# 建立 CVS 储存库（repository）- 以 FreeBSD 的方式

Stijn Hoop <stijn@win.tue.nl>

修订：43126



2013-11-07 由 gabor.

## 摘要

这份文件描述了使用和 FreeBSD 项目相同的命令脚本来建立 CVS 储存库的步骤。 这和标准 CVS 建立的储存库相较之下有许多优点， 它提供了更多对于源代码树的细粒度访问控制， 并能够为每一次的提交生成和发出易读的电子邮件。

## 目录



## 1. 简介

大多数的开放源代码软件项目都使用 CVS 作为它们的源代码控制系统。 尽管 CVS 有许多的优点， 但它也有部份的瑕疵和缺点。 其中之一的原因是和其它的开发者分享源代码树可能会迅速成为系统管理的恶梦， 特别是当有人希望保护部份的源代码树免受于一般的存取时。

FreeBSD 是众多使用 CVS 的项目之一， 因为基于它进行开发的开发人员遍布于全世界。 他们撰写了一些命令脚本使得管理储存库变得更加容易。 最近这些命令脚本由 Josef Karthauser 重新整理过且更标准化， 使得在其它的项目上再次使用这些命令脚本会更加容易。 本文件将描述使用这些新的命令脚本的方法。

为了使本文件中的信息对您有用， 您需要首先熟悉 CVS 的基本操作方式。

## 2. 基本配置

> **警告**
>
> 最好的方式是在一个全新的储存库中执行这些步骤， 并确定你了解所有的后果。 同时， 请确定你有最新且可读的资料备份！

### 2.1. 初始化储存库

首先要做的是建立一个新的储存库， 执行下列命令告诉 CVS 建立并初始化：

```
% cvs --d path-to-repository init
```

这命令告诉 CVS 建立 CVSROOT 的目录， 这个目录里放置了所有的配置文件。

### 2.2. 配置储存库的用户组

现在我们将建立一个拥有该储存库的用户组， 所有的开发者必须加入这个用户组， 这样他们才能够存取该储存库。 我们假设用户组名称是以 FreeBSD 内部所采用的 ncvs。

```
# pw groupadd ncvs
```

接着你需要使用 chown(8) 将目录所有者指定给刚刚新增的用户组：

```
# chown --R -:ncvs path-to-your-repository
```

如此一来， 没有适当的用户组许可的用户， 就不再能够写入该储存库。

### 2.3. 取回源文件

现在你需要从 FreeBSD 储存库中取回 CVSROOT 目录， 从 FreeBSD 匿名的 CVS 镜像站来取回会是最简单的方法。请查阅 在 使用手册 中的相关章节 来获得更多信息。 我们假设取回的文件存放在相同目录下的 CVSROOT-freebsd 目录中。

### 2.4. 复制 FreeBSD 的命令脚本

接下来我们要复制 FreeBSD CVSROOT 里的文件到你的储存库中。 如果你熟悉 CVS， 你也许会想你可以直接汇入 (import) 这些命令脚本， 从而更容易地在未

来有新版时进行版本同步； 不过，事实是 CVS 在这个部份有缺点： 当汇入文件到 CVSROOT 时， 它并不会更新配置文件。 为了要认出这些文件， 你还需要在汇入它们后一一重新提交， 这就失去了 cvs import 的价值。 因此， 推荐的方法是直接将这些命令脚本复制过去。

如果您不了解这些操作也没有关系 — 因为最后的结果都是一样的。 首先汇出 (checkout) 你的 CVSROOT， 然后复制刚刚取回的 FreeBSD 文件到本地的目录中（尚未变动过）：

```
% cvs --d path-to-your-repository checkout CVSROOT
% cd CVSROOT
% cp -../CVSROOT-freebsd/* -.
% cvs add *
```

注意： 你很可能会得到一段关于某些目录没有被复制的警告，这是正常的， 你并不需要用到这些目录。

## 2.5. 命令脚本说明

现在你的工作目录中有了完整 FreeBSD 项目在他们的储存库中使用的命令脚本的副本， 以下是每个文件简单的介绍。

- access – 此文件在预设的安装中没有被用到。 它是在 FreeBSD 的专用配置 中用来控制储存库的存取的。 如果你不希望使用这个配置的话， 则可以删除这个文件。

- avail – 此文件控制储存库的存取。 在此文件中你可以指定允许存取储存库的用户组， 也可以针对目录或文件来拒绝提交。 你应该调整为在你的储存库中将包含的用户组和目录。

