

# 謙言謙語

程式設計、資料庫、Linux的學習筆記。

2013年1月27日星期日

## Windows 版本控制系統 Git ( msysGit, TortoiseGit )

何謂版本控制？

簡單來說就是可以記錄開發者不同時間點所修改的程式碼，可以讓我們比較不同時期版本的程式碼，方便我們 debug，特別是在多人維護同一個專案的時候，可以了解別人改了甚麼東西，有問題也可以隨時回溯至之前的版本找出問題。

軟體開發者每天都在修改同一個或多個專案的程式碼，在做備份的時候總是要壓縮專案檔或複製到隨身碟，雖然簡單的幾個步驟，但是每天都這樣做也很累，而且不小心忘記做備份的動作，或者想要找回之前版本內某段修改過的程式碼，這些都是非常麻煩的事情，透過版本控制系統，我們可以徹底解決這種困擾。

目前常見的版本控制系統有兩種，一種是集中式的系統(e.g. SVN, Subversion)，另一種是分散式的管理(e.g. Git)，而我們這裡要介紹的是 Git。

Git 這套系統和早期 SVN 最大的不同，在於 Git 運作上幾乎都在本機上執行，採用分散式的方式，每個開發者都可以獨立作業，雖然占用的空間較多，但是使用上的彈性更大，Linux 的核心也是使用此套系統來做版本控制的喔！

目前在 Windows 底下使用 Git 所需要的軟體：

msysGit

<http://msysgit.github.com/>

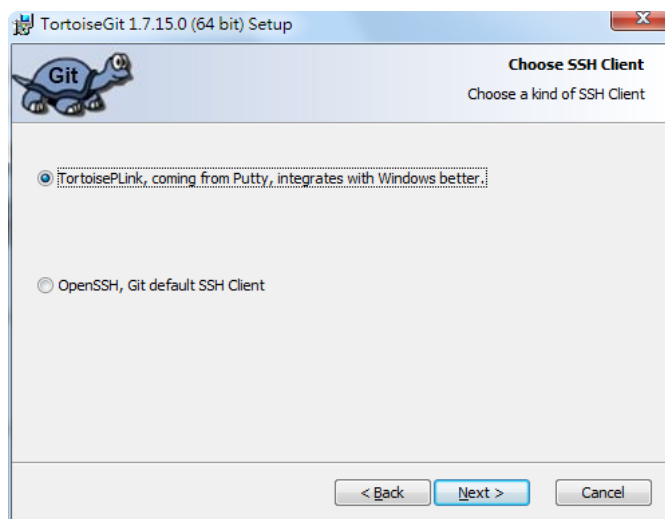
TortoiseGit

<http://code.google.com/p/tortoisegit/>

msysGit 是一定要安裝的主程式，而 TortoiseGit 只是前端的 GUI，msysGit 本身也有附上 GUI，但是還是建議安裝 TortoiseGit 這套介面更好用的工具。

我們先以比較簡單介紹如何操作本機上的版本控制：

一般來說會先安裝 msysGit 再安裝 TortoiseGit，但是我們這裡選擇先安裝 TortoiseGit，軟體安裝過程就是一直下一步而已，這裡就不再詳細敘述，只有一個地方要注意一下：



這裡是詢問要使用的SSH客戶端，預設是使用 TortoisePLink，和 Windows 的相容性較好，而且可以直接在視窗模式下設定SSH私鑰，不用到文字模式下設定，建議選擇此選項。

