

Git Learning Lesson Two

Git Learning Lesson Two v0.1

© 2015 MStar Semiconductor, Inc. All rights reserved.

MStar Semiconductor makes no representations or warranties including, for example but not limited to, warranties of merchantability, fitness for a particular purpose, infringement of any intellectual property right or the accuracy or completeness of this document, and reserves the right to make changes without further notice to any products herein to improve reliability, function or design. No responsibility is assumed by MStar Semiconductor arising out of the application or user of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

MStar is a trademark of MStar Semiconductor, Inc. Other trademarks or names herein are only for identification purposes only and owned by their respective owners.

REVISION HISTORY

Revision No.	Description	Date
V0.1	<ul style="list-style-type: none">{Initial release}	{06/30/2015}
	<ul style="list-style-type: none">	

TABLE OF CONTENTS

REVISION HISTORY	i
TABLE OF CONTENTS.....	ii
LIST OF TABLES.....	iii
LIST OF FIGURES	iii
1. 前言.....	1
2. review 常用名詞介紹	2
2.1. Commit Message 區域.....	3
2.2. Project, branch, owner 資訊區域	3
2.3. Commit-Id & Change-Id.....	4
2.4. 修改檔案列表	4
2.5. History.....	4
3. git review 流程.....	5
3.1. Review Commit.....	5
3.1.1 git web 頁面中，reviewer 本身的 my changes 的 incoming review	5
3.1.2 上 code 者提供需 review 的 commit 的 URI	5
3.1.3 上 code 者提供需 review 的 commit-Id.....	6
3.2. Review modifications.....	6
3.3. +1/-1/+2/-2 評分.....	8
3.4. Submit code	9
4. Q&A.....	10
4.1. Q: Patch set 是什麼？跟 perforce 的 revision 有對應關係嗎？	10
4.2. Q:當 review 完成要 submit 時，發生 error 該怎麼處理呢？	10
4.3. Q:	11

