

# SVN ADMIN 手册

用于 Svn Admin 3.1.0

## 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 .....	4
2.3. 安装 Java 运行环境 .....	5
2.4. 运行 .....	5
3. svn 协议实例 .....	5
3.1. 需求 .....	5
3.2. 登录 .....	6
3.3. 用户管理 .....	6
3.4. 项目管理 .....	7
3.5. 用户组管理 .....	7
3.6. 设置访问权限 .....	8
3.6.1. 设置开发组的权限 .....	9
3.6.2. 设置测试组的权限 .....	10
3.6.3. 设置权限注意问题 .....	11
4. http 协议单库实例 .....	11
4.1. 创建项目 .....	11
4.2. 配置 Apache .....	12
5. http 协议多库实例 .....	12
5.1. 创建项目 .....	12
5.2. 配置 Apache .....	12
6. 高级用户 .....	13
6.1. 多语言支持 .....	13
6.1.1. 增加语言 .....	13
6.1.2. 翻译 .....	13
6.1.3. 导出语言 .....	15
6.1.4. 选择语言 .....	15
6.2. 多数据库支持 .....	15

6.3. 数据库 H2 管理后台 .....	15
6.4. svn 协议输出配置文件 .....	16
6.4.1. svnserve.conf .....	16
6.4.2. passwd .....	16
6.4.3. authz .....	16
6.5. http 协议单库输出配置文件 .....	17
6.5.1. passwd.http .....	17
6.5.2. httpd.conf .....	17
6.5.3. authz .....	17
6.6. http 协议多库输出配置文件 .....	17
6.6.1. passwd.http .....	17
6.6.2. httpd.conf .....	17
6.6.3. authz .....	18
6.7. 导入导出项目配置信息 .....	18
6.8. svn 和 http 协议转换 .....	18
6.8.1. svn 转为 http .....	18
6.8.2. http 转为 svn .....	18
6.9. 用户和项目用户 .....	18
7. 附录 .....	18
7.1. 数据库表结构 .....	19
7.2. 版本记录 .....	19
7.2.1. V 1.0 .....	19
7.2.2. V 1.0.1 .....	19
7.2.3. V 2.0 .....	19
7.2.4. V 2.0.1 .....	19
7.2.5. V 3.0 .....	19
7.2.6. V 3.0.1 .....	20
7.2.7. V 3.0.2 .....	20
7.2.8. V 3.0.3 .....	20
7.2.9. V 3.0.4 .....	20
7.2.10. V 3.0.5 .....	20
7.2.11. V 3.0.7 .....	21
7.2.12. V 3.1.0 .....	21
7.3. 联系方式 .....	21

## 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）
- ◆ 支持多语言版本。管理员可以登录系统后修改所有显示的标签。（从 3.0.2 版本开始）

### 1.3. 是否适合你

- ◆ Svnadmin 在 Java 1.5、Tomcat 6、Subversion 1.6、MySQL 5.1、Apache 2.2、Windows 上开发测试通过，同时支持 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>

如 Windows 32 位系统，下载

<https://zenlayer.dl.sourceforge.net/project/win32svn/1.8.17/Setup-Subversion-1.8.17.msi>

### Windows



- [CollabNet](#) (client only; supported and certified by [CollabNet](#); *requires registration*)
- [SlikSVN](#) (32- and 64-bit client MSI; maintained by [Bert Huijben](#), [SharpSvn project](#))
- [VisualSVN](#) (client and server; supported and maintained by [VisualSVN](#))
- [WANDisco](#) (32- and 64-bit client and server; supported and certified by [WANDisco](#))
- [Win32Svn](#) (32-bit client, server and bindings, MSI and ZIPs; maintained by [David Darj](#))

## 2.2. 安装 Apache 服务器

(如果你不使用 http 协议，可以不安装 Apache 服务器，略过这个步骤。)

使用 mod\_dav\_svn 可以使用 apache 启动 svn，提供 http 服务

Apache 和 SVN 搭配使用请参考以下链接的【安装 Subversion】章节：

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

主要步骤：

### a. 下载 apache

注意：Win32Svn 需要 32 位版本的，可以下载这个版本

<https://www.apachelounge.com/download/vs17/binaries/httpd-2.4.58-win32-vs17.zip>

### b. 配置 apache

#### 1) 复制 so 文件

从 Subversion 安装目录将 bin 下的 mod\_dav\_svn.so 和 mod\_authz\_svn.so 复制到 Apache 的目录 modules 下

