

# SVN ADMIN 手册

用于 Svn Admin 3.0.2

## 目录

目录.....	1
1. 简介.....	3
1.1. 什么是 Svn Admin.....	3
1.2. 有什么优点.....	3
1.3. 是否适合你.....	3
2. 安装运行.....	3
2.1. 安装 Svn 服务器端.....	4
2.2. 安装 Apache 服务器并配置 svn.....	4
2.3. 安装 Java 运行环境.....	4
2.4. 安装 Tomcat 6.....	4
2.5. 初始化 svnadmin 数据库.....	4
2.6. 运行.....	4
3. svn 协议实例.....	5
3.1. 需求.....	5
3.2. 登录.....	5
3.3. 用户管理.....	6
3.4. 项目管理.....	6
3.5. 用户组管理.....	6
3.6. 设置目录权限.....	7
3.6.1. 设置管理组权限.....	8
3.6.2. 设置开发组的权限.....	8
3.6.3. 设置测试组的权限.....	9
3.6.4. 设置权限注意事项.....	9
4. http 协议单库实例.....	10
4.1. 创建项目.....	10
4.2. 配置 Apache.....	10
5. http 协议多库实例.....	11
5.1. 创建项目.....	11
5.2. 配置 Apache.....	11
6. 高级用户.....	11
6.1. 其他数据库支持.....	11
6.2. svn 协议输出配置文件.....	11
6.2.1. svnserve.conf.....	11

6.2.2. passwd.....	12
6.2.3. authz.....	12
6.3. http 协议单库输出配置文件.....	12
6.3.1. passwd.http.....	12
6.3.2. httpd.conf.....	12
6.3.3. authz.....	13
6.4. http 协议多库输出配置文件.....	13
6.4.1. passwd.http.....	13
6.4.2. httpd.conf.....	13
6.4.3. authz.....	13
6.5. 导入导出项目配置信息.....	13
6.6. svn 和 http 协议转换.....	14
6.6.1. svn 转为 http.....	14
6.6.2. http 转为 svn.....	14
6.7. 用户和项目用户.....	14
7. 附录.....	14
7.1. 数据库表结构.....	14
7.2. 版本记录.....	14
7.2.1. V 1.0.....	14
7.2.2. V 1.0.1.....	15
7.2.3. V 2.0.....	15
7.2.4. V 2.0.1.....	15
7.2.5. V 3.0.....	15
7.2.6. V 3.0.1.....	15
7.3. 联系方式.....	15

## 1. 简介

### 1.1. 什么是 Svn Admin

Svn Admin 是一个 Java 开发的管理 Svn 服务器的项目用户的 web 应用。安装好 Svn 服务器端好，把 Svn Admin 部署好，就可以通过 web 浏览器管理 Svn 的项目，管理项目的用户，管理项目的权限。使得管理配置 Svn 简便，再也不需要每次都到服务器手工修改配置文件。

### 1.2. 有什么优点

- ◆ Svn 项目配置数据保存在数据库，支持所有数据库（默认 MySQL/Oracle/SQL Server），支持所有操作系统。
- ◆ 权限控制：管理员可以管理所有信息、项目管理员可以管理项目、成员只能查看和修改自己的密码。
- ◆ 支持多项目、多用户、多用户组 Group（默认带有“项目管理组”、“项目开发组”、“项目测试组”）。
- ◆ 安全：密码加密保存。
- ◆ 支持 svn:// 协议和 http:// 协议(从 2.0 开始支持 Apache 服务器单库方式，从 3.0 开始支持 Apache 多库方式)

### 1.3. 是否适合你

- ◆ Svnadmin 在 Java 1.6、Tomcat 6、Subversion 1.6、MySQL 5.1、Apache 2.2、Windows 上开发测试通过，同时支持 Windows、Linux 等其他操作系统和数据库。
- ◆ 支持 svn 协议