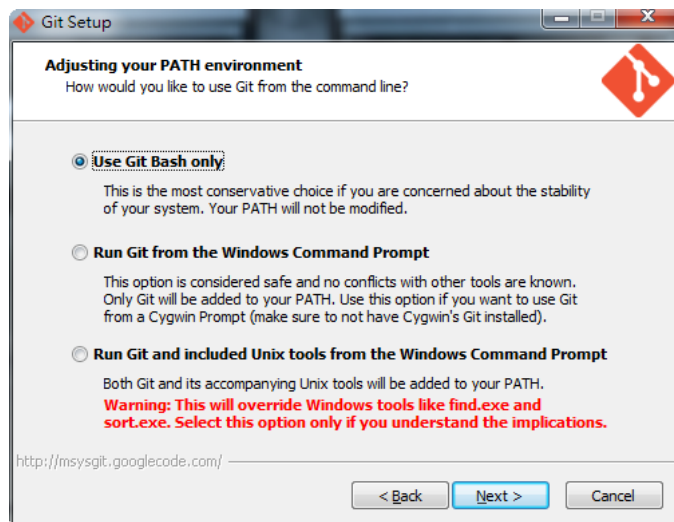
然後再來是安裝 msysGit，要注意幾個重要設定地方：

文章分類

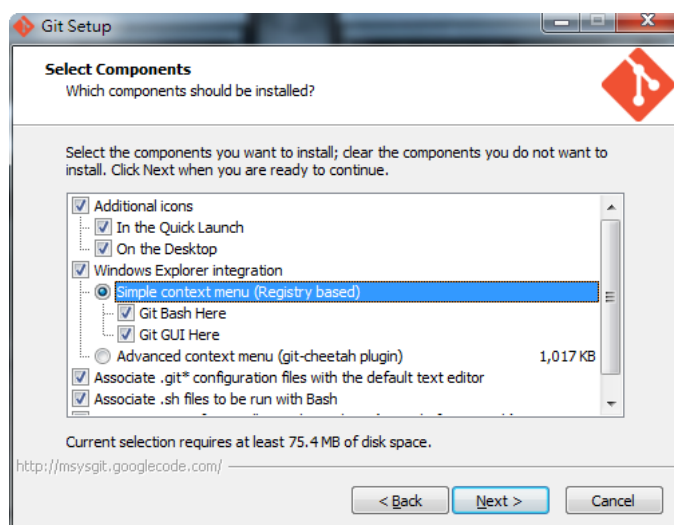
標籤分類

 [電腦](15/32)

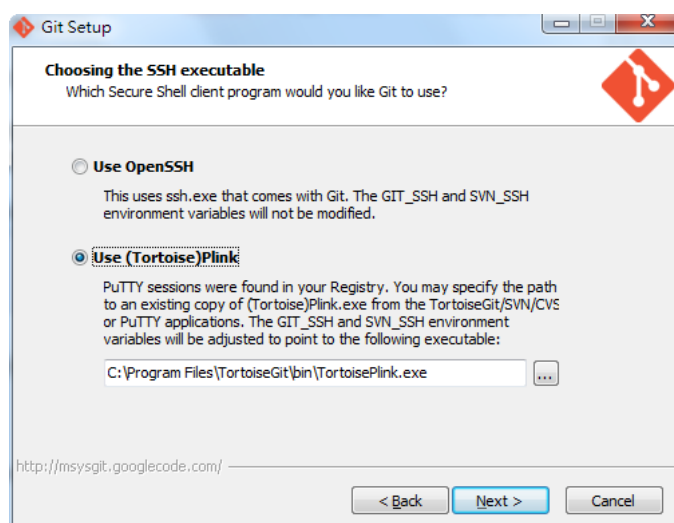
搜尋



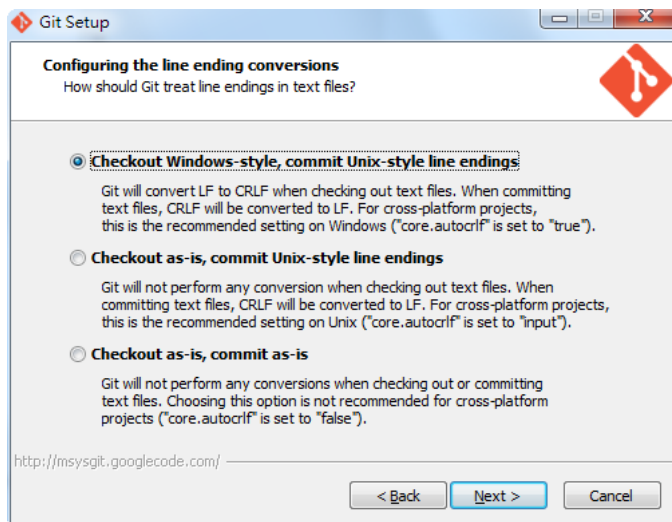
此處設定 Use Git Bash only，不會去修改到 PATH 環境變數，預設值也是最穩定的方式。



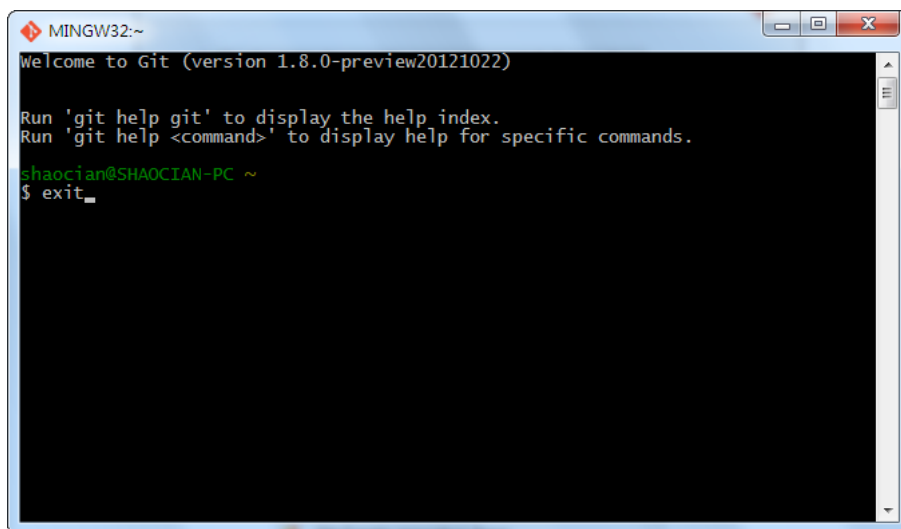
Windows 右鍵選單的整合，選擇 Simple context menu，底下兩個可以打勾，雖然此篇文章並不會用到，但是日後對於想要更進階了解 Git 的人可以先打勾，方便實務上的操作。