LIST OF TABLES

找不到圖表目錄。

LIST OF FIGURES

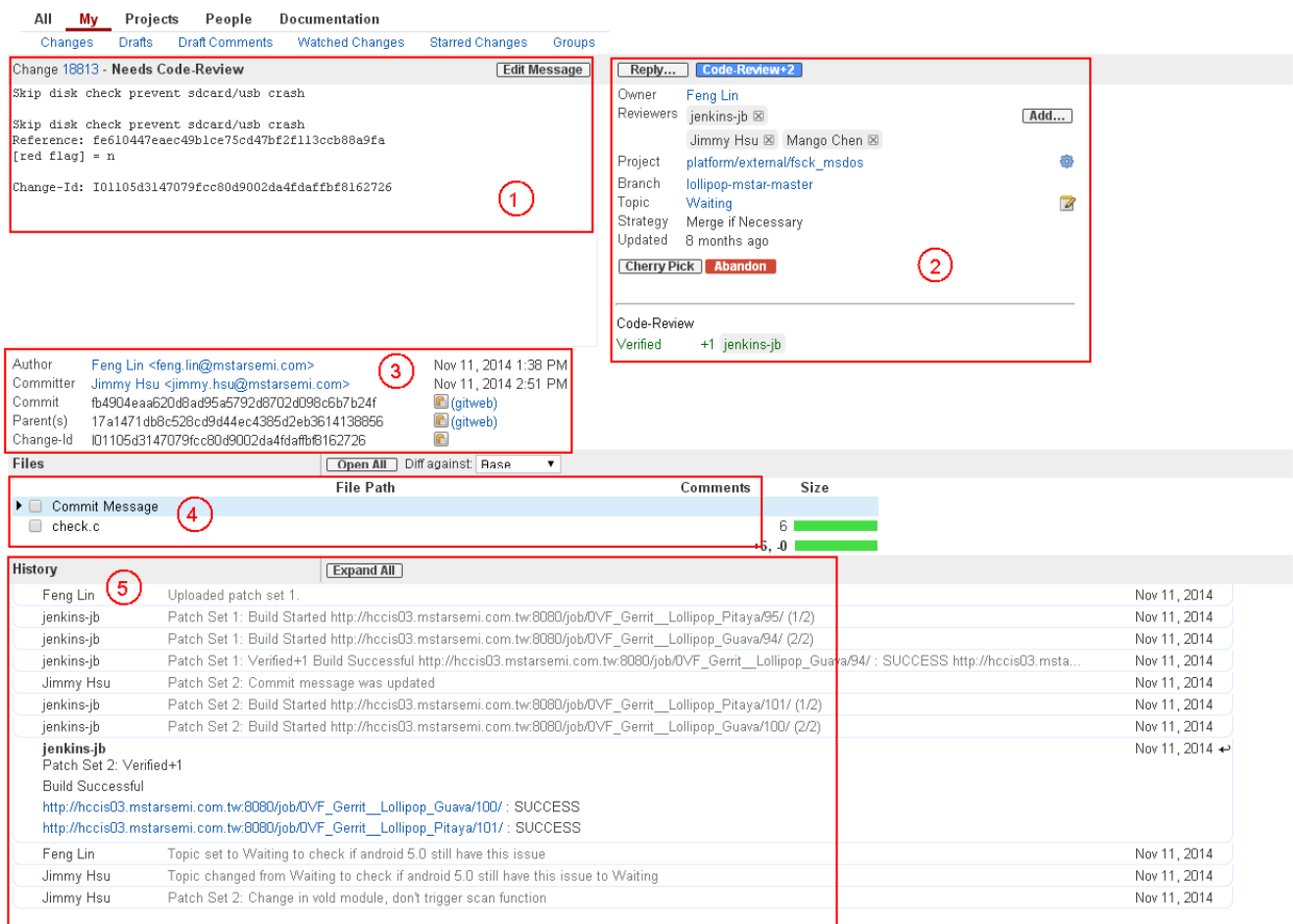
找不到圖表目錄。

1. 前言

因應 git 全面化推行，是故撰寫文件來簡介 git 的基本使用方式，希望能讓大家更輕鬆的從 perforce 環境切換到 git 上。本文件將簡單的介紹使用者 push code 以後，reviewer 端 review code 的流程。

2. REVIEW 常用名詞介紹

本章節針對 **commit** 頁面上常用 **item**，以及 **review** 時常用到的詞語做簡單的介紹。
下圖為 **commit** 頁面的縮圖，各小節將分別介紹圖中標示的區塊。



The screenshot shows the Git commit review interface with the following components:

- 1. Change Summary:** Displays the commit message "Skip disk check prevent sdcard/usb crash", the reference hash "fe610447eae49b1ce75cd47bf2f113ccb88a9fa", and the change ID "I01105d3147079fcc80d9002da4fdaffb8162726".
- 2. Review Metadata:** Shows the owner "Feng Lin", reviewers "jenkins-jb", "Jimmy Hsu", and "Mango Chen". It also displays the project "platform/external/fsck_msdos", branch "lollipop-mstar-master", and topic "Waiting".
- 3. Author/Committer Info:** Lists the author "Feng Lin" and committer "Jimmy Hsu" with their email addresses and commit timestamps.
- 4. File Changes:** Shows the file "check.c" with a size of 6 bytes.
- 5. History:** A list of actions performed on the commit, including patch uploads, build status updates, and topic changes.

2.1. Commit Message 區域

Change 18813 - Needs Code-Review
Edit Message

Skip disk check prevent sdcard/usb crash

Skip disk check prevent sdcard/usb crash
Reference: fe610447eaec49b1ce75cd47bf2f113ccb88a9fa
[red flag] = n

Change-Id: I01105d3147079fcc80d9002da4fdaffbf8162726

與使用者執行 `git commit` 時，輸入的資訊一樣，包含 `commit title`, `commit contents`, `red flag`, 等等。其中 `Change-Id` 項目為系統自動產生，不需使用者自行填入。

Review 的要點著重於是否符合 `commit message` 的寫法。

android commit message rule : <http://hcgit/htdocs/codestyle/Commit%20message.pdf>

2.2. Project, branch, owner 資訊區域

Reply...
Code-Review+2

Owner: Feng Lin
Reviewers: jenkins-jb [x] Add...
Jimmy Hsu [x] Mango Chen [x]
Project: platform/external/fsck_msdos [gear]
Branch: lollipop-mstar-master
Topic: Waiting [notepad]
Strategy: Merge if Necessary
Updated: 2 hours ago
Cherry Pick Abandon

