinese ^⌥⇧⌘C
Convert Text to Traditional Chinese ^⇧⌘C
convert to base64 string and Replace
Date Description
Error Message Test Service
show base64 string
Document Code
Make New Sticky Note ⇧⌘Y
Open
Show in Finder
Show Info in Finder
デフォルトクリッピング に追加 (OmniOutliner)
デフォルトクリッピング に追加 (OmniOutliner) ⌘>
Searching
Look Up in Dictionary
Search With Bing ⇧⌘L
Messaging
New Email With Selection
Services Preferences...
Running BASE64 : BASE64
<key>NSRequiredContext</key>
<dict/>
<key>NSSendTypes</k
<array>
Input:

以下の3つは何も変わらない

ファイル一覧

クラス一覧

オブジェクトモデル

追加された物

B6AppDelegateにメソッド

```
@objc func insertDateServiceName(_ pasteboard: NSPasteboard,
                                 userData: String?,
                                 error: NSErrorPointer)
{
    // 現在の日時を出力
    let theReturnString = Date().description

    pasteboard.clearContents()
    pasteboard.setString(theReturnString as String, forType: .string)
}
```

plsitにNSServices

```
<dict>
    <key>NSMenuItem</key>
    <dict>
        <key>default</key>
        <string>Date Description</string>
    </dict>
    <key>NSMessage</key>
    <string>insertDateServiceName</string>
    <key>NSPortName</key>
    <string>com.mindto01s.mac.BASE64</string>
    <key>NSRequiredContext</key>
    <dict/>
    <key>NSReturnTypes</key>
    <array>
        <string>NSStringPboardType</string>
    </array>
</dict>
```