Svn 的配置信息都在仓库目录的 conf 下的 authz,passwd,svnserve.conf 三个文件中，配置用户和权限都是通过修改 passwd 和 authz，立刻就生效。Svn Admin 的本质是对这 3 个文件进行管理，所有成员、权限的数据都保存在数据库中，一旦在 Svn Admin 的页面上修改，就会把配置信息输出到 conf 下的那 3 个配置文件中。

- ◆ 支持 http 协议

Apache+SVN 配置成功后可以有两种方式：

1) 单库方式：SVNPath 每个仓库单独配置各自的密码和权限文件。优点是各自分开，互相不影响，维护方便。缺点是增加或删除仓库需要修改 apache 的 httpd.conf 后重启。从 2.0 开始支持。

2) 多库方式：SVNParentPath 指定一个父目录，所有仓库在这个父目录下，使用一个密码文件和一个权限配置文件。优点是增加删除仓库不需要改 apache 的 httpd.conf，不需要重启 Apache。缺点是项目多会很混乱。从 3.0 开始支持。

*(PS:不知道是否可以使用 SVNParentPath 指定一个父目录，但密码文件和权限文件可以每个仓库单独配置?)*

修改 Apache 的 httpd.conf 需要重启 Apache，但是修改 svn 的密码和访问权限文件不需要重启 Apache 立刻就生效。

假如你不是 svn 管理员，或配置不是使用上述方式，你可以忽略下文。

## 2. 安装运行

如果你是升级版本，请参考【附录】【版本记录】章节。

### 2.1. 安装 Svn 服务器端

请到这里下载 <http://subversion.apache.org/packages.html>

### 2.2. 安装 Apache 服务器并配置 svn

如果你不使用 http 协议，略过这个步骤。

Apache 和 SVN 搭配使用请参考参考

<http://www.subversion.org.cn/tsvndoc/tsvn-serversetup.html#tsvn-serversetup-apache>。

建议(也是这里采用的方式):

例如配置其中一个工程 projar，需要在 httpd.conf 加上以下内容:

```
<Location /projar>
    DAV svn
    SVNPath D:/repository/projar
    AuthType Basic
    AuthName "Subversion repositories"
    AuthUserFile D:/repository/projar/conf/passwd
    AuthzSVNAccessFile D:/repository/projar/conf/authz
    Require valid-user
</Location>
```

如果项目多的话，频繁改动 httpd.conf。

可以把这段文本保存在另外一个地方如 D:/repository/projar/conf/a.conf，在 apache 的 httpd.conf 文件加入

```
Include D:/repository/projar/conf/a.conf
```

(关于 Include 指令请参考

<http://www.51os.com/manual/Apache/mod/core.html#include>)

### 2.3. 安装 Java 运行环境

这个简单吧？不会，上网搜索一下：)

### 2.4. 安装 Tomcat 6

哈哈，不用我教。

### 2.5. 初始化 svnadmin 数据库

假定你已经安装好数据库，在数据库中创建一个名为 svnamdin 的数据仓库，使用客户端工具连接上数据库。在发行包 db 目录中找到对应的数据库建表脚本，如 MySQL(请使用 UTF-8 编码创建数据库)数据库对应的是 mysql5.sql，执行里面的 sql，就建好 svnadmin 的表结构。

### 2.6. 运行

在发行包中找到 svnadmin.war 文件，使用压缩软件(例如 WinRAR)打开，修改里面的 WEB-INF/jdbc.properties 数据库连接信息。

例如你的数据库是 MySQL，服务器 IP 是 192.168.1.100，请修改红色标明的配置信息：

#指定当前连接数据库

db=MySQL

```
#MySQL
MySQL.jdbc.driver=com.mysql.jdbc.Driver
MySQL.jdbc.url=jdbc:mysql://192.168.1.100:3306/svnadmin?characterEncoding=utf-8
MySQL.jdbc.username=username
MySQL.jdbc.password=psw
```