#### 2) 修改 apache 目录 conf 下 httpd.conf

Define SRVROOT 修改为本机 apache 所在路径

Listen 80 修改端口

LoadModule dav\_fs\_module modules/mod\_dav\_fs.so 取消注释#

LoadModule dav\_module modules/mod\_dav.so 取消注释#

下面加多 2 行

LoadModule dav\_svn\_module modules/mod\_dav\_svn.so

LoadModule authz\_svn\_module modules/mod\_authz\_svn.so

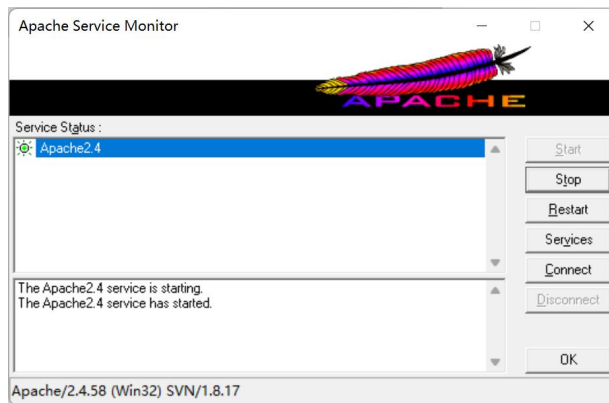
### c. 启动 apache

#### 1) 以管理员身份启动 cmd 命令行窗口，cd 到 apache/bin 目录，执行

httpd -k install 安装服务

#### 2) 以右键以管理员身份打开 ApacheMonitor.exe，启动 apache，会显示 apache 和 svn

版本，绿色表示运行成功



### 2.3. 安装 Java 运行环境

这个简单吧？不会，上网搜索一下：)

使用命令行 `java -version` 打出正确版本号

### 2.4. 运行

将发行包解压，例如版本号 3.1.0，使用命令行运行以下语句：

```
java -jar svnadmin-tomcat-3.1.0.jar
```

控制台显示 tomcat 启动正常消息，浏览器访问 <http://127.0.0.1:8080/svnadmin>，出现登录页面。说明你安装成功。

参数说明：

1. 端口号：默认是 8080 端口，如果要更改，传递第 1 个参数

```
java -jar svnadmin-tomcat-3.1.0.jar 8080
```

2. svnadmin.war 所在目录：默认是和 svnadmin-tomcat-3.1.0.jar 同一个目录，如果要更改位置为/xxx/xxx/svnadmin.war，传递第 2 个参数

```
java -jar svnadmin-tomcat-3.1.0.jar 8080 /xxx/xxx/
```

3. 数据库：默认使用内嵌 H2 数据库，.db 数据库文件保存在当前目录，注意备份。

如果要更改位置为/xxx/xxx/svnadmin.mv.db

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，更改 `jdbc.ur` 参数。

```
jdbc.url=jdbc:h2:/xxx/xxx/svnadmin
```

## 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 仓库目录在服务器本地的 `e:/svn/projar`，访问这个仓库的 svn 地址是：`svn://127.0.0.1/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://127.0.0.1:8080/svnadmin>

登录页面

第一次使用，输入的帐号和密码默认角色为超级管理员

欢迎使用SVN ADMIN, 第一次使用请设置管理员帐号和密码.

帐号

密码

如输入帐号 svnadmin，密码 svnadmin（也可以使用其他用户名和密码），点击【登录】，进入首页。

SVN ADMIN 用户 项目 svnadmin 退出

---

项目管理

项目	<input type="text"/>	协议	svn
路径	<input type="text"/>		
URL	<input type="text"/>		
描述	<input type="text"/>		

### 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	<input type="text" value="tom"/>	*	新密码	<input type="password"/>	角色	<input type="text" value="选择角色"/>	<input type="button" value="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 *	类型	svn *
路径	e:/svn/projar *		
URL	svn://127.0.0.1/projar *		
描述	projar测试		
<input type="button" value="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 仓库的位置：

如果 e:/svn 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】访问 svn 仓库的地址。svn 协议使用 svn://开头。

(如果有多个地址，如内网和外网，可以使用<br>分隔，如：  
svn://127.0.0.1/core/trunk<br>svn://xxx:3690/core/trunk)