Code-Review
Verified +1 jenkins-jb

這個區塊包含 Owner, Reviewer, Project, Branch, Topic, Code-Review 幾個部份。

Owner：上 code 的人，也就是執行 `git commit`. `git push` 的使用者。

Reviewer：此 project 的 reviewer，幫忙執行 code review 動作的人。也可以透過旁邊的 add 按鈕新增 reviewer。

Project：這筆 commit 要上的 code 是屬於哪一個 project。

Branch：這筆 commit 要上到哪一個 branch。通常都是上到開發線，透過 `release` 將 code merge 到主線去。除非此 project 沒有開發線，才是直接上到主線。

Topic：提供使用者做額外註記的地方，常見使用方式：`No Merge`(提醒 reviewer 暫時先不要 merge)、`Waiting`(等待確認是否真的需要這筆 commit 的 solution)

Code-Review：Reviewer review 完 code 以後的結果會顯示在這個區域。不管是 `+1/-1/+2/-2` 均會顯示。而加減分數的方法，則是透過畫面上的 `reply` 按鈕。在後續章節，將有更詳細的介紹。

Abandon：將此筆 commit 移出 review system 列表。如果 commit 已不需要，避免積太多 commit 在 reviewer 的 review pending list 中，建議將這筆 commit 做 abandon 的動作。

2.3. Commit-Id & Change-Id

Author	Feng Lin <feng.lin@mstarsemi.com>	Nov 11, 2014 1:38 PM
Committer	Jimmy Hsu <jimmy.hsu@mstarsemi.com>	Nov 11, 2014 2:51 PM
Commit	fb4904eaa620d8ad95a5792d8702d098c6b7b24f	 (gitweb)
Parent(s)	17a1471db8c528cd9d44ec4385d2eb3614138856	 (gitweb)
Change-Id	I01105d3147079fcc80d9002da4fdaafb162726	

這個區域重點為 Commit-Id 與 Change-Id 兩項。

Change-Id：是使用者第一次執行 `git commit` 的時候，自動產生的 hash number，針對這個 commit，這組 Change-Id 將是唯一的，不會隨著 `rebase`、`git commit --amend` 等動作而被更新。

Commit-Id：代表著每一次使用者 update code 的代號，在每次 `rebase`，或是 update patch set 以後，會重新產生一組新的 Commit-Id。

這邊要注意得是，當要提供 fixed commit id 的時候，記得要給出最後一個 patch set 的 commit id。也就是要給最後一次上 code 時的 commit id。

Ex: Commit A 一共修改了兩次，`rebase` 了三次，順序是：上 code, `rebase`, `rebase`, `update`, `rebase`, `merge` 所以一共會有五個 commit id，一個 change id。

這時候要填寫 mantis resolved CL 時，要填寫的是最後一次 `rebase` 以後產生的 commit id。跟 code 的人才能拿到正確的修改。

2.4. 修改檔案列表

Files		Open All	Diff against:	Base	
		File Path		Comments	Size
<input type="checkbox"/>	Commit Message				
<input checked="" type="checkbox"/>	check.c				6
				+6, -0	