- cfg.pm – 此文件说明了配置内容，并提供预设的配置。 你 不 应修改此文件，而应将修改的配置放到 cfg_local.pm。

- cfg_local.pm – 此文件包含所有的系统配置值。 你应该配置所有列在此的配置，例如提交的邮件要寄到哪、 在哪些主机上的使用者可以提交等等。 更多的相关信息在稍后会提到。

- checkoutlist – 此文件列出所有在 CVS 控制下此目录中的文件， 除了标准在 cvs init 建立出的文件。 你可以删除某些不需要的 FreeBSD 专用的文件。

- commit_prep.pl – 此命令脚本执行各种提交前的检查， 这些检查是否启用， 取决于您在 cfg_local.pm 中所进行的配置。 你不应更动此文件。

- commitcheck – 此命令脚本会直接影响 CVS。 首先它会使用 cvs_acls.pl 来检查提交者是否可以存取指定的源代码树， 然后执行 commit_prep.pl 来确认各种提

交前的检查。 如果一切正常，CVS 将允许此次提交继续执行。 你不应更动此文件。

- commitinfo - 此文件是 CVS 用来定义在提交前所要执行的程序 — 在此例中是 commitcheck。你不应更动此文件。
- config - 储存库的配置选项。 你可以修改为你想要的， 但大多数的管理者可能会保留默认值。 更多关于可以在此配置的选项信息可以查阅 CVS 手册。
- cvs_acls.pl - 此命令脚本定义提交者的身分， 以及他/她是否允许存取源代码树， 这些判断基于 avail 中的配置。 你不应更动此文件。
- cvsignore - 此文件列出哪些文件 CVS 不用处理到储存库中， 你可以修改成你想要的。 更多关于可以此文件的说明可以查阅 CVS 手册。
- cvswrappers - 此文件是 CVS 用来启用或停用关键词展开， 或者是否应将文件视为二进制文件。 你可以修改成你想要的。 更多关于可以此文件的说明可以查阅 CVS 手册。 注意 -t 和 -f 选项在 CVS 客户端/服务器 并不能够正确地运作。
- edithook - 此文件已经不再使用了， 仅为历史原因保留。 你可以安全地删除此文件。
- editinfo - CVS 使用这个文件来强迫你使用特定的编辑器。 FreeBSD 没有使用这个功能， 而对输入的日志信息的检查， 则由 verifymsg 和 logcheck 来完成。 这是因为 editinfo 功能在从远程提交或是使用 -m 或 -F 选项时不会执行。 你不应更动此文件。
- exclude - 此文件列出被 commit_prep.pl 定义不能包含修正版标头的文件。 在 FreeBSD 版本的配置中， 所有在修正版控制下的文件需有一个修正版标头，（类似 $FreeBSD$ 这样）。 此文件逐行列出不应进行检查的文件名字。 你可以在此文件中为不需要修正版标头的文件新增一个正则表达式。 为了安装这些命令脚本， 最好的方法是将 CVSROOT/ 从标头检查中排除。
- log_accum.pl - 此命令脚本会处理由 logcheck 所提供的日志信息， 并且将之为备份目的附加于储存库中的记录文件。 同时也执行要将邮件寄到你提供的信箱中的程序（在 cfg_local.pm 中）。 它和 CVS 之间是由 loginfo 负责沟通。 你不应更动此文件。
- logcheck - 此文件分析提交者提供的日志信息， 并试图对其作清理动作。 它和 CVS 之间是由 verifymsg 负责沟通。 你不应更动此文件。

> **注意**
>
> 此命令脚本依赖于经过 FreeBSD 修改的 CVS: FreeBSD 版本在此命令脚本修改过后才读取日志信息； 标

准的 CVS 版本虽然能够检查语法上是否正确， 但并不会清理日志信息。CVS 1.11.2 可以通过在 config 配置 RereadLogAfterVerify=always 来和 FreeBSD 版本有相同的作用。

