**SVN ADMIN 手册**

用于Svn Admin 3.1.0

**目录**

[目录 1](#_Toc17735)

[1. 简介 3](#_Toc9188)

[1.1. 什么是Svn Admin 3](#_Toc6947)

[1.2. 有什么优点 3](#_Toc6807)

[1.3. 是否适合你 3](#_Toc20821)

[2. 安装运行 3](#_Toc8121)

[2.1. 安装Svn服务器端 4](#_Toc26342)

[2.2. 安装Apache服务器 4](#_Toc7106)

[2.3. 安装Java 运行环境 5](#_Toc14772)

[2.4. 运行 5](#_Toc22330)

[3. svn协议实例 5](#_Toc20931)

[3.1. 需求 5](#_Toc32074)

[3.2. 登录 6](#_Toc11266)

[3.3. 用户管理 6](#_Toc17378)

[3.4. 项目管理 7](#_Toc10328)

[3.5. 用户组管理 7](#_Toc28194)

[3.6. 设置访问权限 8](#_Toc7412)

[3.6.1. 设置开发组的权限 9](#_Toc17118)

[3.6.2. 设置测试组的权限 10](#_Toc24881)

[3.6.3. 设置权限注意问题 11](#_Toc8312)

[4. http协议单库实例 11](#_Toc29840)

[4.1. 创建项目 11](#_Toc24686)

[4.2. 配置Apache 12](#_Toc19096)

[5. http协议多库实例 12](#_Toc4155)

[5.1. 创建项目 12](#_Toc19100)

[5.2. 配置Apache 12](#_Toc29325)

[6. 高级用户 13](#_Toc19118)

[6.1. 多语言支持 13](#_Toc4827)

[6.1.1. 增加语言 13](#_Toc18262)

[6.1.2. 翻译 13](#_Toc18132)

[6.1.3. 导出语言 15](#_Toc26043)

[6.1.4. 选择语言 15](#_Toc5200)

[6.2. 多数据库支持 15](#_Toc31845)

[6.3. 数据库H2管理后台 15](#_Toc26524)

[6.4. svn协议输出配置文件 16](#_Toc32451)

[6.4.1. svnserve.conf 16](#_Toc9565)

[6.4.2. passwd 16](#_Toc18960)

[6.4.3. authz 16](#_Toc15459)

[6.5. http协议单库输出配置文件 17](#_Toc1626)

[6.5.1. passwd.http 17](#_Toc16150)

[6.5.2. httpd.conf 17](#_Toc9390)

[6.5.3. authz 17](#_Toc2417)

[6.6. http协议多库输出配置文件 17](#_Toc3000)

[6.6.1. passwd.http 17](#_Toc24480)

[6.6.2. httpd.conf 17](#_Toc20061)

[6.6.3. authz 18](#_Toc4794)

[6.7. 导入导出项目配置信息 18](#_Toc25616)

[6.8. svn和http协议转换 18](#_Toc25656)

[6.8.1. svn转为http 18](#_Toc22775)

[6.8.2. http转为svn 18](#_Toc26119)

[6.9. 用户和项目用户 18](#_Toc23582)

[7. 附录 18](#_Toc17526)

[7.1. 数据库表结构 19](#_Toc1090)

[7.2. 版本记录 19](#_Toc4177)

[7.2.1. V 1.0 19](#_Toc10240)

[7.2.2. V 1.0.1 19](#_Toc4819)

[7.2.3. V 2.0 19](#_Toc19559)

[7.2.4. V 2.0.1 19](#_Toc17005)

[7.2.5. V 3.0 19](#_Toc1350)

[7.2.6. V 3.0.1 20](#_Toc26218)

[7.2.7. V 3.0.2 20](#_Toc15526)

[7.2.8. V 3.0.3 20](#_Toc10479)

[7.2.9. V 3.0.4 20](#_Toc17228)

[7.2.10. V 3.0.5 20](#_Toc4603)

[7.2.11. V 3.0.7 21](#_Toc23111)

[7.2.12. V 3.1.0 21](#_Toc31560)

[7.3. 联系方式 21](#_Toc9672)