這個區域列出了這筆 commit 修改到的檔案。

透過點擊檔案，可以開啓 git 的 diff 頁面進行 code review。後續章節將有更詳細的介紹。

2.5. History

History		Expand All
Feng Lin	Uploaded patch set 1.	Nov 11, 2014
jenkins-jb	Patch Set 1: Build Started http://hccis03.mstarsemi.com.tw:8080/job/DVF_Gerrit_Lollipop_Pitaya/95/ (1/2)	Nov 11, 2014
jenkins-jb	Patch Set 1: Build Started http://hccis03.mstarsemi.com.tw:8080/job/DVF_Gerrit_Lollipop_Guava/94/ (2/2)	Nov 11, 2014
jenkins-jb	Patch Set 1: Verified+1 Build Successful http://hccis03.mstarsemi.com.tw:8080/job/DVF_Gerrit_Lollipop_Guava/94/ : SUCCESS http://hccis03.msta...	Nov 11, 2014
Jimmy Hsu	Patch Set 2: Commit message was updated	Nov 11, 2014
jenkins-jb	Patch Set 2: Build Started http://hccis03.mstarsemi.com.tw:8080/job/DVF_Gerrit_Lollipop_Pitaya/101/ (1/2)	Nov 11, 2014
jenkins-jb	Patch Set 2: Build Started http://hccis03.mstarsemi.com.tw:8080/job/DVF_Gerrit_Lollipop_Guava/100/ (2/2)	Nov 11, 2014
jenkins-jb	Patch Set 2: Verified+1 Build Successful http://hccis03.mstarsemi.com.tw:8080/job/DVF_Gerrit_Lollipop_Guava/100/ : SUCCESS http://hccis03.msta...	Nov 11, 2014
Feng Lin	Topic set to Waiting to check if android 5.0 still have this issue	Nov 11, 2014
Jimmy Hsu	Topic changed from Waiting to check if android 5.0 still have this issue to Waiting	Nov 11, 2014
Jimmy Hsu	Patch Set 2: Change in vold module, don't trigger scan function	Nov 11, 2014

這個區域可以看到從產生 commit 到目前的各種歷程。

包含 patch set update、jenkins build result、reviewer 給予的 comments 等等。

3. GIT REVIEW 流程

以下簡單介紹 reviewer 執行 review 時的基本操作。

3.1. Review Commit

Reviewer 接收到 review commit 的來源大概有幾種。

3.1.1 git web 頁面中，reviewer 本身的 my changes 的 incoming review

All	My	Projects	People	Documentation
Changes	Drafts	Draft Comments	Watched Changes	Starred Changes
	Groups			

My Reviews

Subject	Status
Outgoing reviews	
★ Skip disk check prevent sdcard/usb crash	
★ Rearrange density by osdWidth/osdHeight.	
▶ ★ Backup boot time	
Incoming reviews	
★ Fixed Cannot change to right input source from LMM	
★ Add Input-Source-Lock APIs	
★ Enable program list sorting function	
★ Add DummySecureView to support dummy secure view	Merge Conflict
★ Add api to operate eThreeDVideoSelfAdaptiveLevel	Merge Conflict
★ Fixed cannot disable SRS	Merge Conflict
★ Refine api get/set AudioPrescale	Merge Conflict
★ Add dvbs blind scan(2/2)	Merge Conflict
★ [STR] Do not switch to storage when suspending.	Merge Conflict
★ Testinq	Merge Conflict

只要點擊 description 即可進入 commit 的詳細資料頁面。

3.1.2 上 code 者提供需 review 的 commit 的 URI

EX: <http://hcgit:8080/#/c/25819/>

只要將網址貼到 web browser 上，並進行 git account 登入，即可進入 commit 的詳細資料頁面。

3.1.3 上 code 者提供需 review 的 commit-Id

EX: db15c3bd352b373c94cb3202be7a2d041533e5e9

登入 git web server 後，將 commit id 貼在右上角搜尋處，按下 enter，即可進入 commit 的詳細資料頁面。

Feng Lin

3.2. Review modifications

如同 2.4 提到的，commit 頁面可以看到修改的 file list。

會列出這筆 commit 一共有修改到的檔案，以及此筆 commit 的 commit message。

值得注意的是如下圖顯示，可以選擇目前要 review 的修改，是要將最新修改與哪一個版本做比對。

預設將會是 Base 選項，也就是將最新一次修改，與未修改以前的 code 做比對。

如果下圖例子選擇了 2:79223a 那麼等下的 review diff 頁面將會顯示這次改動與上一次 patch set 2 的差異部份。

Files	Open All	Diff against	Base		Comments	Size
<input checked="" type="checkbox"/> Commit Message						
<input checked="" type="checkbox"/> core/java/android/content/pm/PackageParser.java						502
<input checked="" type="checkbox"/> core/java/android/content/res/AssetManager.java						9
<input type="checkbox"/> libs/androidfw/AssetManager.cpp						34
<input type="checkbox"/> services/core/java/com/android/server/pm/PackageManagerService.java						166
						+628, -83