選擇 Use (Tortoise)Plink，如果有安裝 TortoisePlink 這裡就會顯示路徑。

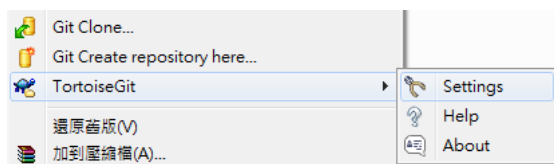


設定換行字元的轉換方式，Windows 環境建議選擇第一種（請參考原文提示）。



點選桌面的 Git Bash 捷徑圖示，即可看到文字的操作介面，輸入 `exit` 可以離開該程式

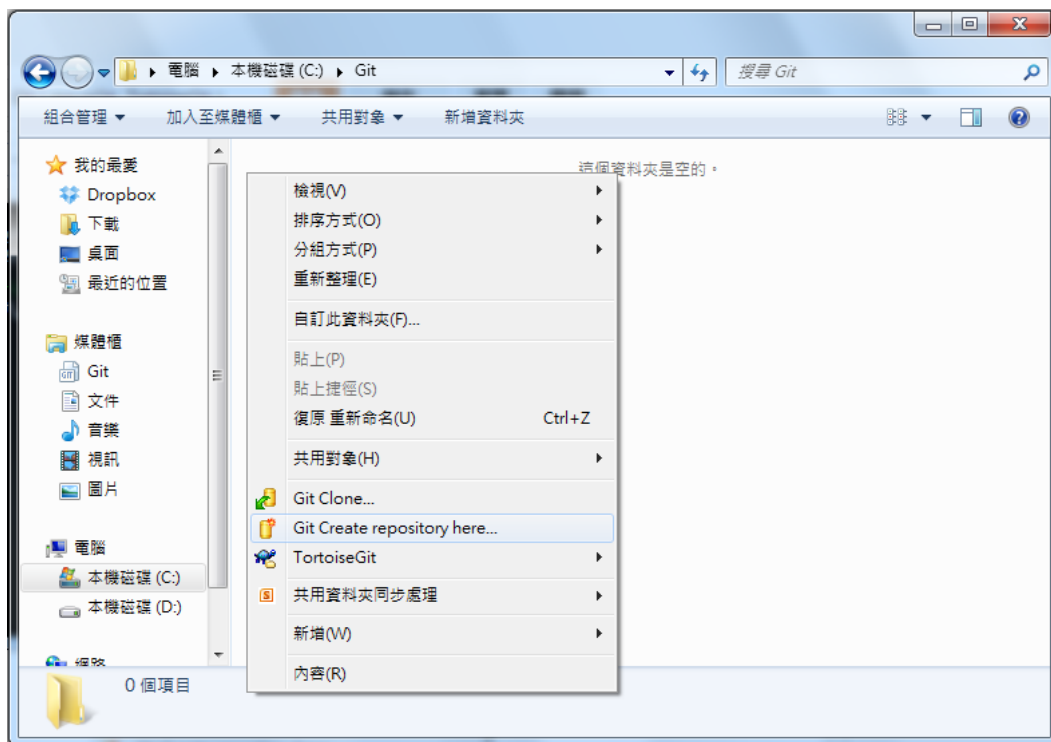
桌面上的 Git Bash 捷徑，或者在資料夾點選右鍵選單中的 Git GUI Here 和 Git Bash Here，都是 msysGit 安裝後提供的功能，這裡都暫時不介紹，而以使用介面比較好的 TortoiseGit 來替代：



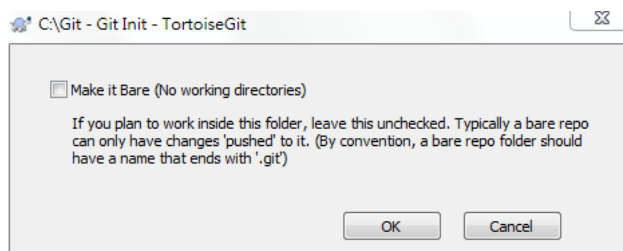
TortoiseGit 資料夾右鍵選單，由於還沒有設定 Repo，因此 TortoiseGit 只有顯示三個選項。

Repo ( Repository )，一般稱之為程式庫或版本庫，簡單來說 Repo 就是一個具有版本控制功能的資料夾。

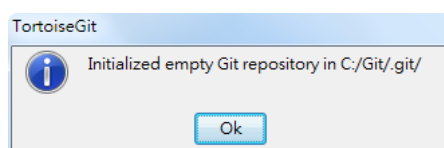
由於文字介面的操作適合比較進階的使用者，我們暫時先不詳細介紹，而是採用 TortoiseGit 提供的 GUI 方式來操作：