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	描述	<input type="button" value="提交"/>
----	--------	----	----	-----------------------------------

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

[manager]组是默认的管理员组，不可以修改或删除。这个组默认具有对项目的根路径 [projar:/] 有可读可写的权限。这个组的用户具有管理这个项目的权限。

点击第一行 developer 的【设置用户】

项目管理(projar) --> 用户组管理(developer)-->项目组用户管理



NO.	项目	用户组	用户	删除
-----	----	-----	----	----

在左边选择 ben，点击【>】（也可以双击左边的 ben），移到右边，表示选择了 ben。使用 Shift 或 Ctrl 可以多选。

选择好后，点击【增加用户】结果如下：

项目管理(projar) --> 用户组管理(developer)-->项目组用户管理



NO.	项目	用户组	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. 设置访问权限

项目管理员 Tom 登录系统可以看到他有权限看到的项目：

(注意：超级管理员不属于任何一个组,所以会看到“认证失败”，除非设定到某个组或权限)



NO.	项目	路径	URL	类型	描述	设置用户	设置用户组	设置权限	删除
1	projar	e:/svn/projar	svn://127.0.0.1/projar	svn	projar测试	设置用户	设置用户组	设置权限	删除

在对应的项目点击 URL 【svn://127.0.0.1/projar】或【设置权限】，进入如下

项目管理(projar) --> 权限管理

左边是仓库浏览器，右边是权限设置区域。

(由于 Tom 属于 manager 组，默认具有对[projar:/]的可读可写权限。)

### 3.6.1. 设置开发组的权限

在左边的仓库浏览器，点击 trunk 目录，目录树会列出 trunk 的文件或文件夹。右边权限设置区域【资源】自动填充[projar:/trunk] (也可以手工输入)，最下面列出这个项目这个路径的[projar:/trunk]的权限设置列表。

输入如下信息：

【资源】：[projar:/trunk]

【用户组】：选择 developer 点击【>】移动右边(或双击)

【权限】：可读

项目管理(projar) --> 权限管理

点击【保存】，保存数据。

同理，在左边仓库浏览器点击 trunk 下面的 src 目录，右边输入以下信息：

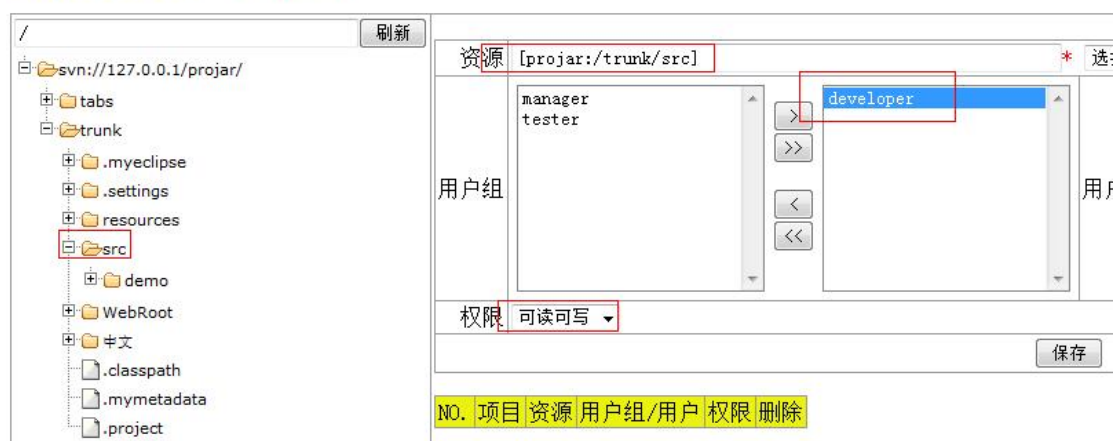
【资源】：[projar:/trunk/src]

【用户组】：developer

【权限】：可读可写

点击【保存】

### 项目管理(projar) --> 权限管理



点击【保存】保存数据。

#### 3.6.2. 设置测试组的权限

同理，在左边仓库浏览器点击 tags 目录，右边输入以下信息：

【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



结果如下：

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

项目经理 Tom 使用 svn://127.0.0.1/projar 检出整个项目，并有读写的权限；

开发人员 Ben 使用 svn://127.0.0.1/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://127.0.0.1/projar/tags 检出项目的 tags 版本目录，只可以读，不能写。