而檔案前面的勾，則是表示這個檔案已經有點進去看過了。

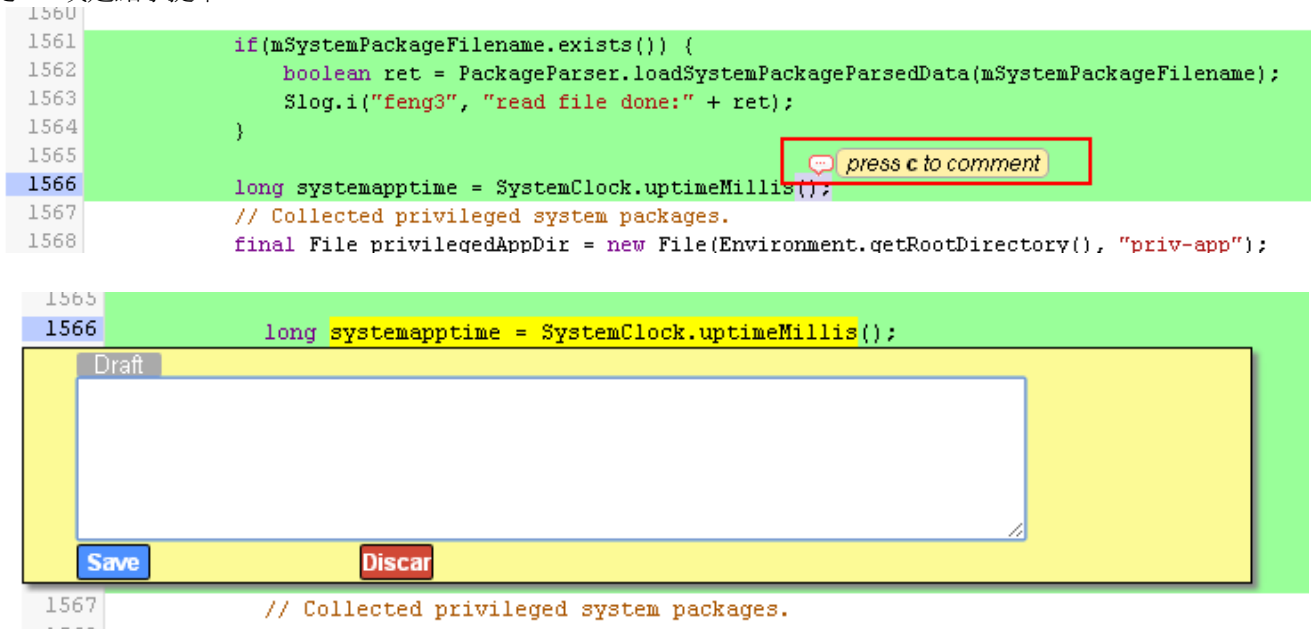
點擊上圖檔案名稱以後，將會進入 diff 頁面。如下圖。

All	My	Projects	People	Documentation
Changes	Drafts	Draft Comments	Watched Changes	Starred Changes
Groups				
<input checked="" type="checkbox"/> platform/frameworks/base / services/core/java/com/android/server/pm/PackageManagerService.java				
1532				1543
1533				1544
1534				1545
1535				1546
1536				1547
1537				1548
1538				1549
1539				1550
1540				1551
1541				1552
1542				1553
1543				1554
1544				1555
1545				1556
1546				1557
1547				1558
1548				1559
1549				1560
1550				1561
1551				1562
1552				1563
1553				1564
1554				1565
1555				1566
1556				1567
1557				1568
1558				1569
1559				1570
1560				1571
1561				1572
1562				1573
1563				1574
1564				1575
1565				1576
1566				1577
1567				1578
1568				1579
1569				1580
1570				1581
1571				1582
1572				1583
1573				1584
1574				1585
1575				1586
1576				1587
1577				1588
1578				1589