- loginfo - 此文件是 CVS 用来控制日志信息要寄到哪里，而 log_accum.pl 负责处理。你不应更动此文件。
- modules - 此文件保留了 CVS 原始的意义。 你应该删除新增的 FreeBSD 模块，并修改为你想要的内容。 更多关于可以此文件的说明可以查阅 CVS 手册。
- notify - 此文件为 CVS 用来控制监看某个文件。 在 FreeBSD 的储存库中没有用到此文件， 你可以修改成你想要的。 更多关于可以此文件的说明， 可以查阅 CVS 手册。
- options - 此文件仅限使用于 FreeBSD 和 Debian 的 CVS 版本。 它包含了需要在修正版标头中展开的关键词。 你可以修改为符合你在 cfg_local.pm 中指定的关键词。
- rcsinfo - 此文件定义提交时， 储存库所要使用的日志信息样式模板， 如 rcstemplate。 FreeBSD 预设为所有的储存库使用同一个样式模板， 你可以加入其它你想要的。
- rcstemplate - 此文件是提交者在提交时会看到的日志信息样式模板， 你应该修改为你在 cfg_local.pm 中定义的各种参数。
- tagcheck - 此文件控制在储存库中贴上标签的存取。 标准的 FreeBSD 版本拒绝名为 RELENG* 的标签，因为这是 交付工程组 的工作。 你可以根据需要来修改此文件。
- taginfo - 此文件控制执行在储存库中贴上标签的存取的命令脚本， 如 tagcheck。你不应更动此文件。
- unwrap - 此命令脚本可以用来在汇出时自动 “解开” 二进制文件 (请见 cvswrappers)。 目前 FreeBSD 并没有使用此配置， 因为此功能在远程提交时执行的并不十分完善。 你不应更动此文件。
- verifymsg - 此文件用来执行和日志信息相关的命令脚本， 如 logcheck。你不应更动此文件。
- wrap - 此命令脚本可以用来在提交时自动 “包裹” 二进制文件 (请见 cvswrappers)。 目前 FreeBSD 并没有使用此配置， 因为此功能在远程提交时执行的并不十分完善。 你不应更动此文件。

## 2.6. 定制命令脚本

接下来的步骤要配置这些命令脚本使得它们可以在你的环境中运作。 你应该检查所有在目录中的文件， 并修改为符合你的配置。 尤其， 你会想要修改下列的文件：

1. 如果你不希望使用 FreeBSD 的专用配置， 你可以安全地删除 access：

   ```
   % cvs rm --f access
   ```

2. 编辑 avail 来包含你想控制存取的各种储存库目录， 请确定你有保留 avail|| CVSROOT 这一行， 否则你将会在下一步把你自己锁在外面。

   另外你可以在此文件中新增开发者的用户组，FreeBSD 预设使用 access 来列出所有的开发者， 但你可以使用任何你想要用的文件。 如果你想的话也可以新增用户组（请使用指定在 cvs_acls.pl 前面所介绍的语法）。

3. 编辑 cfg_local.pm 来包含你需要的选项。 你应该特别检视一下下列的配置项目：

   - %TEMPLATE_HEADERS – 这是用来取得日志信息内容的程序， 并加入将呈现的邮件项目和提供非空值的信息。 你可以删除 PR 和 MFC after 叙述，当然也可以加入你想要的。
   - $MAIL_BRANCH_HDR – 如果你想要在每一封提交的邮件中加入描述是在哪一个分支中提交的标头， 那么请定义为符合你的配置。 如果你不想使用这样的标头， 那么请配置为空值。
   - @COMMIT_HOSTS – 定义使用者能够提交的主机。
   - $MAILADDRS – 配置应该收到提交邮件的邮件地址。
   - @LOG_FILE_MAP – 以你所需要的来修改这个数组， 每个配置值应该符合被提交的目录，而提交的日志信息会以 commitlogs 的名称储存在每个被配置的目录下。
   - $COMMITCHECK_EXTRA – 如果你不想使用 FreeBSD 专用的存取控制 功能， 你可以在此文件中删除对 $COMMITCHECK_EXTRA 的定义。

> **注意**
>
> 修改 $IDHEADER 的功能只有在 FreeBSD 平台上可以运作， 它依赖于 FreeBSD 专用的 CVS 配置。

你可以检查 cfg.pm 是否有其它的参数可以修改， 但是修改最好是有原因的。

4. 删除 exclude 中关于FreeBSD 的专用配置的叙述 (如以 ^ports/ 为开头的每一行等)。 此外， 注释掉以 ^CVSROOT/ 为开头的行列， 然后新增一行只有 ^CVSROOT/。 等到关键词展开的命令脚本安装好后， 你可以在 CVSROOT 目录中的文件里加上标头， 然后再恢复刚刚注释的行列， 但在你还没有提交前则只保持这样。

5. 编辑 modules， 并删除所有 FreeBSD 的模块。 加入你需要的模块。

6. 