修改好配置文件后，把 svnadmin.war 复制到 %Tomcat%/webapps 目录下，启动 Tomcat，假如你的 Tomcat 端口是 8080，在浏览器中输入 <http://192.168.1.100:8080/svnadmin>，出现登录页面。说明你安装成功。

### 3. svn 协议实例

#### 3.1. 需求

假定你有一个 jar 类型的 Eclipse Java 项目 projar，假定 SVN 仓库目录在服务器 192.168.1.100 的 D:/repository/projar，访问这个仓库的 svn 地址是：svn://192.168.1.100/projar，你的项目目录分配如下：

projar

```
|__ branches
|__ tags
|__ trunk
    |__ .project
    |__ .classpath
    |__ bin
    |__ src
        |__ *.java
```

假定项目经理是 Tom，有一个开发人员 Ben，一个测试人员 Kent。你希望 Tom 有管理这个项目目录的所有权限；Ben 可以检出 trunk 目录，但只能对 trunk/src/目录进行写的权限；Kent 只能读项目的 tags，检出发行版本进行测试。

#### 3.2. 登录

浏览器打开 <http://192.168.1.100:8080/svnadmin>

登录页面

第一次使用，输入的帐号和密码是超级管理员

欢迎使用SVN ADMIN, 第一次使用请设置管理员帐号和密码.

帐号	<input type="text"/>
密码	<input type="password"/>
<input type="button" value="登录"/>	

如输入帐号 svnamdin，密码 svnamdin，点击【登录】，进入首页。

## 项目管理

项目	*	协议	svn	*
路径	*			
URL	*			
描述				
提交				

## 3.3. 用户管理

点击右上角菜单【用户】，输入 Tom 信息，点击【提交】

## 用户管理

用户	tom	*	新密码	●●●●	角色	选择角色 ▼	提交
----	-----	---	-----	------	----	--------	----

同理：分别添加用户 kent 和 ben。密码都加密保存，结果如下：

NO.	用户	密码	角色	删除
1	<a href="#">ben</a>	YmVu		<a href="#">删除</a>
2	<a href="#">kent</a>	a2VudA**		<a href="#">删除</a>
3	<a href="#">svnadmin</a>	c3ZuYWRtaW4*	admin	<a href="#">删除</a>
4	<a href="#">tom</a>	dG9t		<a href="#">删除</a>

## 3.4. 项目管理

输入项目的基本信息，如下：

## 项目管理

项目	projar	*	协议	svn	*
路径	D:/repository/projar				
URL	svn://192.168.111.100/projar				
描述	projar测试				
提交					

【协议】可选 svn、http(单库)、http(多库)，这个例子使用 svn 协议。关于 http 协议，请参考【http 协议单库实例】和【http 协议多库实例】章节

【路径】是指服务器 svn 仓库的路径；

**如果 D:/repository 不存在仓库 projar，会自动的创建仓库，类似 svnadmin create 命令。如果已经存在 projar 目录，不会执行创建仓库命令。**

【URL】svn 协议使用 svn://开头

点击【提交】

## 3.5. 用户组管理

点击【设置用户组】，进入如下

### 项目管理(projar) --> 用户组管理

项目	projar	组号	<input type="text"/>	*	描述	<input type="text"/>	提交
----	--------	----	----------------------	---	----	----------------------	----

NO.	项目	组号	描述	设置用户	删除
1	projar	<a href="#">developer</a>	developer	<a href="#">设置用户</a>	<a href="#">删除</a>
2	projar	manager	manager	<a href="#">设置用户</a>	
3	projar	<a href="#">tester</a>	tester	<a href="#">设置用户</a>	<a href="#">删除</a>

