# 开发日志

## 1、数据采集工具在执行任务报错：

###### 1.1、权限不够：给对应的数据库赋权限，在对应的模式下打开命令行执行：

grant all on schema 模式名 to 用户名；

grant all on all tables in shcema 模式名 to 用户名;

###### 1.2、XXX 视图或表 not exists ：在中间库或融合库下的dpksh模式下缺少依赖的表或视图：

重新导入表或视图

###### 1.3、执行关于金钱任务的跑数任务时报错：integer of out range：

将该任务对应的表的对应金钱字段从 int4 修改为 numeric 类型，由于plsql没有numeric 类型，修改为decimal 类型 ，保存后会变为numeric 类型。

1.3.1、在修改的过程中可能会遇到报错内容为依赖视图存在无法修改：

将所有的依赖视图的语句先拷贝下来并删除，修改后重新执行视图语句

1.3.2、需要将gp库下的hyrhk模式和中间库的dpzjk模式下的使用到的表全部修改

###### 1.4 执行任务出错，空间不足

执行任务的所在的硬盘满了，导致的原因可能是因为没有升级数据采集工具

任务不会自动删除执行的任务，解决方法，删除所在路径冗余的数据

## 2、在使用指标计算平台的报错

###### 2.1、新建指标报错：执行sql语句错误：

需要将sql语句放到gp库下的ajzj模式下执行一遍，如果有错误则修改；如果没有错误则检查关联表是否都有”模式名”.”表名”，若没有则需要添加

###### 2.2、导入任务无法初始化：

1、在将任务部署到现场的过程中需要将任务导出、并且导入到现场的指标计算平台中，在修改数据源之后无法初始化任务：平台的bug，不能将修改数据源，需要将导入的数据源的ip地址修改为现场的ip地址即可。

2、将任务中涉及到的指标和任务重建，重新初始化和移送

###### 2.3、现场在修改计算任务和指标的时候会将关联的表和视图删除

###### 2.4、由于指标计算平台的问题导致任务表和视图无法创建或删除

1、首先向开发反映问题

2、可以选择手动创建或删除表和视图

###### 2.5查看指表计算平台的配置

1、查看指标计算平台的服务器地址（pwd:6789@jkl）🡪进入隐藏目录（cd /data）🡪进入TAS8080文件夹 🡪查看jvm.ini(cat jvm.ini)🡪获取服务器配置地址🡪使用管理员用户名和密码登录（username: zbjs; pwd:123 ）

2、向指标计算平台的开发人员进行求助

## 3、python Script

###### 3.1、写脚本的时候需要考虑到数据为None(null)的情况 和‘’的情况

###### 3.2、在给标签置0或‘’之后 判断 是否为‘’ 就可以

###### 3.3、清除缓存脚本

setDataDigits("","总数");#从有数据切换到无数据的时候会保留上一次的数据。通过这个来清空数据缓存-陈真凤

setDataDigits( "","新收");

setDataDigits( "","旧存");

setDataDigits("" ,"已结");

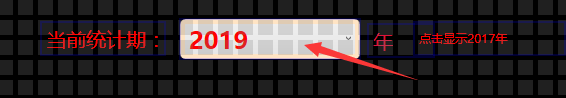
setDataDigits("" ,"未结");

###### 3.4、使用python脚本获取当前时间

## 4、设计器使用

###### 1、根据年限选择统计期，并展示对应年份的数据

1.1、用到了新的控件“选择框”(下拉选项框)



1.2选项值采用key-value的选项存储的，可以设置默认项索引（输入序号从0开始）

1.3赋值：

1.3.1 如果选项不会变化可以直接输入选项的key 和value

1.3.2 选项会变化可以采用绑定数据集的方式获取选项

拟定可以选择近三年，故使用数据集赋值；数据集不需要查询表直接进行结果查询即可

数据集sql： select (EXTRACT(year from now())-2)::varchar as y ,'0' as i

union all

select (EXTRACT(year from now())-1)::varchar as y ,'1' as i

union all

select (EXTRACT(year from now()))::varchar as y ,'2' as i

1.4获取选项值：

绑定在选择框控件 选择项改变

Python脚本：

from VisualDiagram.Event import \*

from VisualDiagram.Utility import \*

from System.Windows.Media import \*

def getSelectedValue():

#获取当前选择的值

value=this['xlk'].GetProperty("SelectedName");

#弹出当前选择的值

MessageBox.Show(value);

def SelectDemo()：

#给选择框选择序号为0的选项

   this["选择框1"].SelectedIndex=0

1.5 sql传参查询

demoSQL:

select

columnvalue5 as year,

columnvalue11 as lx,

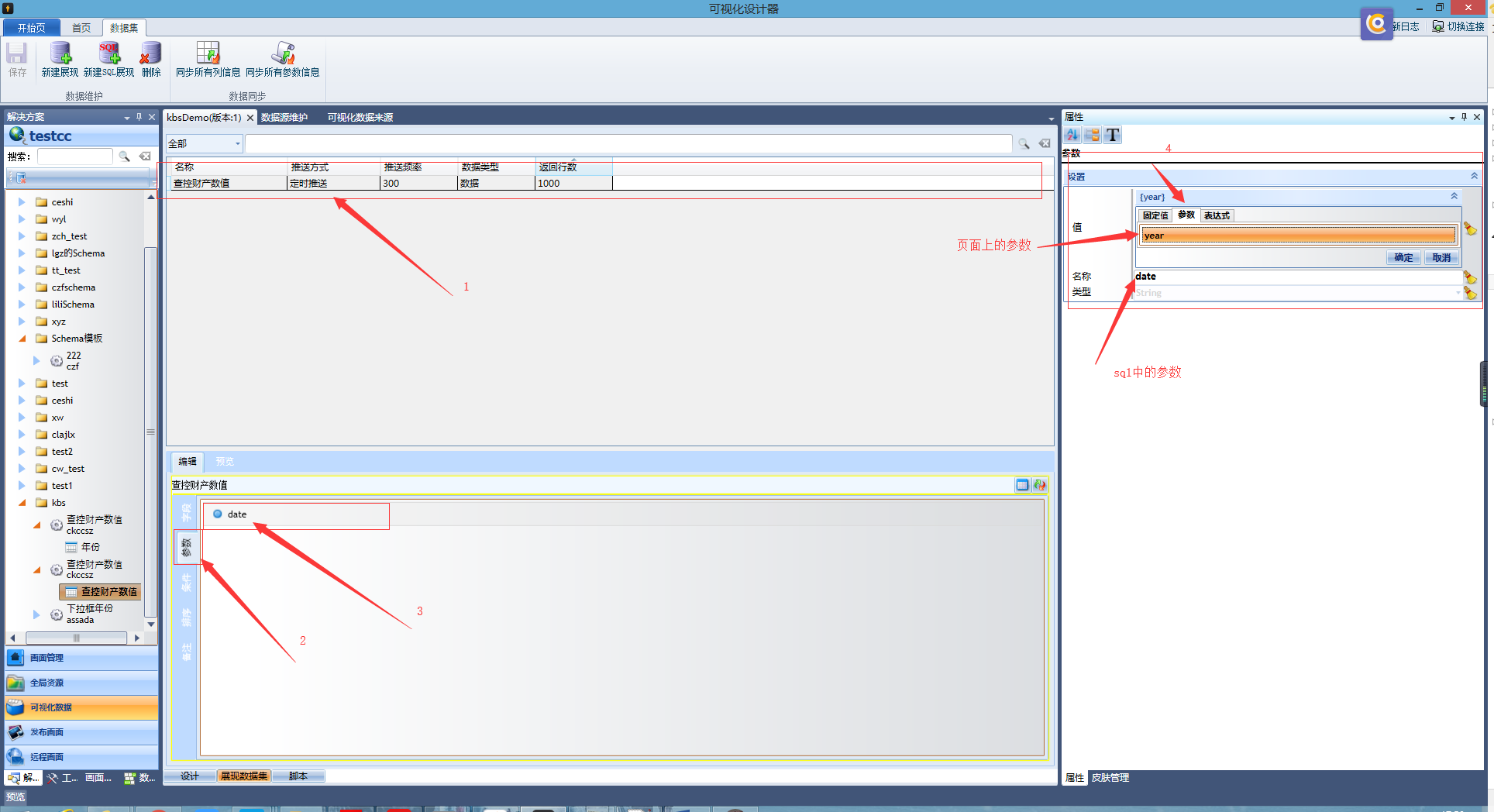
columnvalue12 as value

from db\_cc.z\_visu\_kbs

where schemaid='ckccsz' and columnvalue5='${date}'

参数date作为查询条件，这样一次性读取出来的数据量少，设计器不会卡顿

1.6将带参数的数据集绑定在控件上面，并将页面参数传递到sql的参数中



将sql中有参数的数据集放到展现数据集中🡪选中参数🡪显出sql中定义的参数

🡪选中右侧会显示出设置栏🡪设置参数，将页面参数和sql中的参数进行对应。

🡪在给页面参数赋值后数值也会随之改变

1.7在柱图刷新数据的时候，遇到了数据库中没有的年份，则会遗留上次选择有年份的数据，即没有年份无法刷新。故某年没有数据的正确表现形式为数据库数据库中存在某年 但是数据列为null。

## 5、部署数据库

###### 1、还原数据库时报错：

重新打包一个包发过来然后重新还原成功

###### d 2、备份postgresql数据库：

在本地的\thunisoft\arterybase\3.5\bin目录下Shift+鼠标右击🡪选择在此处打开命令行窗口🡪输入命令：pg\_dump -h localhost -U username databasename > filename.bak

命令含义： -h:要备份的数据库所在的服务器地址（localhost）

-U:用户名(username)

要备份的数据库名称（databasename）

文件名称(filename.bak)

🡪回车后输入密码，不报错即可成功

###### 3、还原postgresql数据库：

详情大屏现场部署文档

## 6、开启服务遇到问题

###### 1、查看license是否过期

使用的是正版过期找恒泰重新申请新的license

使用的是临时版本（只有一天期限），删除licence文件及缓存重新启动

###### 2、webcontext实例化失败（河南高院）

查看tomcat启动的进程 (命令： ps -ef|grep tomcat )

是否有重复启动的tomcat，如果有杀掉所有的该tomcat的进程（命令: kill -s 9 id）

然后重新启动

###### 3、webcontext实例化失败（洛阳中院）

问题排查过程：

1、liscence查看有没有过期，没有过期

2、 查看是否有重复启动的tomcat：没有

查看方法：在Xshell输入命令：ps -ef|grep tomcat 会列出所有正在运行的tomcat的进程

3、查看是端口是否被占用：没有

查看方法：在Xshell输入命令： netstat -tunlp 会列出正在使用的端口

4、怀疑tomcat和visu的包出现问题，从217服务器上下载下来山东高院的包，重新修改配置，启动失败，山东高院的包是可以正常启动的

5、将修改之后的里面的数据库地址改成山东高院的地址，启动成功，改成洛阳的数据库地址，启动失败。得出结论：洛阳的数据库出现了问题

6、重新恢复洛阳的数据库

7、在本地启动成功

8、去服务器启动原来tomcat，启动成功。问题解决

9、大屏无法访问可视化数据，修改数据源。问题解决

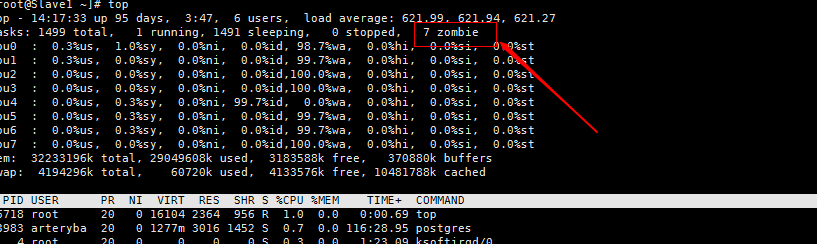
###### 4、服务器无法启动大屏服务和备份数据库

1、场景：大屏服务无法启动，试了几次，输入命令之后会卡死。

数据库无法备份。

2、查看进程（:ps -ef|grep tomcat）显示有进程。

3、查看内存和进程使用情况(:top)