メニューに表示される

メソッド名

アプリのBundleID

必要な条件

返すデータのUTIのはずだが、
PboardTypeでないと動作しない

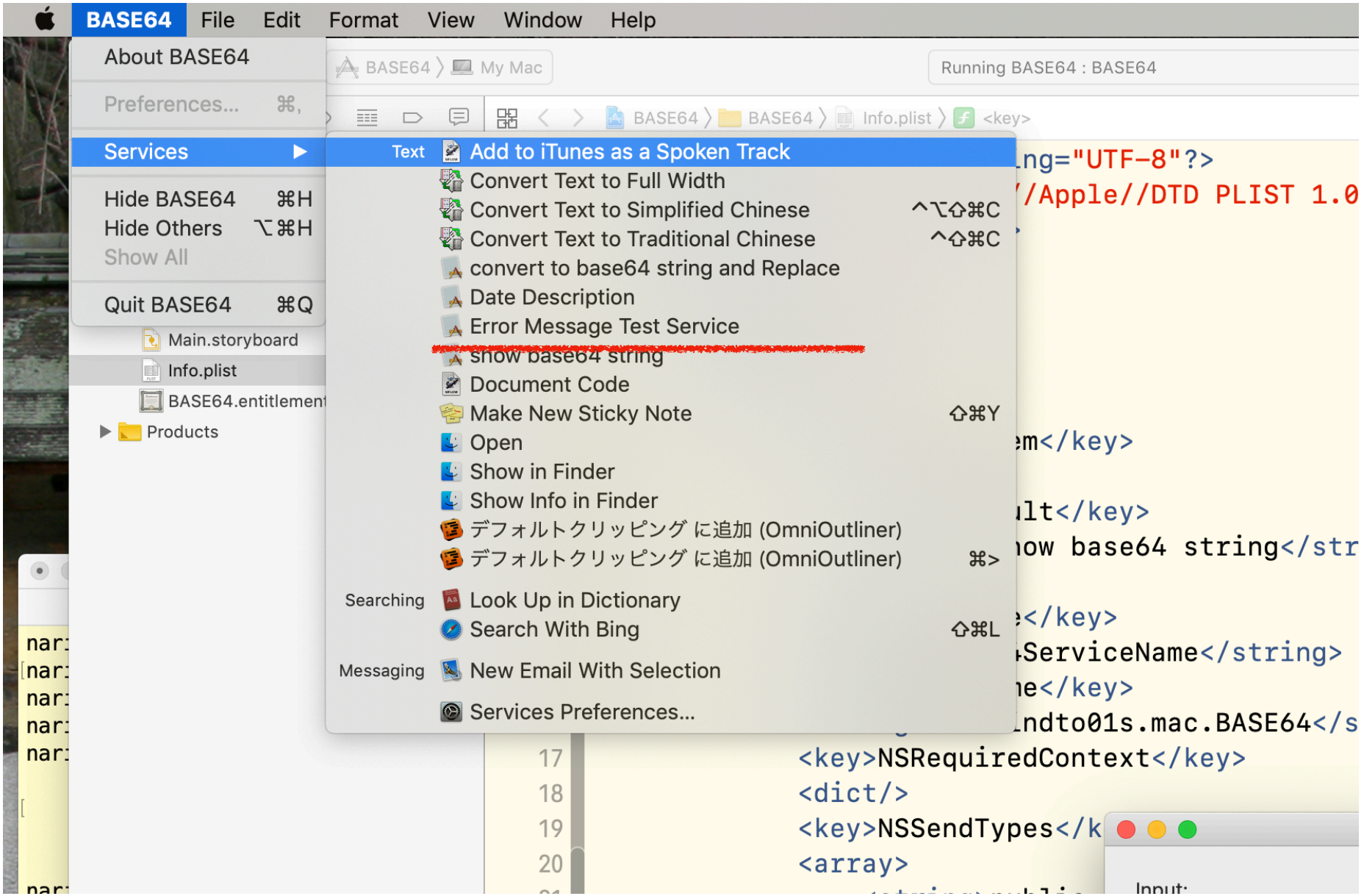
注意事項：

サービスを提供するアプリが前面にこない

例4

- エラー処理、これはよくわからなかった。
- 参考にはなりません。

例4:外見


BASE64
File
Edit
Format
View
Window
Help
About BASE64
Preferences... ⌘,
Services
Hide BASE64 ⌘H
Hide Others ⌥⌘H
Show All
Quit BASE64 ⌘Q
Main.storyboard
Info.plist
Products
Text
Add to iTunes as a Spoken Track
Convert Text to Full Width
Convert Text to Simplified Chinese ^⌥⇧⌘C
Convert Text to Traditional Chinese ^⇧⌘C
convert to base64 string and Replace
Date Description
Error Message Test Service
show base64 string
Document Code
Make New Sticky Note ⇧⌘Y
Open
Show in Finder
Show Info in Finder
デフォルトクリッピング に追加 (OmniOutliner)
デフォルトクリッピング に追加 (OmniOutliner) ⌘>
Searching
Look Up in Dictionary
Search With Bing ⇧⌘L
Messaging
New Email With Selection
Services Preferences...
Running BASE64 : BASE64
<key>NSRequiredContext</key>
<dict/>
<key>NSSendTypes</k
<array>
Input:

以下の3つは何も変わらない

ファイル一覧

クラス一覧

オブジェクトモデル

追加された物

B6AppDelegateにメソッド

```
@objc func errorServiceName(_ pasteboard: NSPasteboard,
                            userData: String?,
                            error: NSErrorPointer)
{
    error?.pointee = NSError(domain: "Error Message", code:-1)
}
```

plsitにNSServices

```
<dict>
	<key>NSMenuItem</key>
	<dict>
		<key>default</key>
		<string>Error Message Test Service</string>
	</dict>
	<key>NSMessage</key>
	<string>errorServiceName</string>
	<key>NSPortName</key>
	<string>com.mindto01s.mac.BASE64</string>
	<key>NSRequiredContext</key>
	<dict/>
	<key>NSReturnTypes</key>
	<array>
		<string>NSStringPboardType</string>
	</array>
</dict>
```

