

人工智能与智慧运营研发中心 软件开发规范

CL (changelist) 编写规范

版本号：1.1

2019-10-24

人工智能与智慧运营研发中心

目 录

一、 CL (changelist) 简介 1

 1. 什么是 CL，为什么要写 CL 1

 2. CL 描述规范..... 1

二、 CL 简洁化..... 2

 1. CL 为什么需要简洁化..... 2

 2. 如何描写简洁的 CL..... 2

一、CL（changelist）简介

1. 什么是 CL，为什么要写 CL

CL 记录了代码变更历史，它描述了这个变更做了什么，为什么采取这种方案。

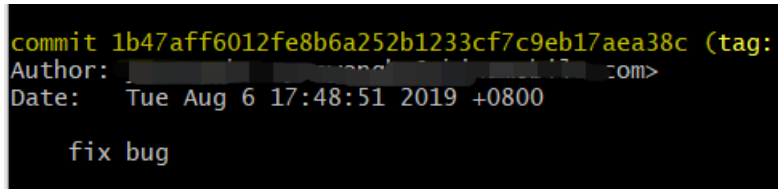
如果一段代码变更没有 CL 描述，那么开发者很难定位该变更对应的问题。

简而言之，CL 就是命令：`git commit -m “changelist”` 中引号部分。

2. CL 描述规范

类似于“修复一个 bug”，“打了一个 patch”等非常笼统含糊的 CL 是非常不可取的。

例 1：下图提交的问题显而易见，不知道修复了什么问题，这个 bug 是否有对应的 Bug Id？也不知道这个 bug 的修复，是否会造成重大的影响？



```
commit 1b47aff6012fe8b6a252b1233cf7c9eb17aea38c (tag:
Author: jason
Date: Tue Aug 6 17:48:51 2019 +0800

    fix bug
```

例 2：对同一个版本某块代码进行重构，不同的人可能描述为 refactor、重构、change 等，很难统计哪些代码做了重构。

因此，对 CL 描述做固定的规范是非常有必要的。

建议采取如下规范：

`<type>-<JIRAID>:<subject>`

`//空行`

`<body>`

解析：

`<type>`：表明 commit 的作用，主要用于分类。

- feat:新功能（feature）
- fix:修补 bug
- doc:文档（document）
- style: 格式（不影响代码运行的变动）
- refactor:重构（既不是新增功能，也不是修改 bug 的代码变动）
- test:增加测试
- chore:构建过程或辅助工具的变动

`<JIRAID>`：可通过搜索 JIRAID 将不同功能、测试等的所有提交归类。

例如：某功能需求共拆分为 10 条，对应 JIRAID 为 ONAP-1 ~ ONAP-10，开发完成后，可运行以下命令对比查看是否所有需求都已纳入开发：`git log --oneline | grep “feat-ONAP-*”`。

<subject>: 用于显示简要的说明，需言简意赅。

<body>:具体描述：问题解决方式，为什么这么做，缺点是什么，背景知识，相关上下文链接。

例如：某提交修改了 README.md 文档，CL 如下图：

```
commit 1de4aa81babb797dac0104d4cfe35255ab93132d (HEAD)
Author: <[redacted]>
Date:   Wed Oct 23 11:24:31 2019 +0800

    doc-BUG-124: 更新模块说明文档README.md

    增加了两个子模块A和B
    并为两个子模块分别添加负责人
    A: xiaoming
    B: xiaohong
```