点击第一行 developer 的【设置用户】

### 项目管理(projar) --> 用户组管理(developer)-->设置用户

<div>ben kent svnadmin tom</div>	<div>增加用户</div>
--	-----------------

NO.	项目	组	用户	删除
-----	----	---	----	----

在下拉框中选择 ben，点击【增加用户】，结果如下

### 项目管理(projar) --> 用户组管理(developer)-->设置用户

<div>kent svnadmin tom</div>	<div>增加用户</div>
--------------------------------------	-----------------

NO.	项目	组	用户	删除
1	projar	developer	ben	<a href="#">删除</a>

同理：分别添加 tom 到管理员组 manager，添加 kent 到测试组 tester。

把 Tom 添加到管理组【manager】后，Tom 已经具备默认的项目管理等权限，可以登录到 svn admin 修改项目资料、自己的用户资料和项目成员的资料。

## 3.6. 设置目录权限

点击【设置权限】，进入如下

[项目管理\(projar\)](#) --> [权限管理](#)

项目	projar	资源	<input type="text" value=""/> * 选择资源 ▼
用户组	<div> <div>developer</div> <div>manager</div> <div>tester</div> </div>	用户	<div> <div>*</div> <div>ben</div> <div>kent</div> <div>svnadmin</div> </div>
权限	没有权限 ▼		
保存			

NO. 项目 资源 用户组/用户 权限 删除

### 3.6.1. 设置管理组权限

输入如下信息：

【资源】: [projar:/] 表示项目的根

【用户组】: manager

【权限】: 可读可写

点击【保存】

[项目管理\(projar\)](#) --> [权限管理](#)

项目	projar	资源	[projar:/] * 选择资源 ▼
用户组	<div> <div>developer</div> <div>manager</div> <div>tester</div> </div>	用户	<div> <div>*</div> <div>ben</div> <div>kent</div> <div>svnadmin</div> </div>
权限	可读可写 ▼		
保存			

### 3.6.2. 设置开发组的权限

输入如下信息：

【资源】: [projar:/trunk]

【用户组】: developer

【权限】: 可读

点击【保存】

[项目管理\(projar\)](#) --> [权限管理](#)

项目	projar	资源	[projar:/trunk] * 选择资源 ▼
用户组	<div> <div>developer</div> <div>manager</div> <div>tester</div> </div>	用户	<div> <div>*</div> <div>ben</div> <div>kent</div> <div>svnadmin</div> </div>
权限	可读 ▼		
保存			

输入如下信息：

【资源】: [projar:/trunk/src]

【用户组】: developer

【权限】: 可读可写



点击【保存】

项目管理(projar) --> 权限管理

项目	projar	资源	[projar:/trunk/src] * 选择资源
用户组	developer manager tester	用户	* ben kent svnadmin
权限	可读可写		
<input type="button" value="保存"/>			

### 3.6.3. 设置测试组的权限

输入如下信息：

【资源】: [projar:/tags]

【用户组】: tester

【权限】: 可读

点击【保存】

项目管理(projar) --> 权限管理

项目	projar	资源	[projar:/tags] * 选择资源
用户组	developer manager tester	用户	* ben kent svnadmin
权限	可读		
<input type="button" value="保存"/>			

结果如下：

NO.	项目	资源	用户组/用户	权限	删除
1	projar	[projar:/tags]	tester	r	<a href="#">删除</a>
2	projar	[projar:/trunk/src]	developer	rw	<a href="#">删除</a>
3	projar	[projar:/trunk]	developer	r	<a href="#">删除</a>
4	projar	[projar:/]	manager	rw	<a href="#">删除</a>

项目经理 Tom 使用 svn://192.168.1.100/projar 检出整个项目，并有读写的权限；

开发人员 Ben 使用 svn://192.168.1.100/projar/trunk 检出项目的 trunk 目录，可以对 trunk/src 读写，但对 trunk/.project 等只能有读的权限；

测试人员 Kent 使用 svn://192.168.1.100/projar/tags 检出项目的 tags 版本目录，只可以读，不能写。

### 3.6.4. 设置权限注意问题

#### 3.6.4.1. 关于资源

【资源】如果不使用[]，则会自动添加。如输入/trunk，则会自动变为[projar:/trunk]