メニューに表示される

メソッド名

アプリのBundleID

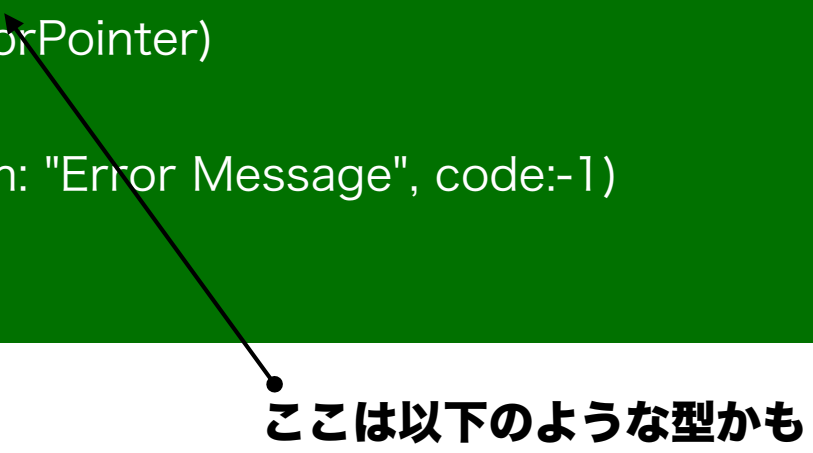
必要な条件

返すデータのUTIのはずだが、
PboardTypeでないと動作しない

注意事項：

サービスを提供するアプリが前面にこない

```
@objc func errorServiceName(_ pasteboard: NSPasteboard,
                            userData: String?,
                            error: NSErrorPointer)
{
    error?.pointee = NSError(domain: "Error Message", code:-1)
}
```



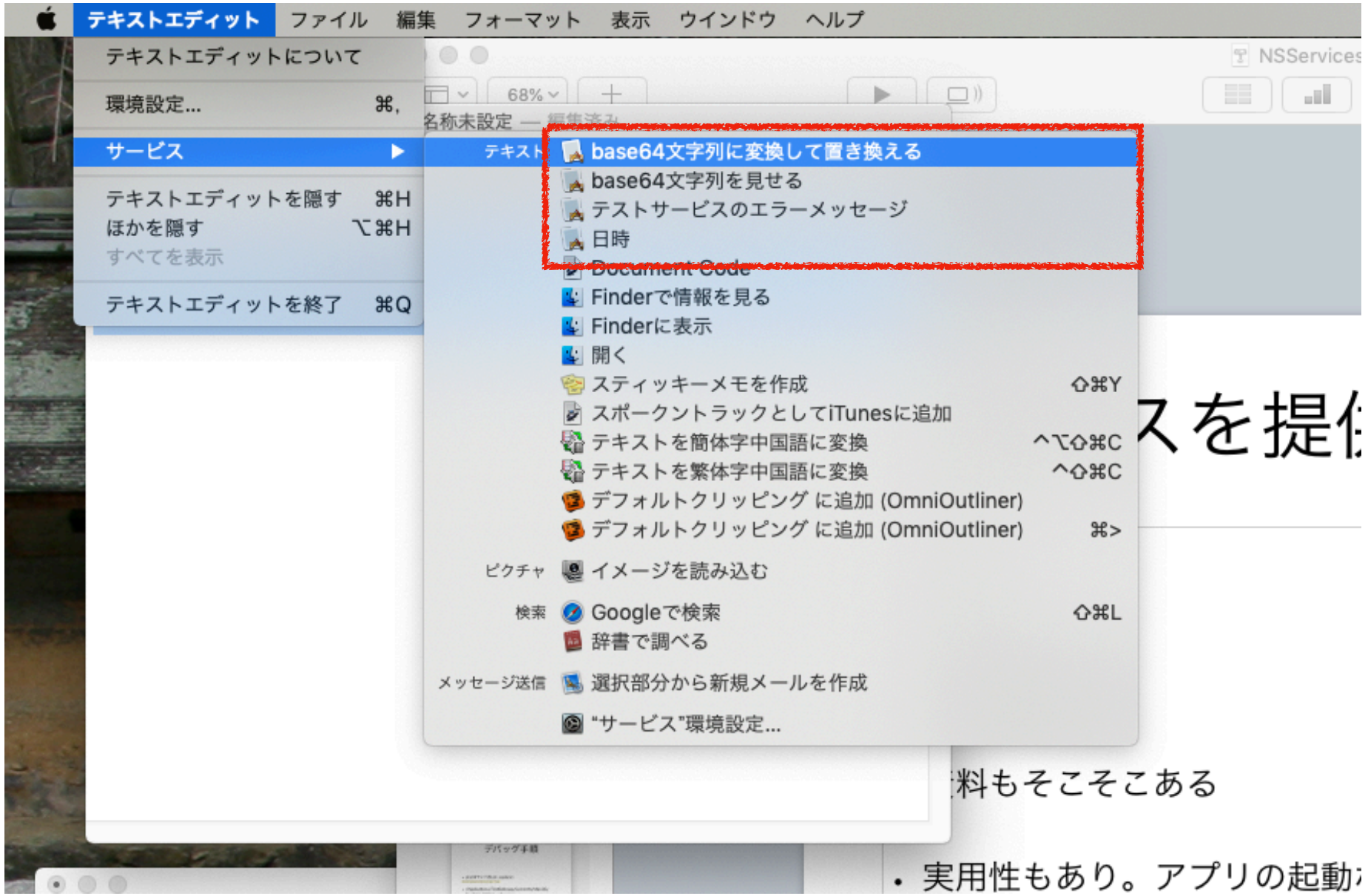
`AutoreleasingUnsafeMutablePointer<NSString?>?`