1. 简介
   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. 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. 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Aapche，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。

1. 安装运行

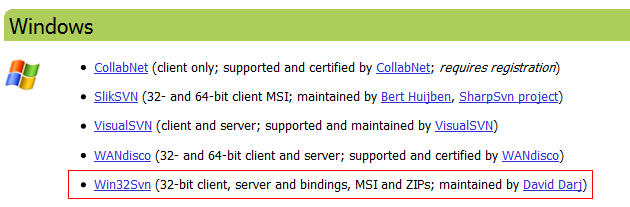
如果你是升级版本，请参考【附录】【版本记录】章节。

* 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>



* 1. 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搭配使用请参考参考以下链接的【安装 Subversion】章节：

[http://www.subversion.org.cn/tsvndoc/tsvn-serversetup.html#tsvn-serversetup-apache](http://www.subversion.org.cn/tsvndoc/tsvn-serversetup.html" \l "tsvn-serversetup-apache)

主要步骤：

1. 下载Apache

注意：Win32Svn需要32位版本的Apache，可以下载这个版本

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

1. 配置Apache
2. 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

1. 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

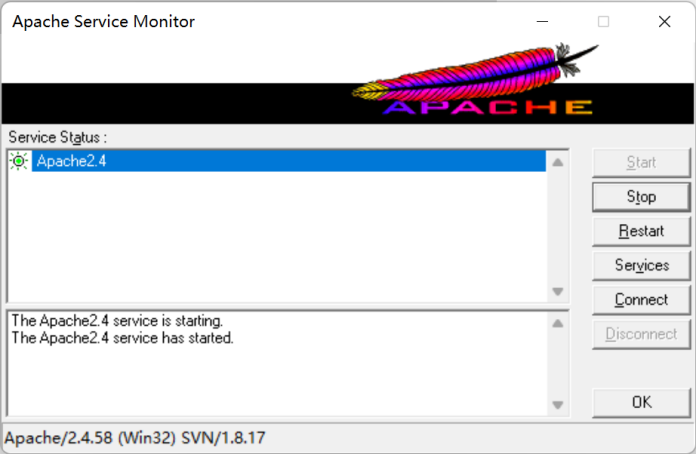
LoadModule dav\_svn\_module modules/mod\_dav\_svn.so

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

1. 启动Apache
2. 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

1. 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



* 1. 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

* 1. 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](http://192.168.1.100:8080/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

1. 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/

1. 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

1. svn协议实例
   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

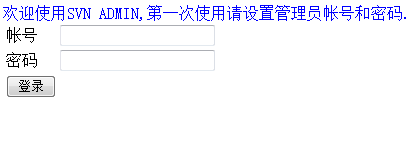
假定项目经理是Tom，有一个开发人员Ben，一个测试人员Kent。你希望Tom有管理这个项目目录的所有权限；Ben可以检出trunk目录，但只能对trunk/src/目录进行写的权限；Kent只能读项目的tags，检出发行版本进行测试。

* 1. 登录

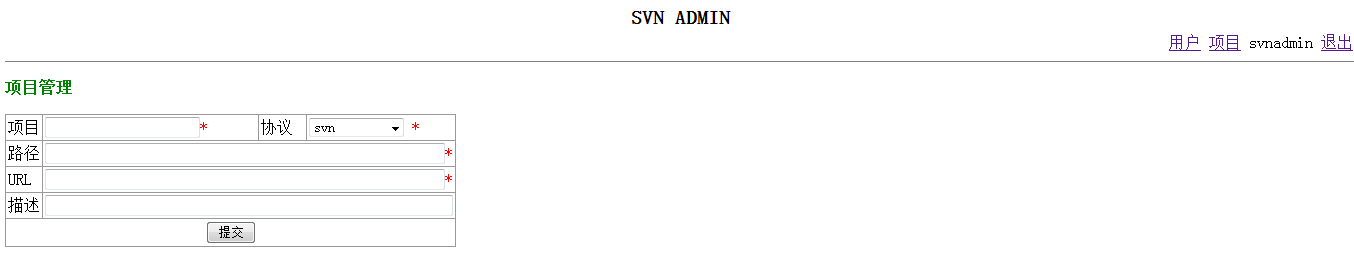
浏览器打开[http://127.0.0.1:8080/svnadmin](http://192.168.1.100:8080/svnadmin)