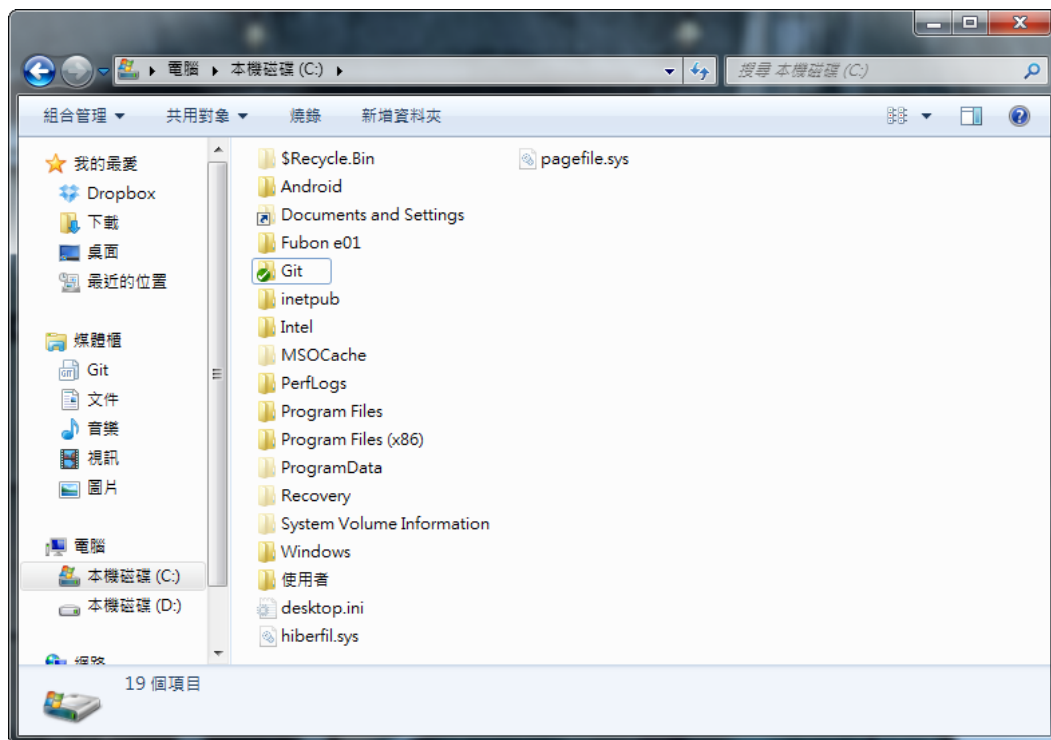
假設我們新增一個資料夾 C:\Git，進入該資料夾後點選右鍵選單中的 Git Create repository here...