そして、`error?.pointee = “エラーメッセージ” as NSString`

例5

- メニューのローカライズ

例5:外見


テキストエディット
ファイル
編集
フォーマット
表示
ウインドウ
ヘルプ
テキストエディットについて
環境設定... ⌘,
サービス
テキストエディットを隠す ⌘H
ほかを隠す ⌥⌘H
すべてを表示
テキストエディットを終了 ⌘Q
テキスト
base64文字列に変換して置き換える
base64文字列を見せる
テストサービスのエラーメッセージ
日時
Finderで情報を見る
Finderに表示
開く
スティッキーメモを作成 ⇧⌘Y
スポークントラックとしてiTunesに追加
テキストを簡体字中国語に変換 ^⌥⇧⌘C
テキストを繁体字中国語に変換 ^⇧⌘C
デフォルトクリッピング に追加 (OmniOutliner)
デフォルトクリッピング に追加 (OmniOutliner) ⌘>
ピクチャ
イメージを読み込む
検索
Googleで検索 ⇧⌘L
辞書で調べる
メッセージ送信
選択部分から新規メールを作成
"サービス"環境設定...
NSServices
料もそこそこある
• 実用性もあり。アプリの起動

例5:ファイル一覧

```
├── BASE64
│   ├── Assets.xcassets
│   ├── B6AppDelegate.swift
│   ├── B6Model.swift
│   ├── B6ViewController.swift
│   ├── BASE64.entitlements
│   ├── Base.lproj
│   │   └── Main.storyboard
│   ├── Info.plist
│   ├── en.lproj
│   │   └── ServicesMenu.strings
│   └── ja.lproj
│       └── ServicesMenu.strings
└── BASE64.xcodeproj
```

追加された物

en.lprojに
ServicesMenu.stringsを追加

```
"show base64 string" = "show base64 string";

"convert to base64 string and Replace" = "convert to base64 string and Replace";

"Date Description" = "Date Description";

"Error Message Test Service" = "Error Message Test Service";
```

ja.lprojに ServicesMenu.stringsを追加

```
"show base64 string" = "base64文字列を見せる";

"convert to base64 string and Replace" = "base64文字列に変換して置き換える";

"Date Description" = "日時";

"Error Message Test Service" = "テストサービスのエラーメッセージ";
```