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



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。



* 1. 用户管理

点击右上角菜单【用户】，输入Tom信息，点击【提交】

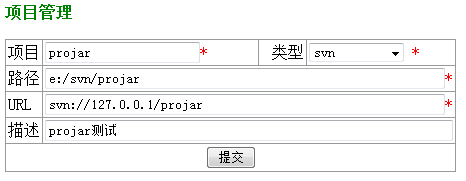


同理：分别添加用户kent和ben。密码都加密保存，结果如下：



* 1. 项目管理

输入项目的基本信息，如下：



【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。

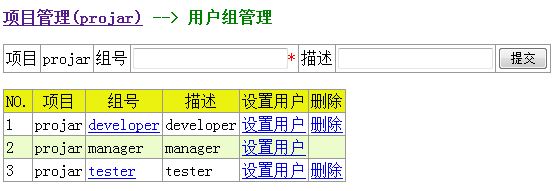
*（如果有多个地址，如内网和外网，可以使用<br>分隔，如：*

*svn://127.0.0.1/core/trunk<br>svn://xxx:3690/core/trunk）*

点击【提交】

* 1. 用户组管理

点击【设置用户组】，进入如下



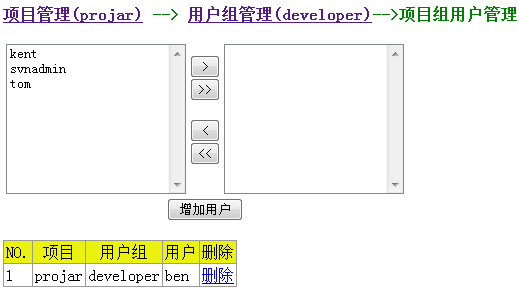
*[manager]组是默认的管理员组，不可以修改或删除。这个组默认具有对项目的根路径[projar:/]有可读可写的权限。这个组的用户具有管理这个项目的权限。*

点击第一行developer的【设置用户】



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:



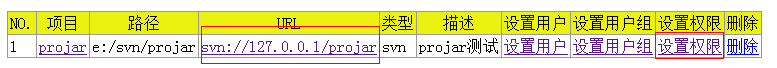
同理：分别添加tom到管理员组manager，添加kent到测试组tester。

*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。*

* 1. 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*)



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



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。)*

* + 1. 设置开发组的权限

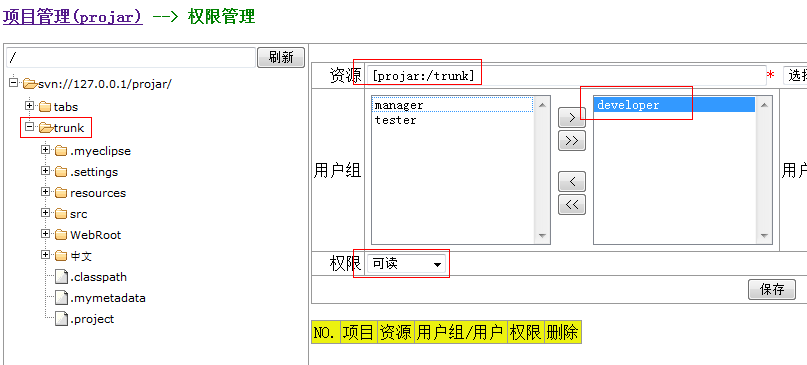
在左边的仓库浏览器，点击trunk目录，目录树会列出trunk的文件或文件夹。右边权限设置区域【资源】自动填充[projar:/trunk] *(也可以手工输入)*，最下面列出这个项目这个路径的[projar:/trunk]的权限设置列表。

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



点击【保存】，保存数据。

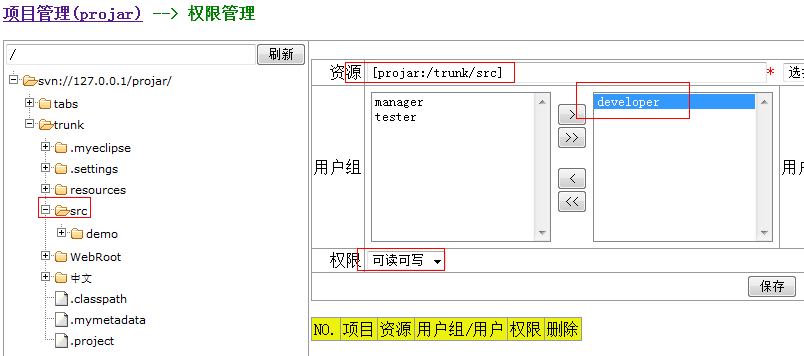
同理，在左边仓库浏览器点击trunk下面的src目录，右边输入以下信息：

【资源】：[projar:/trunk/src]

【用户组】：developer

【权限】:可读可写

点击【保存】



点击【保存】保存数据。

* + 1. 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】



结果如下：



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。

* + 1. 设置权限注意问题
       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]

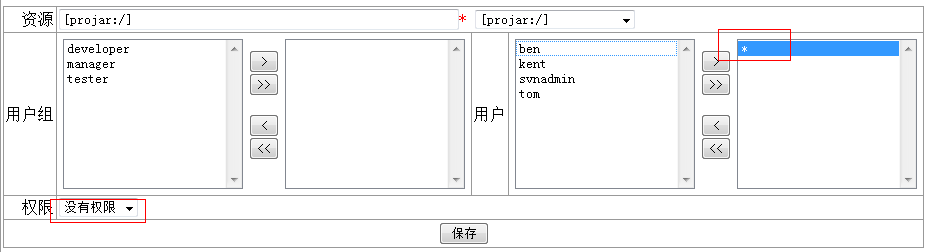
* + - 1. 如何设置 **\*=**

输入如下信息：

【用户组】：不选！

【用户】：\*

【权限】:没有权限

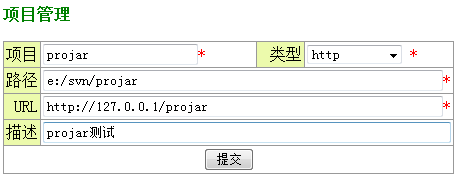


*关于svn目录的权限的设置技巧，已经超出本文的讨论范围，请参考svn管理的相关文档。*

1. http协议单库实例

参考【svn协议实例】，不同的地方是

* 1. 创建项目



【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>*

* 1. 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。

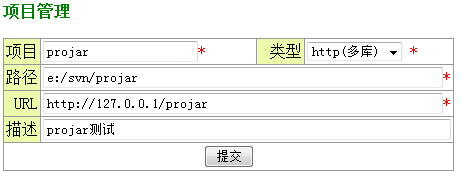
**（注意：每次增加一个新的项目时都要执行这步骤后重启，修改项目不用重启，删除项目时删除这行）**

1. http协议多库实例

参考【svn协议实例】，不同的地方是

* 1. 创建项目

【协议】：http(多库)



这里使用e:/svn做为svn root目录

* 1. 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）**

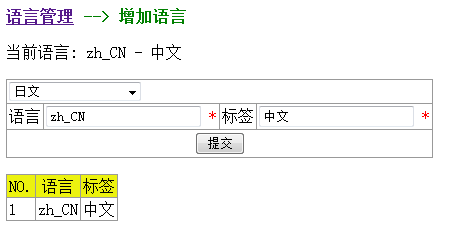
1. 高级用户
   1. 多语言支持

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。)

* + 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：



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）

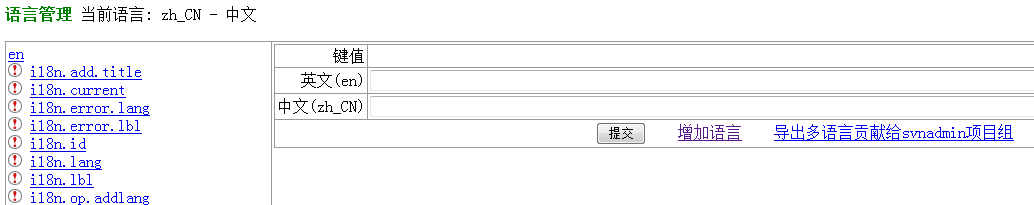
点击【提交】，结果如下：



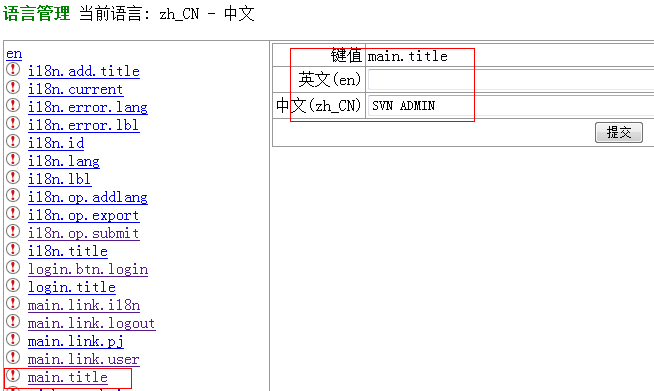
*（注意：打开系统时，默认从浏览器中获取当前的语言）*

* + 1. 翻译

管理员登录后，点击【语言】菜单，进入以下页面：

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



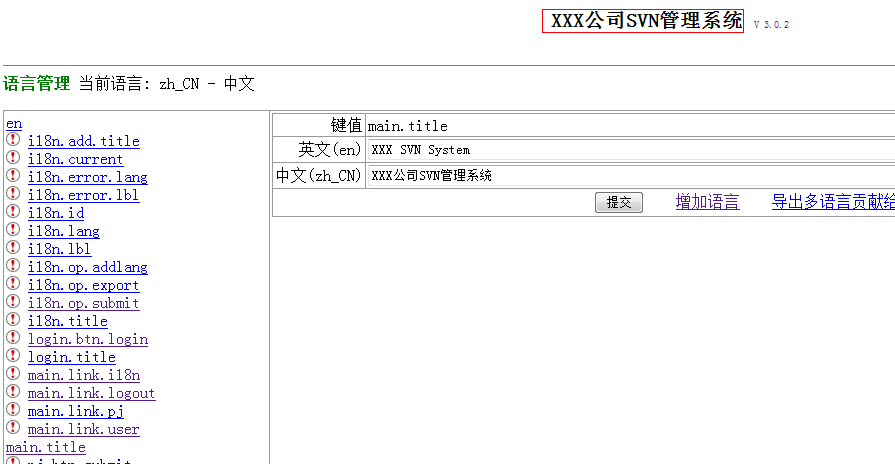
把原来的值“SVN ADMIN”，

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：



*(注意：左边显示的键值，表示没有翻译完整)*

* + 1. 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！

* + 1. 选择语言

在页面右上角的下拉框显示系统支持的语言。



选择【英文】，结果如下：



* 1. 多数据库支持

发行包war中带有H2，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)