### 3.6.3. 设置权限注意事项

#### 3.6.3.1. 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]

#### 3.6.3.2. 如何设置 \*=

输入如下信息：

【用户组】：不选！

【用户】：\*

【权限】：没有权限

资源 [projar:/] \* [projar:/]

用户组: developer, manager, tester

用户: ben, kent, svnadmin, tom

权限: 没有权限

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 *	类型	http *
路径	e:/svn/projar *		
URL	http://127.0.0.1/projar *		
描述	projar测试		
提交			

【协议】：http(单库)

➤ 例如 URL 值为 http://127.0.0.1/projar，则 apache 配置文件 e:/svn/projar/conf/httpd.conf 中的<Location>生成如下：

<Location /projar>

➤ 例如 URL 值为 http://127.0.0.1/svn/projar，则 apache 配置文件

e:/svn/projar/conf/httpd.conf 中的<Location>生成如下:

<Location /svn/projar>

- 例如 URL 值为 http://127.0.0.1/svn/projar/trunk, 则 apache 配置文件 e:/svn/projar/conf/httpd.conf 中的<Location>生成如下:

<Location /svn/projar>

## 4.2. 配置 Apache

到仓库目录打开 e:/svn/projar/conf/httpd.conf, 复制

*Include e:/svn/projar/conf/httpd.conf*

到%Apache%/conf/httpd.conf 最后面。重启 Apache, 访问 http://127.0.0.1/projar。

(注意: 每次增加一个新的项目时都要执行这步骤后重启, 修改项目不用重启, 删除项目时删除这行)

## 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 *	类型	http(多库) ▼ *
路径	e:/svn/projar *		
URL	http://127.0.0.1/projar *		
描述	projar测试		
<input type="button" value="提交"/>			

这里使用 e:/svn 做为 svn root 目录

### 5.2. 配置 Apache

到仓库目录打开 e:/svn/httpd.conf, 复制

*Include e:/svn/httpd.conf*

到 %Apache%/conf/httpd.conf 最后面。重启 Apache, 访问 http://127.0.0.1/repository/projar。

(注意: 第一次增加项目执行这步骤重启, 修改项目不用重启, 删除项目时删除这行)

## 6. 高级用户

### 6.1. 多语言支持

3.0.2 版本开始支持多语言。管理员才又权限修改多语言功能。多语言功能可以修改显示的文本标签，和提示的消息。(由于多语言使用 web 服务器的缓存，如果手工在数据库客户端直接修改表的记录，需要在页面多语言功能任意修改其中一个标签值，来刷新 web 服务器端的缓存，同步数据库的表记录。所以最好使用页面功能来修改数据。)

#### 6.1.1. 增加语言

管理员登录后，点击【语言】菜单。

在多语言页面，点击【增加语言】，进入以下页面：

[语言管理](#) --> [增加语言](#)

当前语言：zh\_CN - 中文

日文 ▼	
语言	zh_CN *
标签	中文 *
<input type="button" value="提交"/>	

NO.	语言	标签
1	zh_CN	中文

这里显示当前的语言，下方表格显示系统提供的语言列表。

例如要增加英文版本语言，下拉列表中选择【英文】，会自动填充

【语言】：en

【标签】：英文

(也可以手工输入)

点击【提交】，结果如下：

[语言管理](#) --> [增加语言](#)

当前语言：zh\_CN - 中文

日文 ▼	
语言	en *
标签	英文 *
<input type="button" value="提交"/>	

NO.	语言	标签
1	en	英文
2	zh_CN	中文

(注意：打开系统时，默认从浏览器中获取当前的语言)

#### 6.1.2. 翻译

管理员登录后，点击【语言】菜单，进入以下页面：

语言管理 当前语言: zh\_CN - 中文

en	键值
! <a href="#">i18n.add.title</a>	英文(en)
! <a href="#">i18n.current</a>	
! <a href="#">i18n.error.lang</a>	中文(zh_CN)
! <a href="#">i18n.error.lbl</a>	
! <a href="#">i18n.id</a>	
! <a href="#">i18n.lang</a>	
! <a href="#">i18n.lbl</a>	
! <a href="#">i18n.op.addlang</a>	

[增加语言](#) [导出多语言贡献给svnadmin项目组](#)