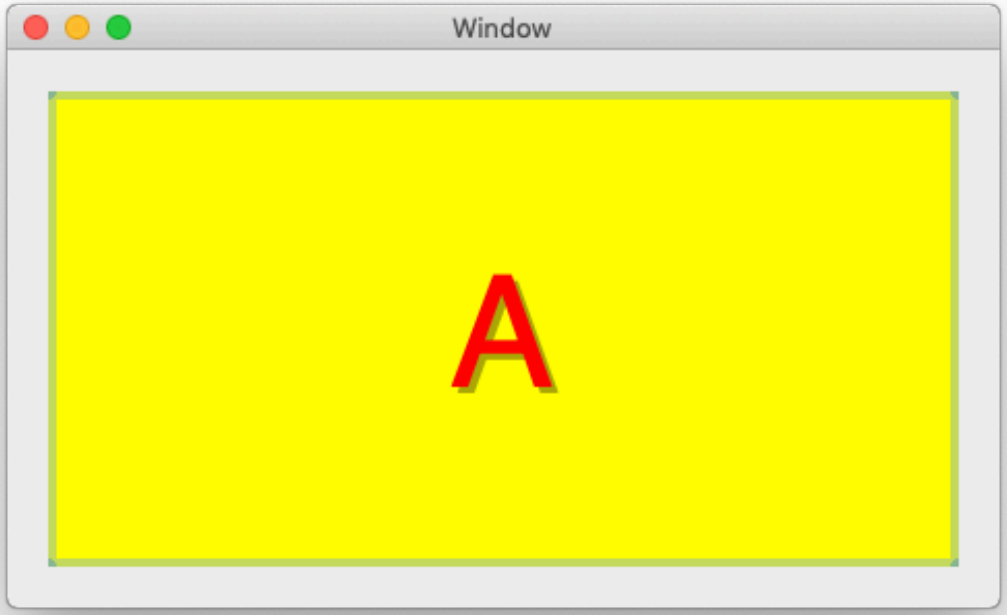
注意事項：

info.plistの中の項目のローカライズは、
他のリソースのローカライズと
少しだけルールが違うので気をつける

サービスを利用するアプリ

- NSTextViewはデフォルトで対応
- 独自のViewにサービスを利用する機能を追加する
- ヒレガス本のTypingTutorをベースに作る

外見



タイプすると1文字だけ表示

Copy&Paste出来る

詳しい解説は、
ヒレガス本を参照する事

実装手順

- NSAppのregisterServicesMenuSendTypes(, returnTypes:)を呼び出して、使用する型を登録する
- NSResponderの validRequestor(sendType:,returnType:) -> Any?をoverrideして、どの型の組み合わせに対応できるかを知らせる。
- NSServicesMenuRequestorのwriteSelectionとreadSelectionで、pasteBoardのデータを操作する。

registerServicesMenuSendTypes

```
override func awakeFromNib()
{
    // 呼び出し場所は適当でない。
    //nib経由以外で作成される場合には、別の方法で呼び出すこと。
    NSApp.registerServicesMenuSendTypes([.string], returnTypes: [.string])
}
```

何度でも呼び出しても良い。最低1回は呼び出される必要がある。

validRequestor(sendType:, returnType:)

```
override func validRequestor(forSendType sendType: NSPasteboard.PasteboardType?,
                             returnType: NSPasteboard.PasteboardType?) -> Any?
{
    // 編集可能か、現在選択されているかで、サービスの利用の可否を判定する。
    // このBigLetterViewでは常に選択されていて編集可能なので、.stringならば全部OKになっています。
    // なお、どちらもnilの場合は考えなくて良い

    if sendType == .string && returnType == nil
    {
        return self
    }

    if sendType == nil && returnType == .string
    {
        return self
    }

    if sendType == .string && returnType == .string
    {
        return self
    }

    return super.validRequestor(forSendType: sendType, returnType: returnType)
}
```