发现僵尸进程和孤儿进程（defunct）

4、猜想可能是因为这个原因造成服务器不能正常使用

使用各种方法杀掉僵尸进程和孤儿进程，无效

僵尸进程需要寻找到它的父进程的id，才能kill掉。在此处无效。

5、孤儿进程无法kill掉只能重新启动服务器

6、打算重启服务器，输入各种重启和关机命令均无效，会卡在命令执行的界面

7、使用虚拟机控制重启，经过寻找虚拟机的登录帐号和密码错误

8、经过百度查询强制重启方法后重启成功。

9、成功解决

10、启动arterybase，需要切换到到普通用户，查找所有用户，切换到arterybase用户，命令:su username。服务启动成功，大屏服务启动成功，数据库备份成功，服务器恢复正常

## 7.sql Script

## 8.网络问题/软件问题

###### 1、关于teamViewer13连接超时的解决办法

a.使用电脑管家或360卸载teamViewer13软件

b.修改MAC地址（network address）设备管理器🡪网卡🡪属性🡪网络地址修改为12个字符的地址

注意：一定要记住本机的mac地址，在命令行中 ipconfig -all 的物理地址为mac地址

c.修改完mac地址后，重新安装teamViewer13.

d.安装完成之后，将mac地址修改为原来的地址

00-23-24-E8-F6-EE

## 9.常用liunx命令

1、cat XXX：查看某个文件内容

2、cd XXX:移动到某个目录下

3、vi XXX：使用vi编辑器编辑某个文件

4、netstat -tunlp:查看占用的端口

5、sudo：以系统管理者的身份执行指令

6、echo’message’ > test.txt :向文件中写入信息

7、ps -ef 显示正在运行的所有的进程 ps -ef|grep tomcat:筛选tomcat的进程

## 10.Git学习日志

1.git的三种状态

已提交（committed）--已经保存到本地的数据库

已修改(modified) --文件已经修改未存到数据库

已暂存(staged)—表示对一个已经修改的文件做了标记，使之包含在下次提交的快照之中

2.git的三个工作区域

Git仓库(git resource)

用户的的工作目录(user resource)

暂存区域(stage )

3.git的基本工作流程

在用户的工作目录中修改文件🡪暂存文件，将修改后的文件快照放入暂存区🡪提交更新，找到暂存区的文件快照，将快照存储到git仓库中去

4.git的安装及配置

下载地址：<https://git-scm.com/download/win/>安装方法百度

安装成功之后进行配置

Windows：运行🡪cmd🡪git config –list显示出配置列表

配置用户信息（在以后的每次提交中git都会使用配置的这个用户信息）：

$git config --global user.name “username”

$git config --global user.email [email@qq.com](mailto:email@qq.com)

$git config <key> ：查看某项配置信息

5.1git基础-获取git仓库

关键英文:仓库(repository)、跟踪(track)、暂存(stage)、提交(commit)、推送(push)、拉取(pull)

获取git项目仓库的两种方法：

在现有的项目或者目录下导入所有的文件到git中

如果你打算使用git来对现有的项目进行管理，你只需要进入该目录并输入

$git init

此命令将创建一个名为.git的子目录，这个子目录含有你初始化的git仓库中所有的必须的文件这些文件是git仓库的骨干。但是项目中的文件还没有被跟踪。

$git add \*.txt

$git add LICENSE

$git commit -m ‘initial project version’

通过git add 命令来实现对指定文件的跟踪(将文件放入暂存区) 和使用commit 来提交

从一个服务器克隆一个现有的git仓库

克隆的仓库的命令git clone [url]，假如要克隆git的可连接库libgit2，可以使用

$ git clone <https://github.com/libgit2/libgit2> [自定义文件夹名称]

这会在当前目录下创建一个名为libgit2的目录，并在这个目录下初始化一个.git文件夹，从远程的仓库拉取下来所有的数据放入.git文件夹中，然后从中读取最新版本的文件进行拷贝

5.2git基础-记录每次更新到仓库

检查当前文件状态：git status

git status -s

$ git status -s

M README

MM Rakefile

A lib/git.rb

M lib/simplegit.rb

?? LICENSE.txt

A：代表是新增加到暂存区的文件

M:代表是已经修改过的文件，左边的M代表是该文件被修改并放入了暂存区

忽略文件可以创建一个.gitignore的文件，列出要忽略的文件模式

cat .gitignore

\*.[oa]

\*~

第二行告诉git忽略所有以 .a或.o结尾的文件。

第三行告诉git 忽略以~结尾的文件

git diff 来分析文件差异

提交更新

当暂存区域已经准备妥当可以提交了。在此之前，一定要确认还有什么修改过的或新建的文件还没有git add过，否则不会记录这些还没有暂存的变化。所以每次准备提交前，先用git status看一下，是不是都已经暂存起来了，然后再运行提交命令：git commit  
git commit -m”message”:提交暂存区的文件并且记录message

git commit -a -m“message”跳过暂存区直接将文件保存到git存储库中去

移除文件

Git rm –cached fileName ：git不再跟踪这个文件

git rm -f fileName :强制删除已经add后的文件，并且不再跟踪

查看提交历史

git log 查看所有的提交历史，最近的提交在最上面显示

git log -p -2 :查看最近两次提交的差异

撤销操作

假如在提交完了才发现漏掉了几个文件没有添加，或者提交信息写错了

git add newFile

git commit --amend -m“message”

执行完这个代码之后上次的提交会被覆盖掉，提交的信息也会被修改为第二次的信息

取消暂存的文件

假如使用了git add \* 增加了所有的文件，其中有一些是不想暂存的则可以使用下面的命令来取消这些文件的暂存

git reset HEAD <fileName>

5.3git基础-远程仓库的使用

查看已经配置的远程仓库服务器命令：git remote -v (需要在指定的地址)

查看远程仓库的详细信息 使用命令:git remote [remote-name] [branch-name]

添加一个新的远程仓库：git remote add <shortname> <url> （shortname可以代替url）

如果想要拉取某个仓库中有但自己没有的信息可以使用代码：

git fecth shortname

如果使用git clone 命令克隆了一个git远程仓库，会自动将其添加到远程仓库并默认以origin为简写，所以git fetch origin 会抓取上一次后新推送的所有的工作，但是不会自动合并到自己工作区中，需要手动合并。

推送到远程仓库

git push [remote-name][branch-name]

只有当你有服务器的写入权限，并且之前没有人推送过时，这条命令才能生效。当你和其他人在同一时间克隆，他们先推送到上游然后你再推送到上游，你的推送就会毫无疑问的被拒绝。你必须先将他们推送的新的信息拉取下来合并到你的工作后才能推送

远程仓库的移除与重命名

远程仓库的重命名：git remote rename oldName newName;

注：这样也会修改你远程分支的名字，过去引用的oldName/master 现在的会引用paul/master

远程仓库的移除：git remote rm [remote-name]

5.4git基础-打标签

和其他的版本控制系统（vcs）一样，git可以给历史中的某一个提交打上标签，以示重要。比较具有代表性的是人们会使用这个功能来标记发布节点

列出标签：git tag

是按照字母的顺序列出标签；但是他们出现的顺序并不重要

创建标签

git的标签主要使用两种类型：轻量标签(lightweight)和附注标签(annotated)

创建一个标签，并且给标签添加注释信息

git tag -a v1.4 -m’message’

6.Git分支-分支简介

创建分支：git branch testing

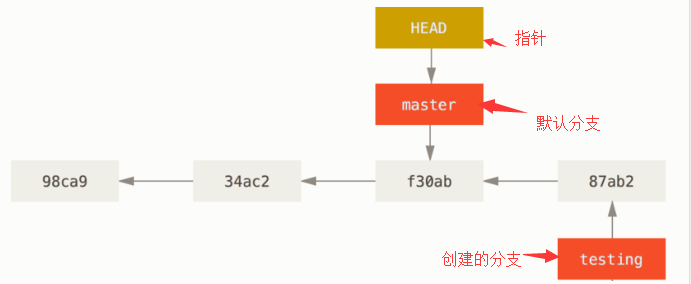
这会在当前所在的提交对象上创建一个指针

git branch 命令仅仅创建一个新的分支，并不会自动切换到新的分支上去

查看各个分支当前所指的对象。命令：git log -decorate

分支切换：git checkout testing

这样HEAD就指向testing分支了



注：当切换到testing的分支下进行开发并且提交后记录会向前进行推进，当用户再次切换到master的分支下时，内容仍是f0ab的内容，综上，从不同的分支可以进行不同的开发流程