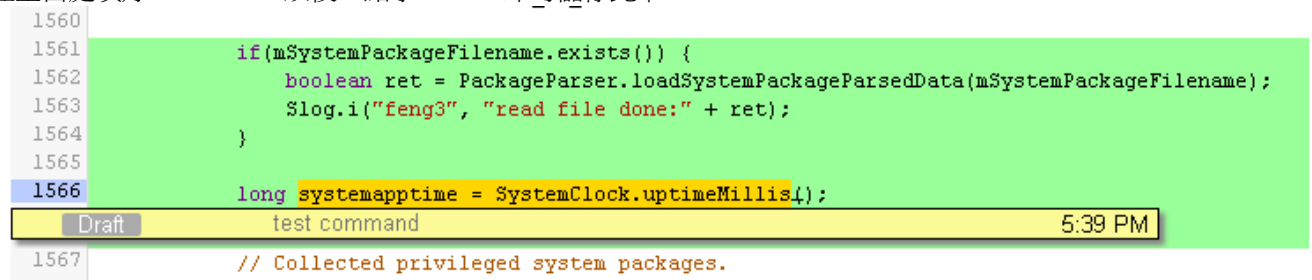
右邊為修改後的樣子，左邊是比對的基準。

以此圖為例，可以看出檔案中新增了兩個區塊。

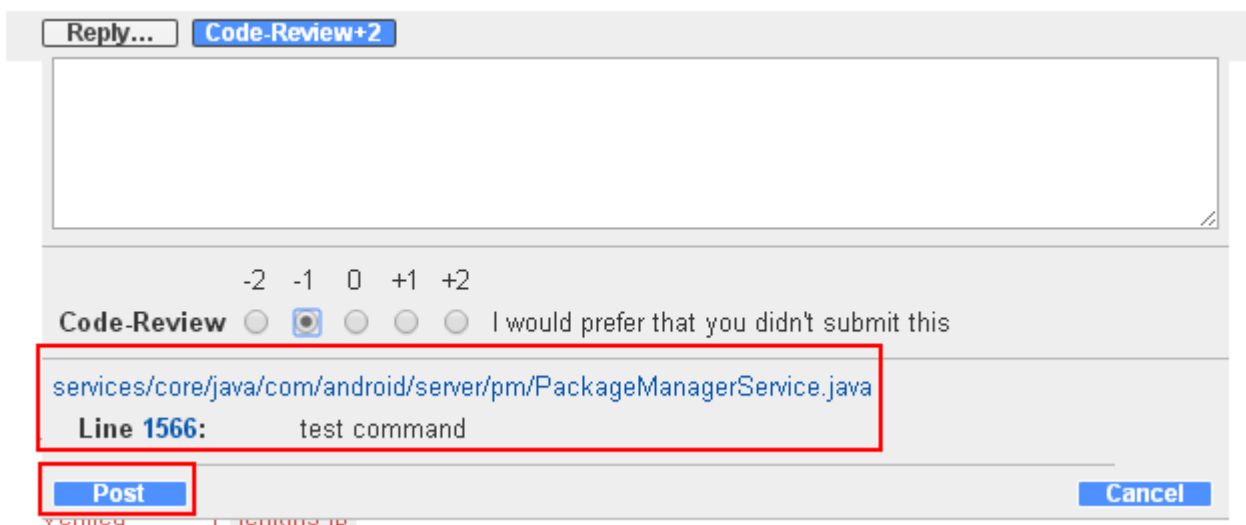
如果想要對修改的 code 加上 review 的註解，可以在右邊的畫面上，要加 comments 的地方反白選取，再按下快捷鍵 **c**，或是點擊提示。



在空白處填好 comments 以後，點擊 **save**，即可儲存此筆 comments。



加完以後，在最後評分時，可以看到加過的 comments 位置與行數。



注意：加註完所有的 **comments** 以後，需要進行完評分動作，上 **code** 的人才能透過 **commit** 頁面的 **history** 區塊看到 **comments**，不然都只會是 **draft**。