左边显示所有系统多语言的键值，右边是标签值编辑区域。

例如想修改主页面的标题，在左边列表中找到键值 `main.title`，点击后

语言管理 当前语言: zh\_CN - 中文

en	键值	main.title
! <a href="#">i18n.add.title</a>	英文(en)	
! <a href="#">i18n.current</a>		
! <a href="#">i18n.error.lang</a>	中文(zh_CN)	SVN ADMIN
! <a href="#">i18n.error.lbl</a>		
! <a href="#">i18n.id</a>		
! <a href="#">i18n.lang</a>		
! <a href="#">i18n.lbl</a>		
! <a href="#">i18n.op.addlang</a>		
! <a href="#">i18n.op.export</a>		
! <a href="#">i18n.op.submit</a>		
! <a href="#">i18n.title</a>		
! <a href="#">login.btn.login</a>		
! <a href="#">login.title</a>		
! <a href="#">main.link.i18n</a>		
! <a href="#">main.link.logout</a>		
! <a href="#">main.link.pj</a>		
! <a href="#">main.link.user</a>		
! <a href="#">main.title</a>		

把原来的值“SVN ADMIN”，

英文文(en)修改为新值：“XXX SVN System”

中文(zh\_CN)修改为新值：“XXX 公司 SVN 管理系统”

点击【提交】

结果：

XXX公司SVN管理系统

V 3.0.2

语言管理 当前语言: zh\_CN - 中文

en	键值	main.title
! i18n.add.title	英文(en)	XXX SVN System
! i18n.current	中文(zh_CN)	XXX公司SVN管理系统
! i18n.error.lang		
! i18n.error.lbl		
! i18n.id		
! i18n.lang		
! i18n.lbl		
! i18n.op.addlang		
! i18n.op.export		
! i18n.op.submit		
! i18n.title		
! login.btn.login		
! login.title		
! main.link.i18n		
! main.link.logout		
! main.link.pj		
! main.link.user		
main.title		

提交

增加语言

导出多语言贡献给

(注意: 左边显示的键值, 表示没有翻译完整)

### 6.1.3. 导出语言

希望您能把其他翻译好的语言贡献给开源项目, 尽自己的一份力量。

打开【语言】功能, 点击【导出多语言贡献 svnadmin 项目组】将会导出多语言到一个 i18n.sql 文件, 把这个文件邮寄给 svnadmin 项目联系人员或直接上传到项目中。

对此我们感激不尽!

### 6.1.4. 选择语言

在页面右上角的下拉框显示系统支持的语言。

XXX公司SVN管理系统

V 3.0.2

中文

[用户](#)
[项目](#)
[语言](#)
[svnadmin](#)
[退出](#)

选择【英文】, 结果如下:

XXX SVN System

V 3.0.2

英文

[用户](#)
[项目](#)
[语言](#)
[svnadmin](#)
[退出](#)

## 6.2. 多数据库支持

发行包 war 中带有 MYSQL 5 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)

## 6.3. 数据库 H2 管理后台

管理员点击【数据库】进入 H2 数据库管理界面

SVN ADMIN

V 3.1.0

中文

[用户](#)
[项目](#)
[语言](#)
[数据库](#)
[admin](#)
[退出](#)

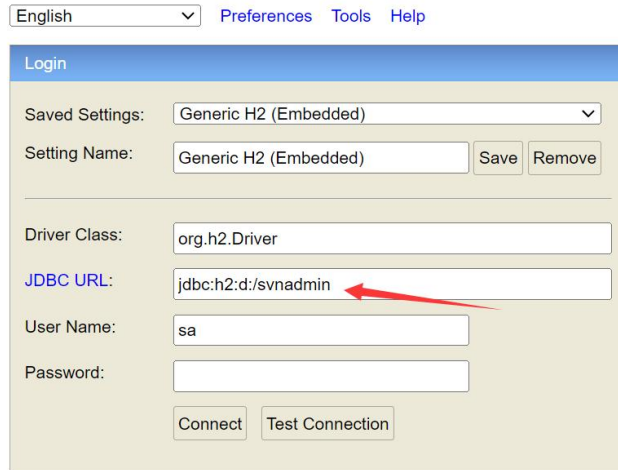


JDBC URL:更改为本机 svnadmin.mv.db, 例如下图是保存在 d:/svnadmin.mv.db

用户名: sa

密码: 空

(和 jdbc.properties 配置文件一致)



