

Xcode 介面建構器

這是一個以 XIB (Xcode Interface Builder) 格式編寫的 XML 文件，用於定義 macOS 應用程式的使用者介面，該應用程式以 Cocoa 編寫。具體來說，這個 XIB 文件描述了一個名為 “Reveal-In-GitHub” 的應用程式的設定視窗。讓我逐步為你解釋：

概述

這個文件定義了一個標題為 “Reveal-In-GitHub Settings” 的視窗，包含各種使用者介面元素，如按鈕、文本欄位和自訂視圖。它是使用 Interface Builder (Xcode 的一部分) 設計的，並使用 Auto Layout 來動態定位元素。這個視窗由一個名為 `RIGSettingWindowController` 的自訂類別控制。

主要部分

1. 標頭資訊

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<document type="com.apple.InterfaceBuilder3.Cocoa.XIB" version="3.0" toolsVersion="9059" systemVersion="14F27" useAutolayout="YES" />
```

- 這指定了這是一個用於 macOS Cocoa 應用程式的 XIB 文件，使用 XIB 格式的版本 3.0。
- `toolsVersion="9059"` 和 `systemVersion="14F27"` 指示了創建這個文件時所使用的 Xcode 和 macOS 版本。
- `useAutolayout="YES"` 意味著佈局會根據視窗大小或螢幕解析度動態調整。

2. 依賴

```
<dependencies>
  <plugin identifier="com.apple.InterfaceBuilder.CocoaPlugin" version="9059" />
</dependencies>
```

- 這參考了 Interface Builder 的 Cocoa 插件，該插件提供了 macOS 使用者介面元素的框架。

3. 物件 這部分定義了介面中的所有物件，包括視窗、視圖、按鈕及其連接。

- **文件擁有者 (RIGSettingWindowController)**

```
<customObject id="-2" userLabel="File's Owner" customClass="RIGSettingWindowController">
  <connections>
    <outlet property="configsView" destination="IKd-Ev-B9V" id="z2z-MF-G88"/>
    <outlet property="mainView" destination="se5-gp-Tj0" id="pGn-EC-mzi"/>
    <outlet property="window" destination="F0z-JX-Cv5" id="gIp-Ho-8D9"/>
  </connections>
</customObject>
```

- 文件擁有者是管理這個視窗的控制器類別 (`RIGSettingWindowController`)。

- 它有到主視圖、配置視圖和視窗本身的出口（連接），將使用者介面元素連接到程式碼。

• 視窗

```
<window title="Reveal-In-GitHub Settings" ... id="F0z-JX-Cv5" userLabel="Settings">
  <windowStyleMask key="styleMask" titled="YES" closable="YES" miniaturizable="YES" resizable="YES" />
  <rect key="contentRect" x="527" y="176" width="651" height="497" />
  <rect key="screenRect" x="0.0" y="0.0" width="1440" height="877" />
```

- 定義了一個具有標題、關閉/最小化/調整大小按鈕和大小為 651x497 像素的視窗。
- contentRect 指定了它在螢幕上的位置和大小，而 screenRect 描述了完整的螢幕解析度（1440x877）。

• 內容視圖

```
<view key="contentView" id="se5-gp-Tj0">
  <rect key="frame" x="0.0" y="0.0" width="651" height="497" />
  <subviews>
```

- 主內容視圖填滿視窗並包含所有使用者介面元素（子視圖）。

4. 使用者介面元素（子視圖）這裡是內容視圖中的一些關鍵元素：

• 文本欄位

```
<textField ... id="G1C-Td-n9Y">
  <rect key="frame" x="18" y="425" width="187" height="17" />
  <textFieldCell ... title="Custom Menu Items" ...>
```

- 位於 (18, 425) 的標籤說 “Custom Menu Items”。
- 其他文本欄位包括 “Menu Title”、“⌘⇧⌘+”（Control+Shift+Command 快捷鍵）和 “URL Pattern”。

• 按鈕

```
<button ... id="EuN-9g-Vcg">
  <rect key="frame" x="14" y="13" width="137" height="32" />
  <buttonCell ... title="Save" ...>
  <connections>
    <action selector="saveButtonClicked:" target="-2" id="efd-QE-4aQ" />
  </connections>
```

- 位於 (14, 13) 的 “Save” 按鈕會觸發控制器中的 saveButtonClicked: 方法。
- 其他按鈕包括 “Clear Default Repos” 和 “Reset Default Menus”，每個都與特定的動作相連。

• 自訂視圖

```
<customView ... id="IKd-Ev-B9V" userLabel="Config View">
  <rect key="frame" x="20" y="54" width="611" height="330" />
```

- 這是一個自訂視圖的佔位符（可能會以程式設計的方式填充），佔據了視窗的大部分。

5. 連接

- 按鈕連接到 RIGSettingWindowController 的方法（例如 saveButtonClicked:、clearButtonClicked:），使使用者互動能觸發程式碼。

這做了什麼？

這個 XIB 文件為一個名為 “Reveal-In-GitHub” 的 macOS 應用程式創建了一個設定視窗，這可能是一個工具，用於將 GitHub 功能整合到另一個應用程式中。這個視窗讓使用者能夠：

- 自訂選單項目（帶有標題和 URL 模式）。
- 保存他們的設定。
- 清除預設存儲庫或將選單重置為預設值。

佈局是固定的 (`fixedFrame="YES"`)，而不是完全利用 Auto Layout 限制，這意味著位置是以像素硬編碼的。這可能使它在不同螢幕大小下不太靈活，但設計起來更簡單。

為什麼使用 XIB？

XIB 文件允許開發者在 Xcode 的 Interface Builder 中視覺設計介面，並通過出口和動作將其連接到程式碼。這個文件將使用者介面（按鈕、文本等）與 `RIGSettingWindowController` 類別連接起來，該類別處理邏輯。