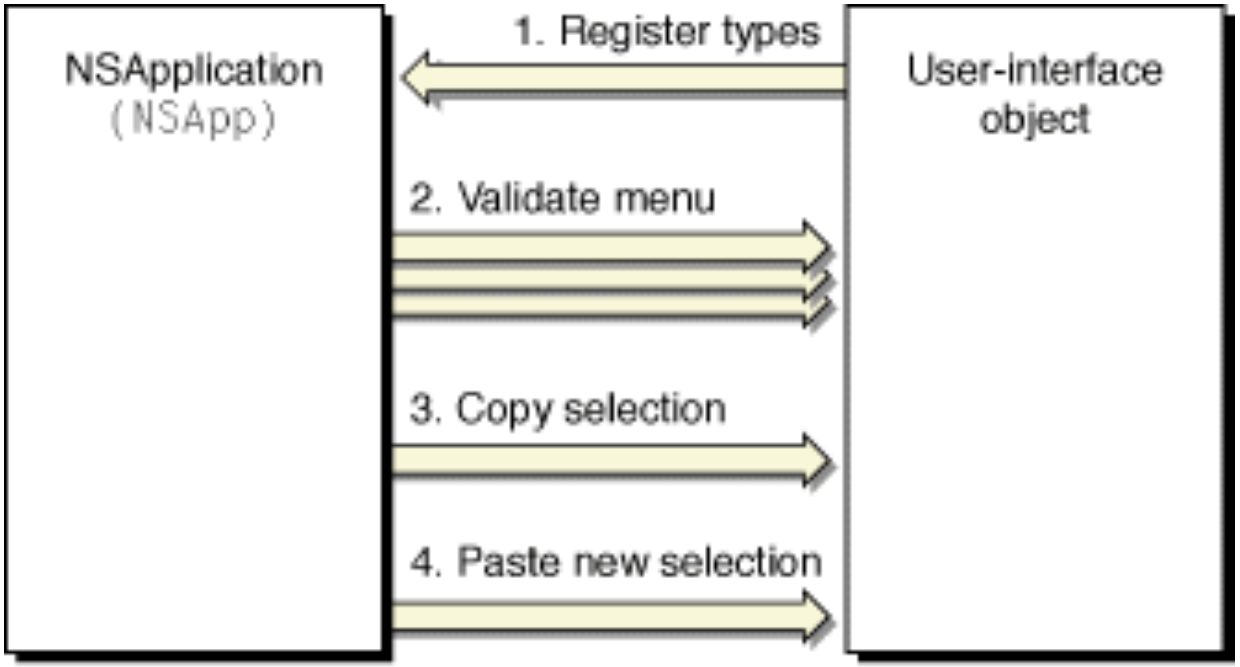
NSServicesMenuRequestor

```
@objc func writeSelection(to pboard: NSPasteboard,
                                types: [NSPasteboard.PasteboardType]) -> Bool
{
    // なぜか、System側は、”.string"でなくて、"NSStringPboardType"を返す
    // info.plistが関係するかもしれない

    if types.contains(.string)
        || types.contains(
          NSPasteboard.PasteboardType(rawValue: "NSStringPboardType"))
    {
        self.writeToPasteboard(pboard)
        return true
    }

    return false
}
```

NSServicesMenuRequestor

```
@objc func readSelection(from pboard: NSPasteboard) -> Bool
{
    return self.readFromPasteboard(pboard)
}
```

こんな順序で呼び出されます。



注意事項：

サービスを受けられる、独自のViewを作るメリットはほとんどない。

サービスメニューをコードから呼び出す

- 実用性は皆無。
- 時間があったら、解説する。

例5

- 使用可能なサービスの列挙と実行
  - いきなり難しくなってます。