二、 CL 简洁化

1. CL 为什么需要简洁化

审查更快捷、更完整，引入错误概率小，即使被驳回损失也很小，更容易被合并，设计更简单，不妨碍当前工作，回滚更容易。

2. 如何描写简洁的 CL

CL 简洁化使审查者不必深入查看代码就可以了解当前代码的功能。

例如：

运行：git log --oneline

```
$ git log --oneline
060bcddeb (HEAD -> master, origin/master, origin/HEAD) chore(saucelabs): switch to latest version of desktop Safari
881167d7b chore(saucelabs): upgrade SauceConnect to 4.5.4
1147f0e21 docs($compile): fix typos and incorrect example
825e25b09 docs(guide/migration): fix typo (misceallenous --> miscellaneous)
10d1b1973 fix(ngStyle): correctly remove old style when new style value is invalid
c8d985a0f chore(*): update jQuery from 3.2.1 to 3.4.0
55075b840 docs(DEVELOPERS): remove git revert misleading information
c8980099d chore(*): update copyright year
e629fe4a7 docs(CHANGELOG): add 1.7.8 release notes
6959bc297 fix(required): correctly validate required on non-input element surrounded by ngIf
a1d15649e docs(guide/di): clarify example description
cbf35f04f chore(github): update Issue template with LTS info
f76802369 chore(github): update PR template to highlight LTS mode
587843bca chore(scripts): use https for git repo URLs
94582f0de docs(CHANGELOG): update with 1.7.7 release notes
fffe7d197 docs(RELEASE): store release instructions
005dd9725 fix(ngRequired): set error correctly when inside ngRepeat and false by default
9f7144b03 docs(angular.copy): fix list of unsupported types
65f800e19 docs(angular.merge): add notes about support and lodash compatibility
63796d3f9 docs(angular.copy): list object types / features that are not supported
2d8b34670 docs(angular.copy): add note about destination and source compatibility
fbc83f00c docs(ngParseExt): note limitations with ngOptions and ngRepeat
0633d8f2b docs(input[email]): note limitations with IPv6 addresses
782577292 docs(ngDisabled): remove restriction to input element
```

简洁 CL 描写原则：

- 一个小的功能单位
- 内容清晰
- 被合入后保证系统可持续正常运行
- 通过将 CL 分类，便于审查者一目了然

例如：

查看文档相关的改动：

运行：git log --oneline | grep “docs(”

```
$ git log --oneline -35 | grep "docs("
1147f0e21 docs($compile): fix typos and incorrect example
825e25b09 docs(guide/migration): fix typo (misceallenous --> miscellaneous)
55075b840 docs(DEVELOPERS): remove git revert misleading information
e629fe4a7 docs(CHANGELOG): add 1.7.8 release notes
a1d15649e docs(guide/di): clarify example description
94582f0de docs(CHANGELOG): update with 1.7.7 release notes
fffe7d197 docs(RELEASE): store release instructions
9f7144b03 docs(angular.copy): fix list of unsupported types
65f800e19 docs(angular.merge): add notes about support and lodash compatibility
63796d3f9 docs(angular.copy): list object types / features that are not supported
2d8b34670 docs(angular.copy): add note about destination and source compatibility
fbc83f00c docs(ngParseExt): note limitations with ngOptions and ngRepeat
0633d8f2b docs(input[email]): note limitations with IPv6 addresses
782577292 docs(ngDisabled): remove restriction to input element
1e8a3de15 docs(guide/Conceptual Overview): use exchangeratesapi.io
```

查看 bug 修复相关的改动：

运行：git log --oneline | grep fix “fix(”

```
cmcc-xhq@cmcc-PC MINGW64 /d/code/GIT/temp/angular.js (master)
$ git log --oneline -35 | grep "fix("
10d1b1973 fix(ngStyle): correctly remove old style when new style value is invalid
6959bc297 fix(required): correctly validate required on non-input element surrounded by ngIf
005dd9725 fix(ngRequired): set error correctly when inside ngRepeat and false by default
8b973e04c fix($compile): fix ng-prop-* with undefined values
```

➤ 对于相互依赖的 CL 在提交时要避免影响 build 编译

例如：changelist B 依赖于 changelist A, 因此提交时应注意：

提交顺序：需先提交 A, 再提交 B。

提交时间间隔：尽量缩短 AB 提交时间间隔，避开编译时间，避免在间隔期间编译因未来得及提交导致编译失败。

➤ 将相关的测试代码保存在同一 CL 中，避免漏测

避免将测试代码拆分为单独的 CL，验证代码修改的测试应该进入相同的 CL。

CL 不能太过简洁让人难以理解，比如你新加了一个 API 接口，至少要有它的用法，这样也可防止一些无用的 API 被合入。

对于某些特殊情况 CL 无法精简，例如：重构工具导致的大量代码变动、删除整个文件等，可以提前告知代码审查者，说明原因，让其有心理预警更易接受。同时务必做好测试，避免引进 bug。

没有硬性标准界定 CL 大小，具体情况具体分析。当你实在不确定 CL 是否足够简洁，那就进一步精简它。

修 订 记 录

版 本	人 员	修 订 情 况
V1.0	徐 惠 群	2019-10-23 初 稿
V1.1	徐 惠 群	2019-10-24 移 动 “ 如 何 处 理 代 码 审 查 ” 章 节 至 代 码 审 查 规 范