不要勾選 Make it Bare，直接點選 OK



初始化資料夾，其實就是在底下產生一個 .git 的資料夾並產生相關設定檔

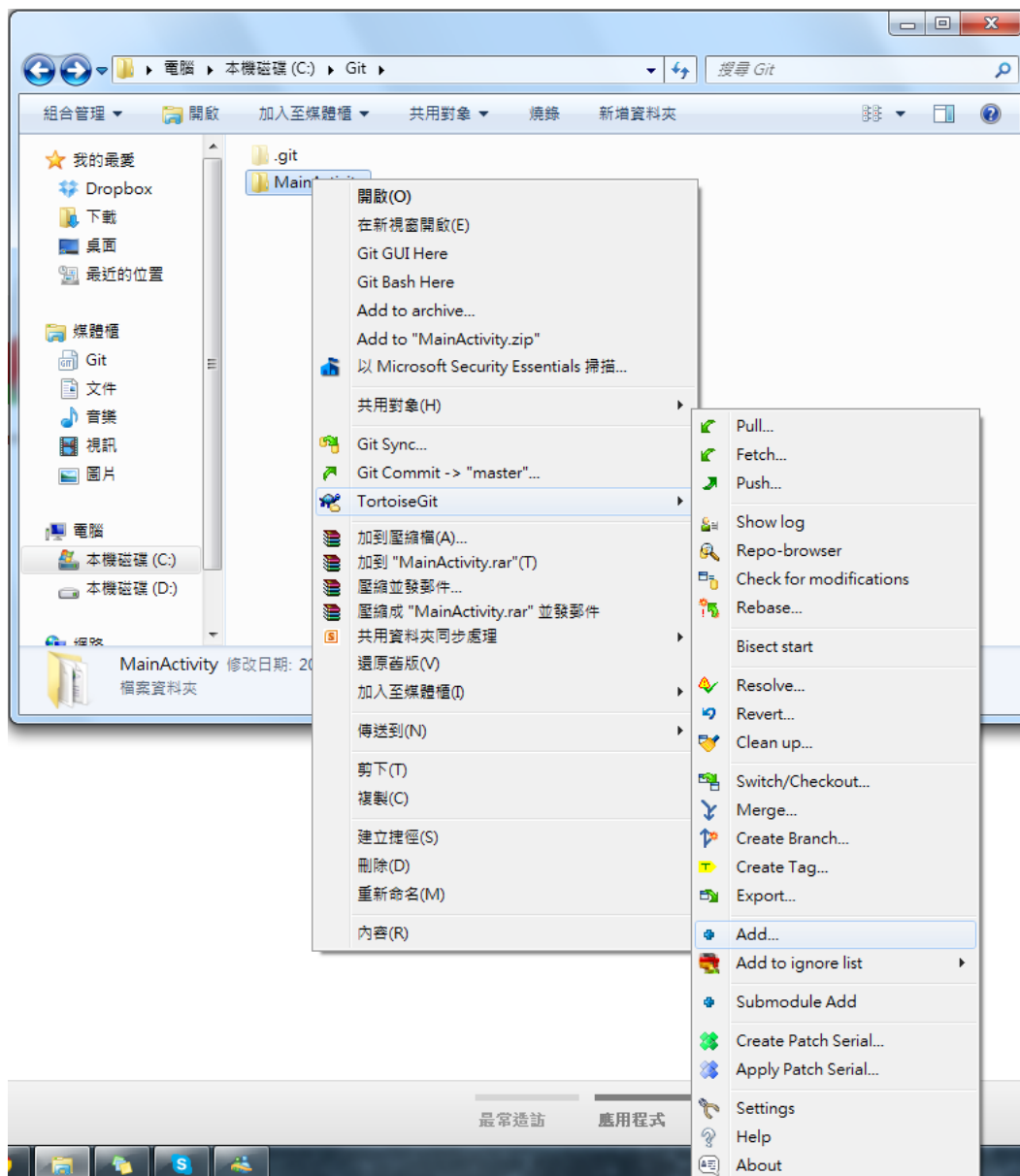


成功後可以看到 Git 資料夾本身的圖示改變（綠色勾勾），代表該目錄為版本控制目錄，點入內部會有一個新的 `.git` 的隱藏目錄，這就是在記錄著所有版本控制的相關訊息，如果把它刪除，此資料夾就會還原成一般的目錄。

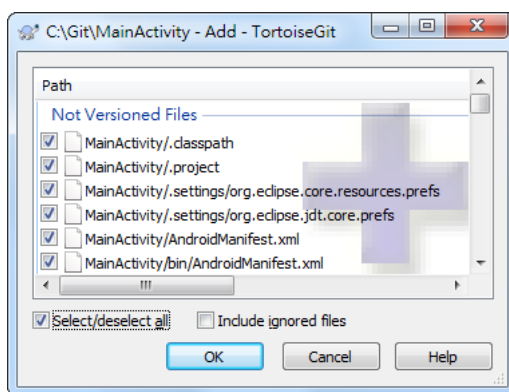
但是此時只有資料夾是具有版本控制的功能，接下來我們要將內部的檔案做版本控制：

例如我們使用 Eclipse 開發工具將 Git 資料夾設定為工作目錄(workspace)並且匯入自己的 Android 專案檔 (e.g. `C:\GitMainActivity`)