#### 6.4. svn 协议输出配置文件

对于 svn 协议的例子, 查看服务器 127.0.0.1 的 e:/svn/projar/conf 目录

##### 6.4.1. svnserve.conf

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

##### 6.4.2. passwd

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

##### 6.4.3. 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

```

## 6.5. 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 127.0.0.1 的 e:/svn/projar/conf 目录

### 6.5.1. passwd.http

```

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

```

### 6.5.2. httpd.conf

```

#Include e:/svn/projar/conf/httpd.conf
<Location /projar>
    DAV svn
    SVNPath e:/svn/projar
    AuthType Basic
    AuthName "projar"
    AuthUserFile e:/svn/projar/conf/passwd.http
    AuthzSVNAccessFile e:/svn/projar/conf/authz
    Require valid-user
</Location>

```

### 6.5.3. authz

（同 svn 协议的 authz）

## 6.6. 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 127.0.0.1 的 e:/svn 目录

### 6.6.1. passwd.http

（同 http 协议单库的 passwd.http）

### 6.6.2. httpd.conf

```

#Include e:/svn/httpd.conf
<Location /repository/>
    DAV svn
    SVNListParentPath on
    SVNParentPath e:/svn

```

```

AuthType Basic
AuthName "Subversion repositories"
AuthUserFile e:/svn/passwd.http
AuthzSVNAccessFile e:/svn/authz
Require valid-user
</Location>
RedirectMatch ^(/repository)$ $1/

```

### 6.6.3. authz

(同 svn 协议的 authz)

### 6.7. 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 *	类型	http(多库) *
路径	e:/svn/projar *		
URL	http://127.0.0.1/projar *		
描述	projar测试		
提交			

### 6.8. svn 和 http 协议转换

#### 6.8.1. 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。

#### 6.8.2. 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。

### 6.9. 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



- 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。](#)

#### 7.2.7. V 3.0.2

- 1) [Issue 10: 多语言版本](#)
- 2) [Issue 11: 用户管理的列表显示的序号不正确](#)
- 3) [Issue 12: 进入系统前验证数据库是否连接成功](#)
- 4) [Issue 14: 支持 SVN 中文目录](#)
- 5) [Issue 16: 权限设置小问题](#)
- 6) [Issue 18: 用户名不正确错误提示](#)
- 7) [Issue 21: ORA-00942 表或视图不存在](#)
- 8) [Issue 22: 提供英文版本多语言的 sql 脚本](#)
- 9) [Issue 23: 增加仓库浏览并设置权限功能](#)
- 10) [Issue 24: 发布 3.0.2 版本](#)
- 11) [Issue 25: 保存项目时报错 ORA-00911 无效字符](#)
- 12) [Issue 26: 保存项目如果报错返回后表单没有保留之前的录入数据](#)
- 13) [Issue 27: 提供多语言导出功能](#)
- 14) [Issue 28: \[退出\]显示为乱码](#)
- 15) [Issue 29: 增加项目管理员组默认具有项目根路径的可读可写权限](#)

从 3.0 或 3.0.1 升级到 3.0.2:

1. 使用 3.0.2 的 svnadmin.war
2. 升级数据库:

可以在发行包得 db/update 目录找到对应的升级 sql 脚本, 例如 mysql5 数据库对应的事 mysql5-3.0.x update to 3.0.2.sql, 在客户端执行就可以升级数据库到新版本。

#### 7.2.8. V 3.0.3

- 1) [Issue 31: 权限设置自动加上项目的 id 不正确](#)
- 2) [Issue 32: 新建新仓库及导入旧仓库中含有大写字母支持性](#)
- 3) [Issue 34: url 包含中文时在权限设置左边的树的根节点显示乱码](#)

#### 7.2.9. V 3.0.4

- 1) [Issue 38: MySQL5 ERROR 1071 max key length is 1000 bytes](#)
- 2) [Issue 39: 支持 SVN 1.7 版本](#)

#### 7.2.10. V 3.0.5

- 1) [Issue 43: 用户管理增加查看所有权限](#)

- 2) [Issue 44: 修复跨项目授权](#)

#### 7.2.11. V 3.0.7

- 1) 增加用户姓名

#### 7.2.12. V 3.1.0

- 1) 默认使用内置数据库 H2
- 2) 默认使用 tomcat-embed 启动应用
- 3) 更改样式

### 7.3. 联系方式

项目地址: <https://github.com/jsvnadmin/jsvnadmin>

QQ : 56099823

QQ 群: 241488036

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

( 完 )