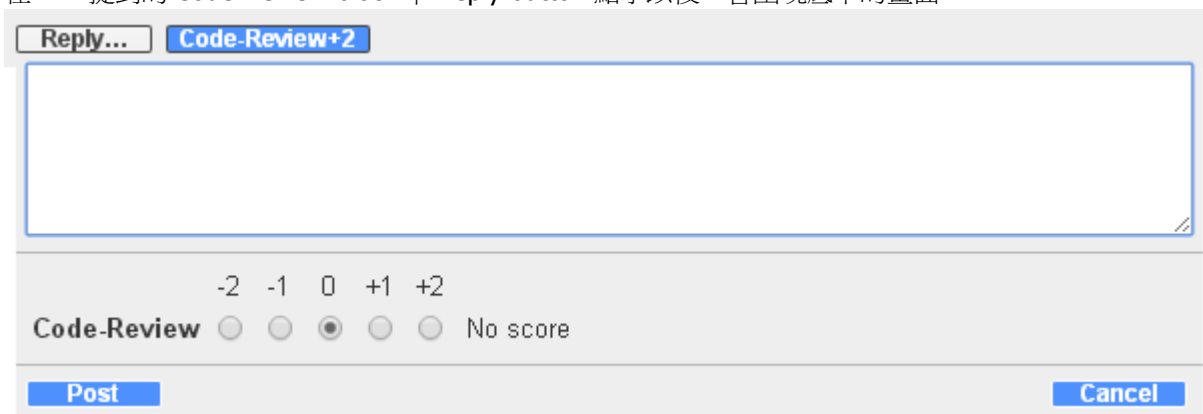
完成評分以後，上 code 的人可以透過 commit 頁面的 history 區塊看到加註個 comments

jenkins-jb	Patch Set 3: Build Started http://hccis03.mstarsemi.com.tw:8080/job/OVF_Gerrit_LMF
jenkins-jb	Patch Set 3: Build Started http://hccis03.mstarsemi.com.tw:8080/job/OVF_Gerrit_LMF
jenkins-jb	Patch Set 3: Build Started http://hccis03.mstarsemi.com.tw:8080/job/OVF_Gerrit_LMF
jenkins-jb	Patch Set 3: Verified-1 Build Failed http://hccis03.mstarsemi.com.tw:8080/job/OVF_Ger
Feng Lin	
Patch Set 3: Code-Review-1	
(1 comment)	
services/core/java/com/android/server/pm/PackageManagerService.java	
Line 1566:	test command

3.3. +1/-1/+2/-2 評分

Reviewer 在看完 code 以後，可以評分的分數有四種。（視乎權限設定，較低權限的 reviewer 可能僅有+1/-1 的權限）。

在 2.2 提到的 Code-Review block 中 reply button 點擊以後，會出現底下的畫面。



The image shows a 'Code-Review+2' dialog box. At the top, there are two buttons: 'Reply...' and 'Code-Review+2'. Below them is a large text input area. Under the input area, there are five radio buttons labeled -2, -1, 0, +1, and +2. The +2 radio button is selected. To the right of these buttons is a 'No score' label. At the bottom left is a 'Post' button, and at the bottom right is a 'Cancel' button.

在這個界面，reviewer 可以在中間空白欄位寫上這次 review 的 comments，並且給予評分，然後按下 post 完成評分。

各分數所代表的意義如下：

- +2 : looks good to me, approved.
- +1 : looks good to me, but someone else must approve.
- 0 : no score.
- 1 : I would prefer that you don't submit this.
- 2 : Do not submit.

要特別注意的是，這些分數並不是採累加制，也就是說，有兩個 reviewer 對此筆 commit 分別給予+1 的評分，並不等同於有一個 reviewer 給予+2 的評分。

所以可以給予+2 評分的 reviewer 將有責任做好這筆 review 的最後把關。

3.4. Submit code

在完成評分以後，若分數已達+2，submitter 的 review 頁面上，將會出現 submit 按鈕。如下圖。

The screenshot shows a code review interface. At the top is a 'Reply...' button. Below it, the review details are listed: Owner (Feng Lin), Reviewers (Fred Chang with a close icon), Project (GitTesting/FW with a gear icon), Branch (test-dev), Topic (with a notepad icon), Strategy (Fast Forward Only), and Updated (in the future). At the bottom of the details are three buttons: 'Cherry Pick' (red), 'Abandon' (red), and 'Submit' (blue, highlighted with a red box). Below the details, the review scores are shown: 'Code-Review +2' by Feng Lin and '+1' by Fred Chang.