删除分支：git branch -d testing 将testing分支删除

合并分支：git merge -testing 将testing分支合并到主分支下

6.1git分支-分支的新建与合并

应用场景：

1. 开发某个程序
2. 正常开发实现新的需求创建一个分支iss53
3. 在iss53上展开工作
4. 此时有一个紧急的bug需要修补，操作步骤如下
5. 切换到你的线上分支
6. 为这个紧急的任务新建一个分支，并在其中修复它
7. 在测试完成之后，切换到线上分支，然后合并这个修补的分支，最后将改动推送到线上分支，推送完成之后删除修复分支
8. 切换回iss53的开发分支，继续开发

git综合

笔记一：git commit 、git push 、git pull、git fetch、git merge的含义与区别

1、git commit :是将本地修改过的文件提交到本地库中

2、git push:是将本地仓库的最新信息发送给远程仓库；

3、git pull:是从远程仓库获取最新版本到本地，并自动merge

4、git fetch:是从远程仓库获取到最新版本到本地，不会自动merge；

5、git merge:是用于从指定的commit合并到当前分支，用来合并两个分支 $ git merge -b//将b分支合并到当前分支，有时候会产生合并冲突（同时修改了一个文件的相同部分），需要手动修改，修改完成之后我们可以用git add 来告诉git冲突已经解决，并且重新提交

git pull =git fatch+git merge

6、git commit -am’message’：跳过add的步骤直接提交,

## 11.IntellJ IDEA 的使用

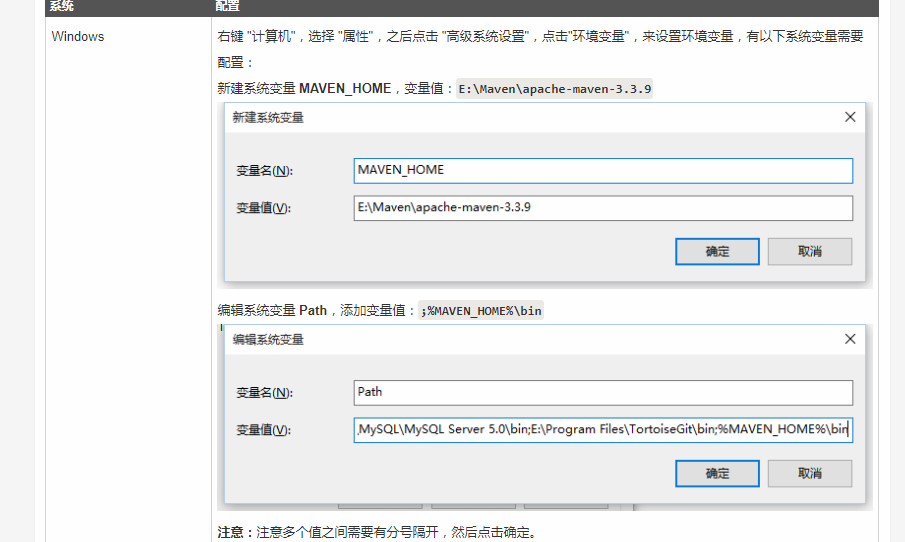
## 12.Maven学习日志

###### 1、maven的安装及环境配置

下载：百度自行下载，下载完成之后解压到对应的目录（全英文路径）

设置Maven的环境变量

如图所示：



###### 2、Maven POM(Project Object Model,项目对象模型)

1、pom 是maven工程工作的基本单元，是一个XML文件，包含了项目的基本信息，用于描述项目如何构建，声明项目依赖。

当执行任务或者目标时，maven会在当前目录中查找POM。他读取POM，获取所需要的配置信息，然后执行目标

2、POM中指定的配置：

项目依赖、插件、执行目标、项目构建profile、项目版本、、项目开发者信息、相关邮件列表信息。



3、所有的pom文件都需要有project元素和三个必需的字段：groupId，artifactId，version。

project：工程的根标签

mode|Version:模型版本需要设置为4.0？？？？

groupId：工程组的表示。他在一个组织或者项目中通常是唯一的

artifactId：这是工程的标识，它通常是工程的名称

version：工程的版本号。