#### 3.6.4.2. 如何设置 \* =

输入如下信息：

【用户组】: 不选！

【用户】: \*

【权限】: 没有权限

## 项目管理(projar) --> 权限管理

项目	projar	资源	[projar:/] * 选择资源 ▼
用户组	developer manager tester ▼	用户	ben kent svnadmin ▼
权限	没有权限 ▼		
<input type="button" value="保存"/>			

关于 svn 目录的权限的设置技巧，已经超出本文的讨论范围，请参考 svn 管理的相关文档。

### 4. http 协议单库实例

参考【svn 协议实例】，不同的地方是

#### 4.1. 创建项目

### 项目管理

项目	projar *	协议	http(单库) ▼ *
路径	d:/repository/projar *		
URL	http://192.168.111.100/projar *		
描述	projar测试		
<input type="button" value="提交"/>			

【协议】: http(单库)

- 例如 URL 值为 http://192.168.1.100/projar，则 apache 配置文件 D:/repository/projar/conf/httpd.conf 中的<Location>生成如下：

*<Location /projar>*

- 例如 URL 值为 http://192.168.1.100/svn/projar，则 apache 配置文件 D:/repository/projar/conf/httpd.conf 中的<Location>生成如下：

*<Location /svn/projar>*

- 例如 URL 值为 http://192.168.1.100/svn/projar/trunk，则 apache 配置文件 D:/repository/projar/conf/httpd.conf 中的<Location>生成如下：

*<Location /svn/projar>*

#### 4.2. 配置 Apache

每次增加一个新的项目时都要执行这步骤，修改项目不用，删除项目时删除 **httpd.conf**

这行配置即可。

到仓库目录打开 D:/repository/projar/conf/httpd.conf，复制

```
Include D:/repository/projar/conf/httpd.conf
```

到%Apache%/conf/httpd.conf 最后面。重启 Apache，访问 http://192.168.1.100/projar。

## 5. http 协议多库实例

参考【svn 协议实例】，不同的地方是

### 5.1. 创建项目

【协议】: http(多库)

### 项目管理

项目	projar *	协议	http(多库) ▼ *
路径	d:/repository/projar *		
URL	http://192.168.111.100/repository/projar *		
描述	projar测试		
<input type="button" value="提交"/>			

这里使用 D:/repository 做为 svn root 目录

### 5.2. 配置 Apache

第一次增加项目时执行这步

到仓库目录打开 D:/repository/httpd.conf，复制

```
Include D:/repository/httpd.conf
```

到 %Apache%/conf/httpd.conf 最后面。重启 Apache，访问 http://192.168.1.100/repository/projar。

## 6. 高级用户

### 6.1. 其他数据库支持

发行包 war 中带有 MYSQL 5、Oracle 10 g、SQL Server 2005 的 jdbc 驱动，如果你使用其他数据库，请把对应的 jdbc 驱动 jar 包复制到发行包中的 WEB-INF/lib 目录，参考【安装运行】-【运行】章节，修改 jdbc.properties 配置文件。并参考【附录】-【数据库表结构】章节建立数据库。(也可以使用 PowerDesigner 导入 MySQL 建表语句后更改 DBMS 生成对应数据库结构)

### 6.2. svn 协议输出配置文件

对于 svn 协议的例子，查看服务器 192.168.1.100 的 D:/repository/projar/conf 目录

#### 6.2.1. svnserve.conf

```
[general]
```

```
anon-access = none
auth-access = write
password-db = passwd
authz-db = authz
[sasl/]
```

#### 6.2.2. passwd

```
[users]
ben=ben
kent=kent
tom=tom
```

#### 6.2.3. authz

```
[aliases]
[groups]
developer=ben
manager=tom
tester=kent

[projar:/tags]
@tester=r

[projar:/trunk/src]
@developer=rw

[projar:/trunk]
@developer=r

[projar:/]
@manager=rw
```