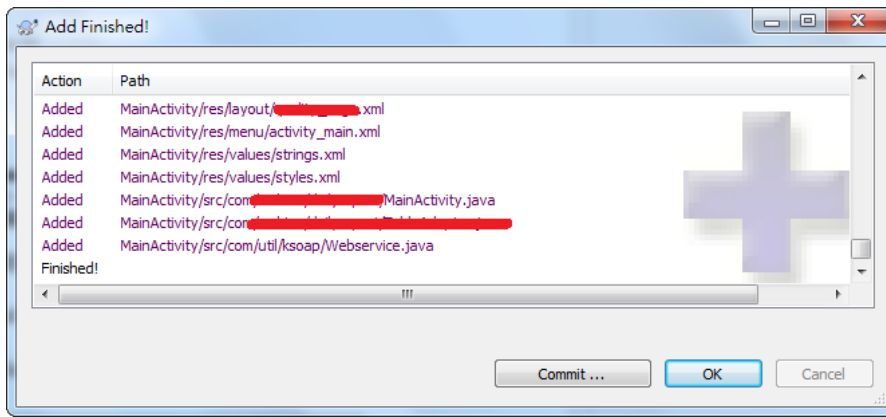
在該資料夾按右鍵點選 TortoiseGit | Add



在該資料夾按右鍵點選 TortoiseGit | Add



選擇想要監控的檔案，也可以直接勾選 Select/deselect all，然後點選 OK

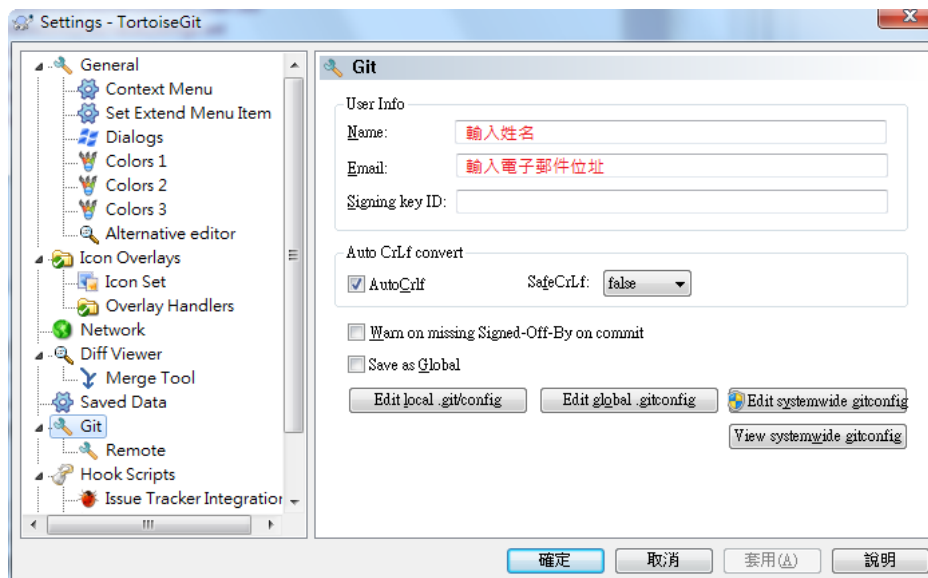


加入檔案完成後，您也可以在這裡直接做第一次的 Commit。

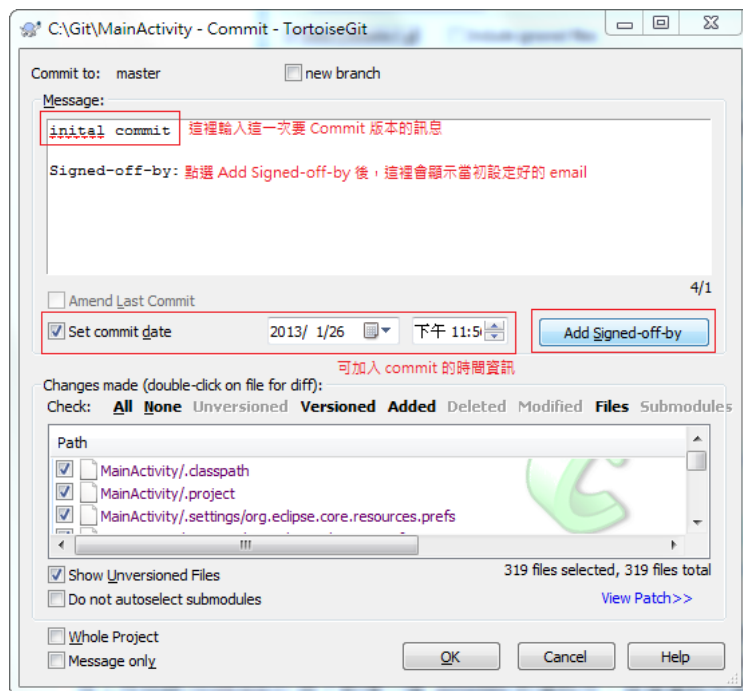
在 Git 系統中我們會使用到「Add」這一個指令，將檔案加入版本控制的監控（追蹤），雖然我們不一定要將該資料夾所有檔案都加入，例如 Eclipse 這套開發工具會在工作目錄 (workspace) 底下新增一個 .metadata 的資料夾，這個資料夾是存放一些開發工具的設定，和我們的開發軟體的版本之間並沒有關聯，因此可以不用加入版本控制系統的監控。

然而我們除了加入需要被追蹤的檔案之外，我們還要建立一個「初始版本」，這時候就要使用 commit 這個指令，先從英文上的意思來看，commit 有交付、承諾的意思，從功能上簡單來說，就是我們要將這些被追蹤的檔案做一個「版本」。

在需要 Commit 的資料夾上使用右鍵選單中的 Get Commit -> "master"

