目录

[javascript 5](#_Toc391565496)

[Javascript数字相加的时候要做的处理:(否者就是字符串相加) 5](#_Toc391565497)

[javascript设计模式: 5](#_Toc391565498)

[Javascript 日期格式化 6](#_Toc391565499)

[JSON.parse()和JSON.stringify() 7](#_Toc391565500)

[判断Javascript变量是否为空 8](#_Toc391565501)

[javascript数组使用调用方法汇总 9](#_Toc391565502)

[javascript arguments 10](#_Toc391565503)

[**什么是arguments** 10](#_Toc391565504)

[**创建一个灵活的函数** 11](#_Toc391565505)

[**把arguments对象转换成一个真正的数组** 11](#_Toc391565506)

[**通过预置的arguments对象创建函数** 11](#_Toc391565507)

[**创建引用自身的函数** 12](#_Toc391565508)

[javascript中等于（==）与全等（===）的区别说明 14](#_Toc391565509)

[Extjs 15](#_Toc391565510)

[搭建开发环境不出效果在控制台看到的错: TypeError: r.prototype.styleHooks is not an object or null 15](#_Toc391565511)

[CSS 15](#_Toc391565512)

[span 设CSS样式总是不起作用 15](#_Toc391565513)

[windows 16](#_Toc391565514)

[sc create命令行添加/创建/修改服务 16](#_Toc391565515)

[CMD 查询所有服务 ||文件||使用工具合集 18](#_Toc391565516)

[CMD 很不错的例子 21](#_Toc391565517)

[批处理常用命令及用法大全 22](#_Toc391565518)

[windows7启动远程桌面失败的解决办法： 32](#_Toc391565519)

[服务器:tomcat 35](#_Toc391565520)

[java.lang.OutOfMemoryError：PermGen space 或者Java heap space的产生原因以及解决办法 35](#_Toc391565521)

[**Tomcat version 6.0 only supports J2EE 1.2, 1.3, 1.4, and Java EE 5 Web modules** 37](#_Toc391565522)

[关于The APR based Apache Tomcat Native library警告 37](#_Toc391565523)

[eclipse下启动tomcat出现Setting property 'source' to 'org.eclipse.jst.jee.server: '错误的解决办法 38](#_Toc391565524)

[Tomcat部署项目发布的时候报错：Could not publish server configuration for Tomcat v6.0 Server at localhost. Multiple Contexts have a path of "/jeecms". 39](#_Toc391565525)

[Java 39](#_Toc391565526)

[微软面试题——反转字符串 39](#_Toc391565527)

[J ava字符串反转的三种实现方法 43](#_Toc391565528)

[Java中的instanceof关键字 44](#_Toc391565529)

[Java的native关键字 45](#_Toc391565530)

[乱码处理: 48](#_Toc391565531)

[String intern()方法 55](#_Toc391565532)

[JVM发生OOM时，自动生成DUMP文件。 55](#_Toc391565533)

[java 10大常见异常 56](#_Toc391565534)

[java打开文件夹 58](#_Toc391565535)

[setlocal和endlocal命令执行结果是让中间的程序对于系统变量的改变只在程序内起作用，不会影响整个系统级别 58](#_Toc391565536)

[运算符优先级： 58](#_Toc391565537)

[动态绑定 vs 静态绑定 58](#_Toc391565538)

[技术必知点 62](#_Toc391565539)

[MIME笔记 62](#_Toc391565540)

[数据库 67](#_Toc391565541)

[Oracle 67](#_Toc391565542)

[查看oracel是32位还是64位的 67](#_Toc391565543)

[方法一：使用sqlplus 67](#_Toc391565544)

[方法二： 查看v$version 视图 68](#_Toc391565545)

[方法三：查看v$sql 视图 68](#_Toc391565546)

[关于navicat连接oracle 报 ORA-12737 set CHS16GBK错误的解决方案 69](#_Toc391565547)

[oracle建立用户与授权 70](#_Toc391565548)

[Oracle创建表空间 创建用户 授权 76](#_Toc391565549)

[sqlplus 远程连接oracle 命令 76](#_Toc391565550)

[ora-123737:instant client light:unsupported server character set zhs16gbk 关于navicat连接oracle 76](#_Toc391565551)

[解锁scott用户（oracle11g） 78](#_Toc391565552)

[Mysql 79](#_Toc391565553)

[Mysql之SQL语句优化: 79](#_Toc391565554)

[查看数据库的编码: 79](#_Toc391565555)

[查看数据库详细信息找到占用时间长的Show full processlist 80](#_Toc391565556)

[绿色版mysql注册卸载服务方法 81](#_Toc391565557)