### 6.3. http 协议单库输出配置文件

对于 http 协议单库的例子，查看服务器 192.168.1.100 的 D:/repository/projar/conf 目录

#### 6.3.1. passwd.http

```
ben:{SHA}c2dd682KQ2vkjsIiEdz0T+DfCmQ=
kent:{SHA}CWsg2ZsgkBI3iU9FiUTNF8Vw2F4=
tom:{SHA}loNd2L+nGL1kR8zIevia4Wddrso=
```

#### 6.3.2. httpd.conf

```
#Include D:/repository/projar/conf/httpd.conf
<Location /projar>
    DAV svn
```

```
SVNPath D:/repository/projar
AuthType Basic
AuthName "projar"
AuthUserFile D:/repository/projar/conf/passwd.http
AuthzSVNAccessFile D:/repository/projar/conf/authz
Require valid-user
</Location>
```

### 6.3.3. authz

(同 svn 协议的 authz)

## 6.4. http 协议多库输出配置文件

对于 http 协议多库的例子，查看服务器 192.168.1.100 的 D:/repository 目录

### 6.4.1. passwd.http

(同 http 协议单库的 passwd.http)

### 6.4.2. httpd.conf

```
#Include d:/repository/httpd.conf
<Location /repository/>
    DAV svn
    SVNListParentPath on
    SVNParentPath d:/repository
    AuthType Basic
    AuthName "Subversion repositories"
    AuthUserFile d:/repository/passwd.http
    AuthzSVNAccessFile d:/repository/authz
    Require valid-user
</Location>
RedirectMatch ^(/repository)$ $1/
```

### 6.4.3. authz

(同 svn 协议的 authz)

## 6.5. 导入导出项目配置信息

你懂的！

可能你有很多项目，在页面上输入数据太麻烦，可以根据表结构，使用数据库到导入导出，批量插入数据到数据库里。

然后，只要在 svn admin 页面随意修改一下项目的配置信息，即可把数据库的数据输出到 svn 服务器配置文件。

如在【项目管理】，进入编辑项目信息，点击【提交】，触发修改项目信息的动作，会立刻把数据库的配置输出到配置文件。(你总要点一下，我才知道要重新输出配置把，呵呵！)

## 项目管理

项目	projar *	协议	http(多库) *
路径	d:/repository/projar *		
URL	svn://192.168.111.100/projar *		
描述	projar测试		
<input type="button" value="提交"/>			