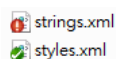


第一次使用會跳出設定畫面，請您先設定 Name 以及 email，可以自己在右鍵選單中先行設定 TortoiseGit | Settings。

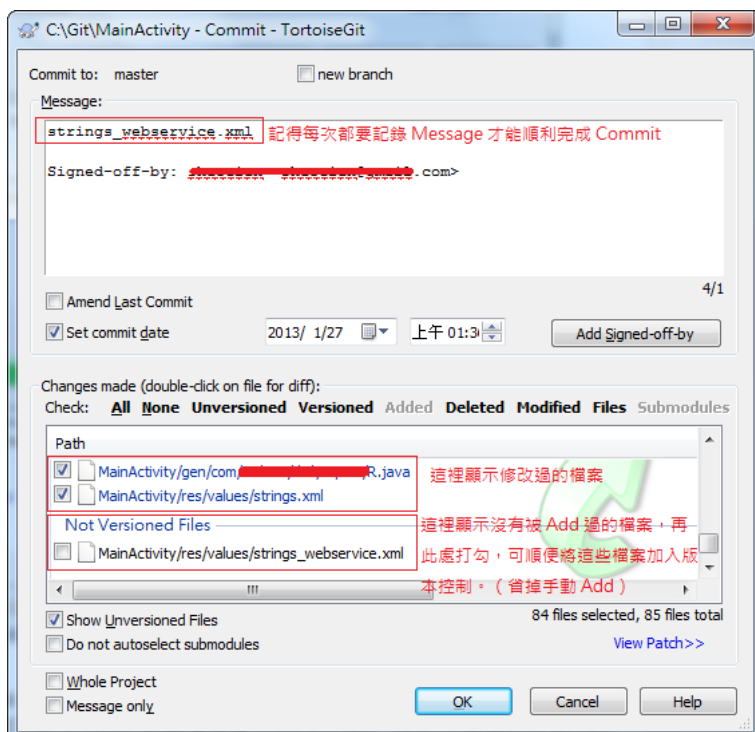


commit 設定畫面，輸入這一次 commit 的訊息(Message)，然後點選 Add Signed-off-by 簽署後點選OK即可進行 Commit。

第一個初始版本已經 Commit 成功，假設我們修改了一些檔案（例如 strings.xml 這個檔案），由於此檔案當初已經 Add 加入版本控制的追蹤檔案當中(tracked file)，所以Git偵測到有修改的檔案就會將其圖示顯示為紅色的驚嘆號（其每一個外層的資料夾也會同步顯示），提醒您有檔案經過修改了。