Submit 按鈕點下去以後，就完成了這筆 commit merge 到 git server 上的動作。在沒有完成這個動作以前，git server 上都不會有這筆 commit 在裡面。也就是說，其他人用 git pull、git clone 等抓 code 指令，都抓不到這筆 commit。僅能透過 git cherry-pick 取得暫時的修改。

一般 submit 權限會跟+2 權限綁在一起，所以是否將這筆 commit 合進 code line 中，最後一道關卡就是擁有 submit 權限的人。

4. Q&A

本節整理常常會遇到的問題。

4.1. Q: Patch set 是什麼？跟 perforce 的 revision 有對應關係嗎？

A: 一個 Patch set 對應一次 push，也對應一個 commit id。

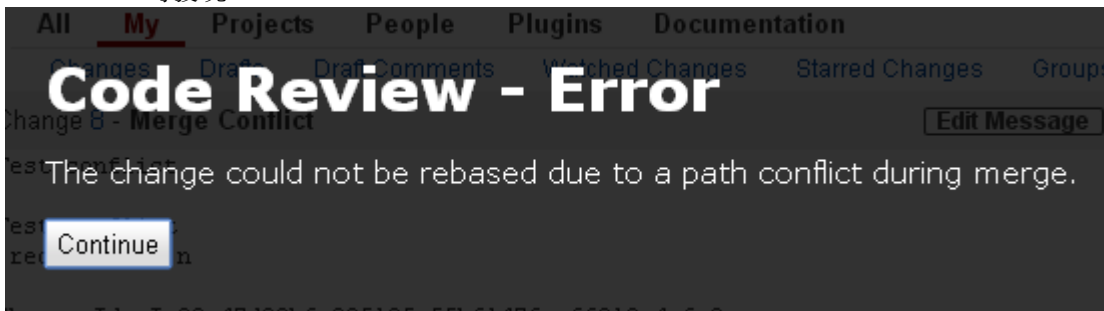
Ex: 大毛爲了修正一個 mantis 而上了一個 commit，因爲 reviewer 的 reject，大毛重新修改了 solution 又 push 了一次，因此這個 change id 便有了兩個 patch set 出現。Patch set 1 爲第一次上 code 時的改動，Patch set 2 爲重新修改後的改動。

Perforce 的 revision 是已經完成上 code 的 history，而 patch set 僅是代表這筆 commit 的修改歷史，並不是已經上 code 的內容，即使這筆 commit 已經被 merge 了，被 merge 進去的也只有最後一個 patch set。

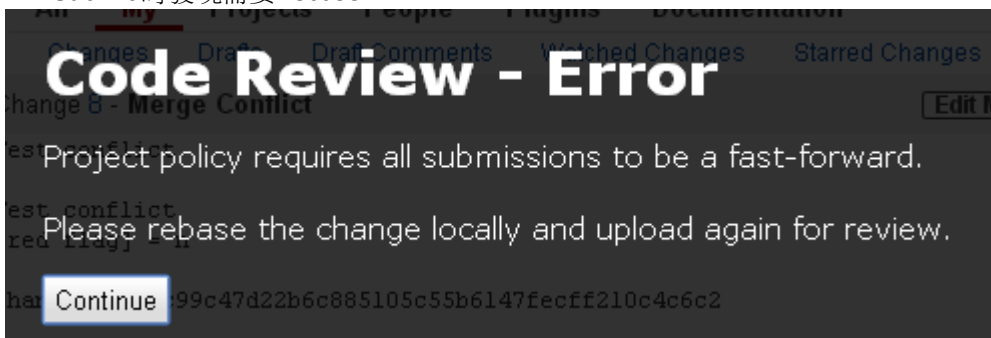
4.2. Q: 當 review 完成要 submit 時，發生 error 該怎麼處理呢？

A: submit 時發生 error 通常有幾種情況。

1. Rebase 時發現 conflict。



2. Submit 時發現需要 rebase。



處理方式：評判分數為 -2，加上 comment 請 owner 修正 conflict 以後，重新 push 一次。

Reply...

```
rebase fail due to conflict
```

-2 -1 0 +1 +2

Code-Review ☒ ☐ ☐ ☐ ☐ This shall not be merged

Post

Cancel

4.3. Q: