潘荣涛

UIT分支管理策略

介绍UIT分支管理策略

德赛西威

[ 主分支 3](#_Toc9938)

[ 多人合作 3](#_Toc20517)

[ 提交 3](#_Toc19237)

# 主分支

由于UIT的分支是基于Baseline的，因此master分支在工作中基本上是没有太多意义，包括开发和集成过程。

在每个Baseline基准下，诸如1.0，1.1， 2.0等这些都是主分支。开发者自检的分支为非主分支。

# 多人合作

以上提到，数据库以Baseline作为准则，每个版本都有其主分支：1.0, 2.0等。那么，在多人合作开发一个数据库的时候，该怎样高效合作呢？

***为了维护主分支的纯净和高追溯性，我们应该尽可能的在关键节点进行主分支的提交（例如集成节点）。而对于日常的开发需要，应当以自己的uid号新建开发分支。等到新任务完成或者bug修复后提交该分支到主分支上。***

每个数据库应该有一个负责合并的人，以这种方式，来保持主分支的纯净和低冲突。

# 提交

Commits Message修改要求如下：

1. 格式1：[模块:] [子模块:] 修改内容

例：uit-fw-data: controller: fix connect logic

    uit-lib-core: add interface getXXX

1. 格式2：[模块:] [子模块:] 修改内容 for 项目

例：uit-fw-screen: layer manager: add navi layer for support for M36T

1. 字段间需增加一个空格作为间隔, []表示，该字段可按需添加；
2. 不可以包含中文内容；