* 1. 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

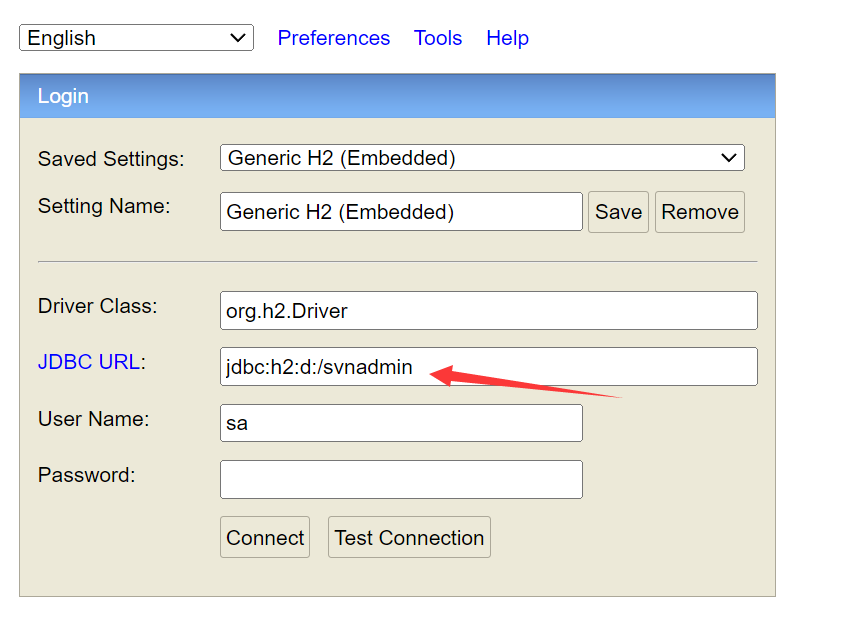


JDBC URL:更改为本机svnadmin.mv.db，例如下图是保存在d:/svnadmin.mv.db

用户名：sa

密码：空

（和jdbc.properties配置文件一致）



* 1. 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

* + 1. svnserve.conf

*[general]*

*anon-access = none*

*auth-access = write*

*password-db = passwd*

*authz-db = authz*

*[sasl]*

* + 1. passwd

*[users]*

*ben=ben*

*kent=kent*

*tom=tom*

* + 1. 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*

* 1. 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

* + 1. passwd.http

*ben:{SHA}c2dd682KQ2vkjsIiEdz0T+DfCmQ=*

*kent:{SHA}CWsg2ZsgkBI3iU9FiUTNF8Vw2F4=*

*tom:{SHA}loNd2L+nGL1kR8zIevia4Wddrso=*

* + 1. 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>*

* + 1. authz

（同svn协议的authz）

* 1. 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

* + 1. passwd.http

（同http协议单库的passwd.http）

* + 1. 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/*

* + 1. authz

（同svn协议的authz）

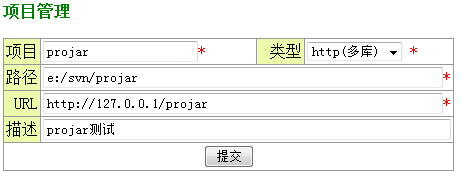
* 1. 导入导出项目配置信息

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！)



* 1. svn和http协议转换
     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。
   * 1. http转为svn

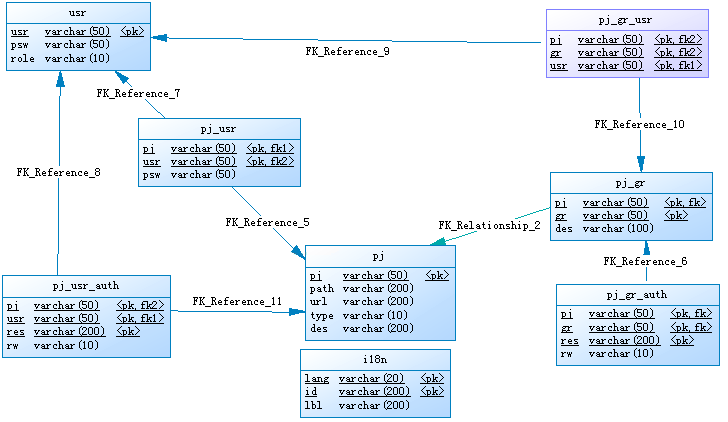
例如projar原来是http协议的，现在要采用svn协议。

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

1. 附录
   1. 数据库表结构

以MySQL数据库为例



* 1. 版本记录
     1. V 1.0

1. 支持svn协议配置项目等相关信息。
   * 1. V 1.0.1
2. 修复bug:SQL Server版本增加项目时报错。
   * 1. V 2.0
3. 支持Apache+SVN配置http协议(单库)；
4. 建立项目时自动创建svn仓库；
5. 项目权限设置的【资源】如果没有使用[]自动添加；
6. 增加【项目管理-增加】权限；
7. 修复bug：登录账户修改不了数据；
8. 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 ;*

