

SVN 生命周期

本章讨论了版本控制系统的生命周期。

创建版本库

版本库相当于一个集中的空间，用于存放开发者所有的工作成果。版本库不仅能存放文件，还包括了每次修改的历史，即每个文件的变动历史。

Create 操作是用来创建一个新的版本库。大多数情况下这个操作只会执行一次。当你创建一个新的版本库的时候，你的版本控制系统会让你提供一些信息来标识版本库，例如创建的位置和版本库的名字。

检出

Checkout 操作是用来从版本库创建一个工作副本。工作副本是开发者私人的工作空间，可以进行内容的修改，然后提交到版本库中。

更新

顾名思义，update 操作是用来更新版本库的。这个操作将工作副本与版本库进行同步。由于版本库是由整个团队共用的，当其他人提交了他们的改动之后，你的工作副本就会过期。

让我们假设 Tom 和 Jerry 是一个项目的两个开发者。他们同时从版本库中检出了最新的版本并开始工作。此时，工作副本是与版本库完全同步的。然后，Jerry 很高效的完成了他的工作并提交了更改到版本库中。

此时 Tom 的工作副本就过期了。更新操作将会从版本库中拉取 Jerry 的最新改动并将 Tom 的工作副本进行更新。

执行变更

当检出之后，你就可以做很多操作来执行变更。编辑是最常用的操作。你可以编辑已存在的文件来，例如进行文件的添加/删除操作。

你可以添加文件/目录。但是这些添加的文件目录不会立刻成为版本库的一部分，而是被添加进待变更列表中，直到执行了 commit 操作后才会成为版本库的一部分。

同样地你可以删除文件/目录。删除操作立刻将文件从工作副本中删除掉，但该文件的实际删除只是被添加到了待变更列表中，直到执行了 commit 操作后才会真正删除。

Rename 操作可以更改文件/目录的名字。"移动"操作用来将文件/目录从一处移动到版本库中的另一处。

复查变化

当你检出工作副本或者更新工作副本后，你的工作副本就跟版本库完全同步了。但是当你对工作副本进行一些修改之后，你的工作副本会比版本库要新。在 commit 操作之前复查下你的修改是一个很好的习惯。

Status 操作列出了工作副本中所进行的变动。正如我们之前提到的，你对工作副本的任何改动都会成为待变更列表的一部分。

Status 操作就是用来查看这个待变更列表。

Status 操作只是提供了一个变动列表，但并不提供变动的详细信息。你可以用 diff 操作来查看这些变动的详细信息。

修复错误

我们来假设你对工作副本做了许多修改，但是现在你不想要这些修改了，这时候 revert 操作将会帮助你。

Revert 操作重置了对工作副本的修改。它可以重置一个或多个文件/目录。当然它也可以重置整个工作副本。在这种情况下，revert 操作将会销毁待变更列表并将工作副本恢复到原始状态。

解决冲突

合并的时候可能会发生冲突。Merge 操作会自动处理可以安全合并的东西。其它的会被当做冲突。例如，"hello.c" 文件在一个分支上被修改，在另一个分支上被删除了。这种情况就需要人为处理。Resolve 操作就是用来帮助用户找出冲突并告诉版本库如何处理这些冲突。

提交更改

Commit 操作是用来将更改从工作副本到版本库。这个操作会修改版本库的内容，其它开发者可以通过更新他们的工作副本来查看这些修改。

在提交之前，你必须将文件/目录添加到待变更列表中。列表中记录了将会被提交的改动。当提交的时候，我们通常会提供一个注释来说明为什么会进行这些改动。这个注释也会成为版本库历史记录的一部分。Commit 是一个原子操作，也就是说要么完全提交成功，要么失败回滚。用户不会看到成功提交一半的情况。

[← SVN 安装](#)[SVN 启动模式 →](#)[✎ 点我分享笔记](#)