[mysql5.6.16绿色版配置、运行：包括修改root用户的密码 82](#_Toc391565558)

[MySQL创建用户与授权方法 85](#_Toc391565559)

[SQLSERVE问题总结： 87](#_Toc391565560)

[总结JDBC连接SQLServer的错误Error establishing socket 87](#_Toc391565561)

[本地计算机上的SQL Server(MSSQLSERVER)服务启动后停止。某些服务在未由其他服务或程序使用时将自动停止 88](#_Toc391565562)

[SQL Server无法启动,错误代码 3417，提示Windows不能在本地计算机启动，Sql server错误框提示为：“Windows不能在 本地计算机 启动 SQL Server(MSSQLSERVER)。有关更多信息，查阅系统事件日志。如果这是非Microsoft 服务，请与服务厂商联系，并参考特定服务错误代码3417 89](#_Toc391565563)

[SQL 91](#_Toc391565564)

[SQL查询null和””的用法: 91](#_Toc391565565)

[Eclipse:使用总结： 92](#_Toc391565566)

[修改窗口的背景色 92](#_Toc391565567)

[Svn： 93](#_Toc391565568)

[解决subclipse同步冲突问题A conflict in the working copy obstructs the current operation 93](#_Toc391565569)

[办公技巧 94](#_Toc391565570)

[word 94](#_Toc391565571)

[Word自动生成页码: 94](#_Toc391565572)

[Excel 95](#_Toc391565573)

[Powerpoint 95](#_Toc391565574)

[SSH 95](#_Toc391565575)

[Struts 95](#_Toc391565576)

[spring 95](#_Toc391565577)

[hibernate 96](#_Toc391565578)

[android 96](#_Toc391565579)

[在建立android application project项目时出现 The tools need to be updated via the SDK Manager 96](#_Toc391565580)

[对Android Target SDK的理解 97](#_Toc391565581)

[解决eclipse下android项目不能自动生成R.java文件 97](#_Toc391565582)

[tools:context=".MainActivity的作用 98](#_Toc391565583)

# javascript

## Javascript数字相加的时候要做的处理:(否者就是字符串相加)

|  |
| --- |
| var   a   =   3;      var   b   =   98           c   =   a   +   b      想得到c=101      确变成了字符串拼接，得到了398           我该则么做呢？  c   =   parseInt(a)   +   parseInt(b) |

## javascript设计模式:

|  |
| --- |
| Javascript是一种弱类型的语言,其基本类型就三种:字符串,数值型.布尔型,:相互之间的转换:toString()方法可以把数值型,Boolean型转换为字符串:parseFloat(),parseInt可以把字符串转化为数值型双重非可以把字符串或者数值型变成boolean:  Var Boolean=!!num;  函数是最高等的对象:  Javascript的闭包:  三、闭包的概念  闭包就是能够读取其他函数内部变量的函数。  由于在Javascript语言中，只有函数内部的子函数才能读取局部变量，因此可以把闭包简单理解成"定义在一个函数内部的函数"。  所以，在本质上，闭包就是将函数内部和函数外部连接起来的一座桥梁。 |