* + 1. V 2.0.1

1. 增加sortable，点击表头可以排序。
   * 1. V 3.0
2. 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；
3. 取消父项目功能；
4. 取消登录账户和账户权限功能；
5. 下拉列表输入支持多选；
6. 支持Apache+SVN配置http协议(多库)。
   * 1. V 3.0.1
7. Issue 2: [http多库方式时也把svn或http单库的项目一起导出](http://code.google.com/p/jsvnadmin/issues/detail?id=2)；
8. Issue 3: [显示当前的版本](http://code.google.com/p/jsvnadmin/issues/detail?id=3)；
9. Issue 4: [组取消项目前缀](http://code.google.com/p/jsvnadmin/issues/detail?id=4)；如原来的projar\_manager变为manager；
10. Issue 5: [http单库输出location去掉trunk](http://code.google.com/p/jsvnadmin/issues/detail?id=5)；
11. Issue 6: [项目组设置用户的下拉列表只显示未选的用户](http://code.google.com/p/jsvnadmin/issues/detail?id=6)；
12. Issue 7 : [完善注释javadoc](http://code.google.com/p/jsvnadmin/issues/detail?id=7)；
13. Issue 8 : [所有可以排序的表头添加提示排序](http://code.google.com/p/jsvnadmin/issues/detail?id=8)。
    * 1. V 3.0.2
14. [Issue 10: 多语言版本](http://code.google.com/p/jsvnadmin/issues/detail?id=10)
15. [Issue 11: 用户管理的列表显示的序号不正确](http://code.google.com/p/jsvnadmin/issues/detail?id=11)
16. [Issue 12: 进入系统前验证数据库是否连接成功](http://code.google.com/p/jsvnadmin/issues/detail?id=12)
17. [Issue 14: 支持SVN中文目录](http://code.google.com/p/jsvnadmin/issues/detail?id=14)
18. [Issue 16: 权限设置小问题](http://code.google.com/p/jsvnadmin/issues/detail?id=16)
19. [Issue 18: 用户名不正确错误提示](http://code.google.com/p/jsvnadmin/issues/detail?id=18)
20. [Issue 21: ORA-00942 表或视图不存在](http://code.google.com/p/jsvnadmin/issues/detail?id=21)
21. [Issue 22: 提供英文版本多语言的sql脚本](http://code.google.com/p/jsvnadmin/issues/detail?id=22)
22. [Issue 23: 增加仓库浏览并设置权限功能](http://code.google.com/p/jsvnadmin/issues/detail?id=23)
23. [Issue 24: 发布3.0.2版本](http://code.google.com/p/jsvnadmin/issues/detail?id=24)
24. [Issue 25: 保存项目时报错ORA-00911 无效字符](http://code.google.com/p/jsvnadmin/issues/detail?id=25)
25. [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](http://code.google.com/p/jsvnadmin/issues/detail?id=26)
26. [Issue 27: 提供多语言导出功能](http://code.google.com/p/jsvnadmin/issues/detail?id=27)
27. [Issue 28: [退出]显示为乱码](http://code.google.com/p/jsvnadmin/issues/detail?id=28)
28. [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](http://code.google.com/p/jsvnadmin/issues/detail?id=29)

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。

* + 1. V 3.0.3

1. [Issue 31: 权限设置自动加上项目的id不正确](http://code.google.com/p/jsvnadmin/issues/detail?id=31)
2. [Issue 32: 新建新仓库及导入旧仓库中含有大写字母支持性](http://code.google.com/p/jsvnadmin/issues/detail?id=32)
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](http://code.google.com/p/jsvnadmin/issues/detail?id=34)
   * 1. V 3.0.4
4. [Issue 38: MySQL5 ERROR 1071 max key length is 1000 bytes](http://code.google.com/p/jsvnadmin/issues/detail?id=38)
5. [Issue 39: 支持SVN 1.7版本](http://code.google.com/p/jsvnadmin/issues/detail?id=39)
   * 1. V 3.0.5
6. [Issue 43: 用户管理增加查看所有权限](http://code.google.com/p/jsvnadmin/issues/detail?id=43)
7. [Issue 44: 修复跨项目授权](http://code.google.com/p/jsvnadmin/issues/detail?id=44)
   * 1. V 3.0.7
8. 增加用户姓名
   * 1. V 3.1.0
9. 默认使用内置数据库H2
10. 默认使用tomcat-embed启动应用
11. 更改样式
    1. 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 )