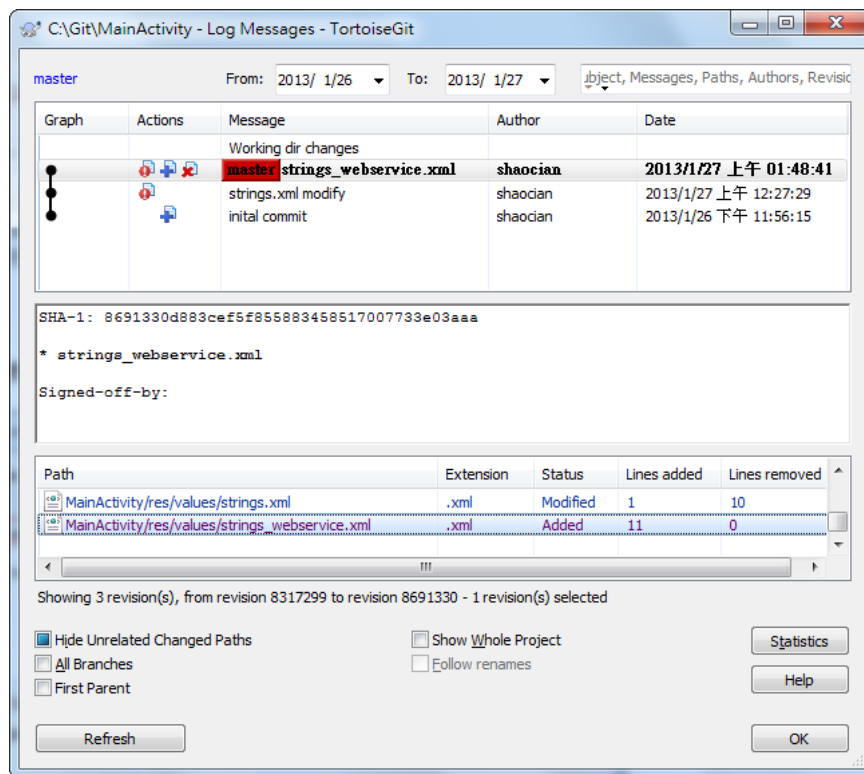


假設專案修改後，想要有一個新的版本紀錄，因此再做一次 Commit

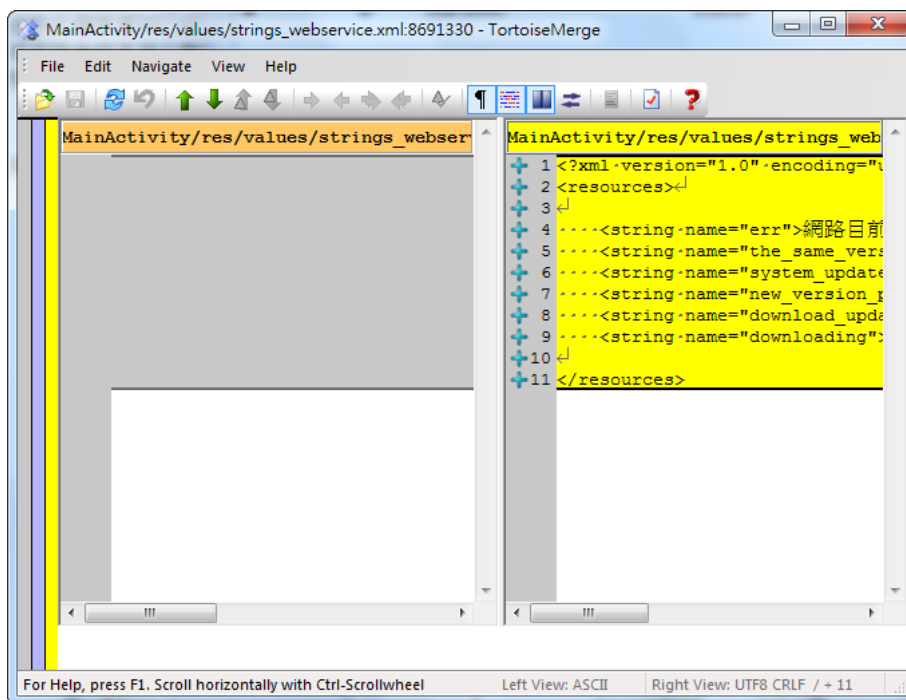


順利完成 Commit 後，我們可以利用 TortoiseGit 右鍵選單內有一個 Show log，可以檢查每一次 Commit 的紀錄。



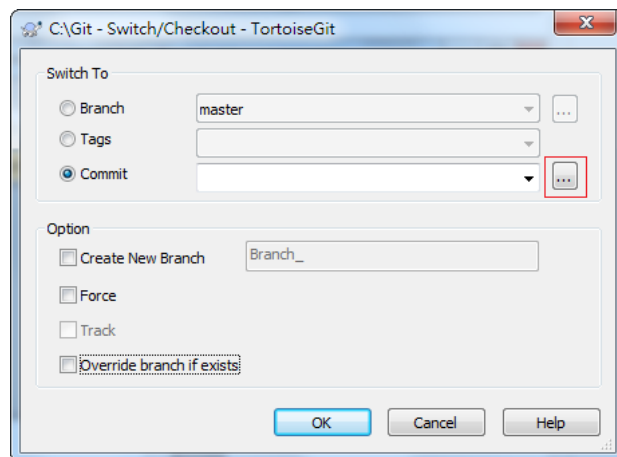


點選檔案還可以看修改了哪些東西

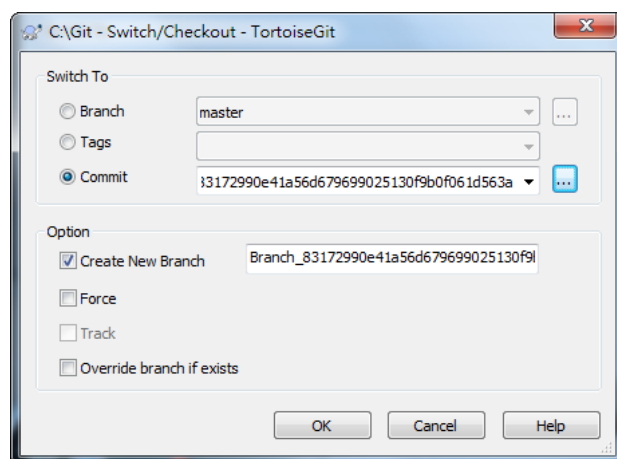


如果我們修改了專案後發現程式碼整個壞掉了卻找不出原因，想要直接還原到之前 Commit 的版本：

在資料夾右鍵點選 TortoiseGit | Switch/Checkout



選 **Commit**，然後點選右邊的圖示，會帶出如同 **Show log** 出現的畫面，然後點選想要還原的 **Commit** 版本後，就會自動帶出如以下的一連串字元



每一個 **Commit** 的版本都會經過SHA-1演算法加密，產生具有唯一性的識別碼，由於這一連串符號很難靠人去記憶，所以只能用選擇讓系統自行帶出。

要注意的是 **Option** 的部分，**Create New Branch** 勾選決定是否需要產生新的分支(**Branch**)，如果確定要還原到之前的版本，可是又沒有產生新的分支的情況下，那你選擇的 **Commit** 版本之後的所有的資料都會被刪掉（整個回溯到先前的版本），這點務必注意。

只要做完了 **Switch/Checkout** 的動作，記得在開發工具內做一次檔案更新（例如在 **Eclipse** 的專案上點選右鍵有一個 **Refresh**，該功能會重新掃描專案的檔案）。

p.s. 基本上如果您已經會切換 **Commit**，那切換分支(**Branch**)一定也不會有甚麼問題，一樣是在這個畫面上做切換。

到這裡為止，應該可以很清楚了解如何透過簡單的圖形使用者介面來操作本機，以及紀錄每一次專案修改後的版本、並做還原的動作。

張貼者：謙 於 上午3:06

標籤：[電腦][程式設計][開發工具]版本控制

## 4 則留言：

 **wade** 2013年4月30日 下午2:41

Thanks for this article , help me a lot

[回覆](#)

[回覆](#)



**謙** 2013年4月30日 下午8:12

Thanks for your reply, I'm glad I could help :)

[回覆](#)[卓三凱](#) 2013年10月14日 上午10:24

謝謝你 很清楚的流程講解~

[回覆](#)[獨羨西江水](#) 2014年2月6日 下午1:11

獲益良多~謝謝

[回覆](#)

輸入您的留言...

發表留言的身分：

schubertlee (G ▼)

登出

發佈

預覽

☐ 通知我[較新的文章](#)[首頁](#)[較舊的文章](#)訂閱：[張貼留言 \(Atom\)](#)

謙言謙語由謙製作，以創用CC 姓名標示-非商業性-禁止改作 3.0 Unported 授權條款釋出。  
此作品衍生自<http://shaocian.blogspot.tw/>。

由 [Blogger](#) 技術提供。