> **注意**
>
> 此步骤只有在你于 cfg_local.pm 中指定了 $IDHEADER 才有必要配置 (只有在 FreeBSD 专用的 CVS 配置上才能够执行)。

   编辑 options 以符合你在 cfg_local.pm 中配置的标签名称。 在所有的文件中搜寻 FreeBSD 并替换为你配置的标签名称。

7. 修改 rcstemplate 为和在 cfg_local.pm 中相同的配置。

8. 选择性的删除在 tagcheck 中针对 FreeBSD 检查的配置。你可以仅仅在文件的最上层加上 exit 0 来取消所有标签的检查。

9. 在你完成前的最后一件事是确认 commitlogs 可以正确储存。 预设会储存在储存库中 CVSROOT 里的 commitlogs 子目录中， 而这个目录需要事先建立:

```
% mkdir commitlogs
% cvs add commitlogs
```

现在， 在细心的检视过后， 你可以提交你的修改了。 确定你先前有在 avail 中允许你自己存取 CVSROOT 目录， 因为如果没有这样做的话你会把你自己锁在外面。 完整确认过后请执行下列命令:

```
% cvs commit -m '- Initial FreeBSD scripts commit'
```

## 2.7. 测试配置

你已经准备好做基本的测试了: 强制提交 avail 以确认每件事都如预期的运作。

```
% cvs commit -f -m 'Forced commit to test the new CVSROOT scripts' avail
```

如果一切正常， 那么恭喜了！ 你现在已经为你的储存库建立好 FreeBSD 的命令脚本了。 如果 CVS 仍然有警告什么， 回头检视上述的步骤是否有正确的执行。

# 3. FreeBSD 的专用配置

FreeBSD 项目自己使用一个有点不同的配置，那就是同时也使用 FreeBSD CVSROOT 中的 freebsd 子目录。 因为大量的 committer 必须在相同的用户组中， 因此项目写了一个简单的 wrapper 来确保 committer 可以正确的提交， 并配置储存库的用户组名称。

如果你的储存库也需要这样的功能，那么下面就会介绍如何建立， 不过首先要先来看一段复杂的概述。

## 3.1. FreeBSD 配置中使用的文件

- access – 此文件用来控制储存库的存取。 你应该编辑并加入所有在项目中的成员。
- freebsd/commitmail.pl – 此文件已经不再使用了， 只是因为历史原因而保留。 你不应更动此文件。
- freebsd/cvswrap.c – 此 CVS wrapper 源代码是用来建立检查所有存取的工作。更多的信息在稍后会提出。 你应该编辑 ACCESS 和 REALCVS 的路径以符合你的配置。
- freebsd/mailsend.c – 此文件是 FreeBSD 设定 mailing lists 需要的， 你不应更动此文件。

## 3.2. 步骤

1. 只把你的用户名加到 access 中。
2. 编辑 cvswrap.c 的路径以符合你的配置， 定义在大写的 ACCESS 中。 同时如果默认值不符合你的情况的话也应该修改本地实际的 cvs 程序所在位置。原始的 cvswrap.c 希望替代服务器端的 CVS 程序， 例如将其改名为 /usr/bin/ncvs。

   我的 cvswrap.c 是这样：

   ```
   #define ACCESS -"/local/cvsroot/CVSROOT/access"
   #define REALCVS -"/usr/bin/ncvs"
   ```

3. 接下来是建立 wrapper 来确认你在提交时是在正确的用户组中。 在你的 CVSROOT 中的 cvswrap.c 要能够使用。

在你完成编辑并加入正确的路径后我们要来编译源代码:

```
% cc --o cvs cvswrap.c
```

然后进行需要配置 (此步骤需要 root 权限) :

```
# mv -/usr/bin/cvs -/usr/bin/ncvs
# mv cvs -/usr/bin/cvs
# chown root:ncvs -/usr/bin/cvs -/usr/bin/ncvs
# chmod o-rx -/usr/bin/ncvs
# chmod u-w,g+s -/usr/bin/cvs
```

这会将 wrapper 安装成预设的 cvs 程序, 请确定任何要使用储存库的人应该有正确的存取权限。

4. 现在你可以删除所有在储存库用户组中的使用者, 所有的存取控制会经由 wrapper 完成, 同时 wrapper 会配置存取的正确用户组。

## 3.3. 测试配置

你的 wrapper 现在应该已经安装好了, 你当然也可以强制提交 access 来测试是否正常:

```
% cvs commit --f --m -'Forced commit to test the new CVSROOT scripts' access
```

同样地, 如果有错误, 检查是否上述所有步骤都有正确的执行。