### 6.6. svn 和 http 协议转换

#### 6.6.1. svn 转为 http

例如 projar 原来是 svn 协议的，现在要采用 http 协议。

1. 【项目管理】改变【协议】为 http(单库)或 http(多库)
2. 参考【http 协议实例】章节，配置 apache。

#### 6.6.2. http 转为 svn

例如 projar 原来是 http 协议的，现在要采用 svn 协议。

1. 【项目管理】改变【协议】为 svn
2. 参考【svn 协议实例】章节访问。

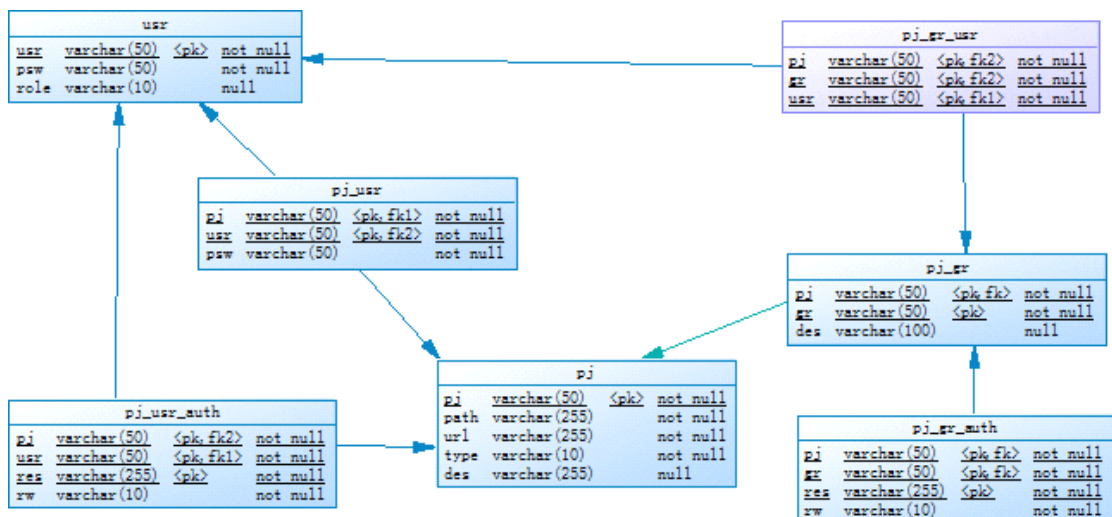
### 6.7. 用户和项目用户

用户是所有项目公用的用户；项目用户只针对 svn 协议或 http 协议(单库)方式的项目，对 http 协议(多库)方式无效。svn 协议或 http 协议(单库)方式的项目可以分别对每个项目的用户设置不同的密码。

## 7. 附录

### 7.1. 数据库表结构

以 MySQL 数据库为例



### 7.2. 版本记录

#### 7.2.1. V 1.0

- 1) 支持 svn 协议配置项目等相关信息。

#### 7.2.2. V 1.0.1

- 1) 修复 bug:SQL Server 版本增加项目时报错。

#### 7.2.3. V 2.0

- 1) 支持 Apache+SVN 配置 http 协议(单库);
- 2) 建立项目时自动创建 svn 仓库;
- 3) 项目权限设置的【资源】如果没有使用[]自动添加;
- 4) 增加【项目管理-增加】权限;
- 5) 修复 bug: 登录账户修改不了数据;
- 6) 从 1.x 升级到 2.x 方法:

◆ 请使用 2.0 版本的 svnadmin.war

◆ 数据库把 pj 表的列 type 改名为 protocol。

可以使用数据库的客户端工具直接修改,也可以使用 SQL 命令。例如 MySQL 数据库执行以下命令:

```
ALTER TABLE `svnadmin`.`pj` CHANGE COLUMN `type` `protocol`  
VARCHAR(10) NOT NULL ;
```

#### 7.2.4. V 2.0.1

- 1) 增加 sortable, 点击表头可以排序。

#### 7.2.5. V 3.0

- 1) 迁移到 google code, 不再兼容 2.x 以下版本, 如需要从低版本升级, 请联系;
- 2) 取消父项目功能;
- 3) 取消登录账户和账户权限功能;
- 4) 下拉列表输入支持多选;
- 5) 支持 Apache+SVN 配置 http 协议(多库)。

#### 7.2.6. V 3.0.1

- 1) [Issue 2: http 多库方式时也把 svn 或 http 单库的项目一起导出;](#)
- 2) [Issue 3: 显示当前的版本;](#)
- 3) [Issue 4: 组取消项目前缀;](#) 如原来的 projar\_manager 变为 manager;
- 4) [Issue 5: http 单库输出 location 去掉 trunk;](#)
- 5) [Issue 6: 项目组设置用户的下拉列表只显示未选的用户;](#)
- 6) [Issue 7: 完善注释 javadoc;](#)
- 7) [Issue 8: 所有可以排序的表头添加提示排序。](#)

### 7.3. 联系方式

项目地址: <http://code.google.com/p/jsvnadmin>

QQ : 56099823

Email: [yuanhuiwu@gmail.com](mailto:yuanhuiwu@gmail.com)

2011-09

( 完 )