## Javascript 日期格式化

|  |
| --- |
| /\*\*  \* 时间格式化 返回格式化的时间  \* @param date {object} 可选参数，要格式化的data对象，没有则为当前时间  \* @param fomat {string} 格式化字符串，例如：'YYYY年MM月DD日 hh时mm分ss秒 星期' 'YYYY/MM/DD week' (中文为星期，英文为week)  \* @return {string} 返回格式化的字符串  \*  \* 例子:  \* formatDate(new Date("january 01,2012"));  \* formatDate(new Date());  \* formatDate('YYYY年MM月DD日 hh时mm分ss秒 星期 YYYY-MM-DD week');  \* formatDate(new Date("january 01,2012"),'YYYY年MM月DD日 hh时mm分ss秒 星期 YYYY/MM/DD week');  \*  \* 格式：  \* YYYY：4位年,如1993  　　 \*　　YY：2位年,如93  　　 \*　　MM：月份  　　 \*　　DD：日期  　　 \*　　hh：小时  　　 \*　　mm：分钟  　　 \*　　ss：秒钟  　　 \*　　星期：星期，返回如 星期二  　　 \*　　周：返回如 周二  　　 \*　　week：英文星期全称，返回如 Saturday  　　 \*　　www：三位英文星期，返回如 Sat  \*/  function formatDate(date, format) {  if (arguments.length < 2 && !date.getTime) {  format = date;  date = new Date();  }  typeof format != 'string' && (format = 'YYYY年MM月DD日 hh时mm分ss秒');  var week = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', '日', '一', '二', '三', '四', '五', '六'];  return format.replace(/YYYY|YY|MM|DD|hh|mm|ss|星期|周|www|week/g, function(a) {  switch (a) {  case "YYYY": return date.getFullYear();  case "YY": return (date.getFullYear()+"").slice(2);  case "MM": return date.getMonth() + 1;  case "DD": return date.getDate();  case "hh": return date.getHours();  case "mm": return date.getMinutes();  case "ss": return date.getSeconds();  case "星期": return "星期" + week[date.getDay() + 7];  case "周": return "周" + week[date.getDay() + 7];  case "week": return week[date.getDay()];  case "www": return week[date.getDay()].slice(0,3);  }  });  } |
| 用法：第一个参数传一个Date对象（没有则使用当前时间），第二个参数是格式化字符串，格式如下：  　　　　YYYY：4位年,如1993  　　　　YY：2位年,如93  　　　　MM：月份  　　　　DD：日期  　　　　hh：小时  　　　　mm：分钟  　　　　ss：秒钟  　　　　星期：星期，返回如 星期二  　　　　周：返回如 周二  　　　　week：英文星期全称，返回如 Saturday  　　　　www：三位英文星期，返回如 Sat  　　示例：  \* formatDate(new Date("january 01,2012"));  \* formatDate(new Date());  \* formatDate('YYYY年MM月DD日 hh时mm分ss秒 星期 YYYY-MM-DD YY年 week');  \* formatDate(new Date("january 01,2012"),'YYYY年MM月DD日 hh时mm分ss秒 星期/周 www YYYY/MM/DD week'); |

## JSON.parse()和JSON.stringify()

|  |
| --- |
| parse用于从一个字符串中解析出json对象,如  var str = '{"name":"huangxiaojian","age":"23"}'  结果：  JSON.parse(str)  Object   1. age: "23" 2. name: "huangxiaojian" 3. \_\_proto\_\_: Object   注意：单引号写在{}外，每个属性名都必须用双引号，否则会抛出异常。    stringify()用于从一个对象解析出字符串，如  var a = {a:1,b:2}  结果：  JSON.stringify(a)  "{"a":1,"b":2}" |

## 判断Javascript变量是否为空

|  |
| --- |
| 在不确定变量类型的情况下，判断Javascript变量是否为空不要使用：  if(tmp){}  上述判断的问题在于，例如：当tmp的值为数字0时，也会返回false。  所以尽量使用：  if(tmp != null){}    下面是if(tmp){}判断的结果：  if(tmp) if(tmp != null)  tmp未定义时 //false //false  var tmp; //false //false  var tmp = 0; //false //true  var tmp = 10; //true //true  var tmp = ""; //false //true  var tmp = "ddd"; //true //true  var tmp = 0.0; //false //true  var tmp = new String(); //true //true  结论：要判断tmp未定义或者为null可以使用if(tmp != null)，不能使用if(tmp) |

## javascript数组使用调用方法汇总

|  |
| --- |
| JS数组方法汇总  shift：删除原数组第一项，并返回删除元素的值；如果数组为空则返回undefined  var a = [1,2,3,4,5];  var b = a.shift(); //a：[2,3,4,5] b：1  unshift：将参数添加到原数组开头，并返回数组的长度  var a = [1,2,3,4,5];  var b = a.unshift(-2,-1); //a：[-2,-1,1,2,3,4,5] b：7  注：在IE6.0下测试返回值总为undefined，FF2.0下测试返回值为7，所以这个方法的返回值不可靠，需要用返回值时可用splice代替本方法来使用。  pop：删除原数组最后一项，并返回删除元素的值；如果数组为空则返回undefined  var a = [1,2,3,4,5];  var b = a.pop(); //a：[1,2,3,4] b：5  push：将参数添加到原数组末尾，并返回数组的长度  var a = [1,2,3,4,5];  var b = a.push(6,7); //a：[1,2,3,4,5,6,7] b：7  concat：返回一个新数组，是将参数添加到原数组中构成的  var a = [1,2,3,4,5];  var b = a.concat(6,7); //a：[1,2,3,4,5] b：[1,2,3,4,5,6,7]  splice(start,deleteCount,val1,val2,...)：从start位置开始删除deleteCount项，并从该位置起插入val1,val2,...  var a = [1,2,3,4,5];  var b = a.splice(2,2,7,8,9); //a：[1,2,7,8,9,5] b：[3,4]  var b = a.splice(0,1); //同shift  a.splice(0,0,-2,-1); var b = a.length; //同unshift  var b = a.splice(a.length-1,1); //同pop  a.splice(a.length,0,6,7); var b = a.length; //同push  reverse：将数组反序  var a = [1,2,3,4,5];  var b = a.reverse(); //a：[5,4,3,2,1] b：[5,4,3,2,1]  sort(orderfunction)：按指定的参数对数组进行排序  var a = [1,2,3,4,5];  var b = a.sort(); //a：[1,2,3,4,5] b：[1,2,3,4,5]  slice(start,end)：返回从原数组中指定开始下标到结束下标之间的项组成的新数组  var a = [1,2,3,4,5];  var b = a.slice(2,5); //a：[1,2,3,4,5] b：[3,4,5]  join(separator)：将数组的元素组起一个字符串，以separator为分隔符，省略的话则用默认用逗号为分隔符  var a = [1,2,3,4,5];  var b = a.join("|"); //a：[1,2,3,4,5] b："1|2|3|4|5" |

## javascript arguments

|  |
| --- |
| **什么是arguments** **arguments** **是是JavaScript里的一个内置对象，它很古怪，也经常被人所忽视，但实际上是很重要的。所有主要的js函数库都利用了arguments对象。所以agruments对象对于javascript程序员来说是必需熟悉的。**  所有的函数都有属于自己的一个arguments对象，它包括了函所要调用的参数。他不是一个数组，如果用typeof arguments，返回的是’object’。虽然我们可以用调用数据的方法来调用arguments。比如length,还有index方法。但是数 组的push和pop对象是不适用的。 **创建一个灵活的函数** 看起来貌似argument对象使用起来十分有限，但是实际上它是一个非常有用的对象。你可以通过使用argument对象让函数能够调用数量不定 的参数。在Dean Edwards的base2库里有个格式化的函数，展示了这个灵活性。  function format(string) {    var args = arguments;    var pattern = new RegExp(“%([1-" + arguments.length + "])”, ”g”);    return String(string).replace(pattern, function(match, index) {      return args[index];    });  };  我们提供了一个模板字符串，你可以用”%1”到“%9”给返回值添加一个占位符。然后提供给九个其他参数插入。  format(“And the %1 want to know whose %2 you %3″, ”papers”, ”shirt”, ”wear”);  上面的代码将返回：And the papers want to know whose shirt you wear" .  有件事情我们需要注意下，在定义函数的时候，我们只规定了一个参数，string。Javascript允许我们传递任何数量的参数到一个函数里，不管怎么定义这个函数。Arguments对象对这些都是允许的。 **把arguments对象转换成一个真正的数组** 虽然arguments对象不是一个真正的javascript数组，但是我们还是可以轻易的把它转换成标准的数据 ，然后进行数组操作。  var args = Array.prototype.slice.call(arguments);  那么现在这个变量args就含有一个含有函数所有参数的标准javascript数组对象。 **通过预置的arguments对象创建函数** Arguments对象允许我们去执行所有类型的javascript方法。这边附上一个makeFunc函数的定义。这个函数允许我们去提供一个函数引用和这个函数的所有参数。他将返回一个匿名函数去调用你规定的函数，也提供了匿名函数调用时所附带的参数。  function makeFunc() {    var args = Array.prototype.slice.call(arguments);    var func = args.shift();    return function() {      return func.apply(null, args.concat(Array.prototype.slice.call(arguments)));    };  }  第一个argument对象给makeFunc提供了你想调用的函数的引用。他是从arguments数组里移除的。然后makeFunc返回了一个匿名函数去运行规定的方法。  第一个应用的argument指向了函数调用的范围，主要是函数内部关键部分所指向的。我们先保持这个为null。第二个arguments是一个 数组，会为这个函数转变为arguments对象。makeFunc把原始的数组值串联到arguments对象里提供给匿名函数和所调用函数的数组。  你需要输出一个模板总是相同的位置，这样就可以不用总是在每次引用模板的时候调用format函数。你可以使用makeFunc的通用功能去返回可以调用format然后自动补充模板的函数。  var majorTom = makeFunc(format, ”This is Major Tom to ground control. I’m %1.”);  你可以像这样调用majorTom函数：  majorTom(“stepping through the door”);  majorTom(“floating in a most peculiar way”);  每一次你调用majorTom，它会同时调用format函数和第一个argument，已经写好的模板。那么将返回  “This is Major Tom to ground control. I’m stepping through the door.”  “This is Major Tom to ground control. I’m floating in a most peculiar way.” **创建引用自身的函数** 你可能觉得这样很酷，不过，arguments还有更多的惊喜。他还有其他有用的特性：callee方法 。Arguments.callee包括了一个函数的引用去创建一个argument对象。那么该如何使用呢？  Arguments.callee方法能让一个匿名函数很方便的指向本身。  Repeat是一个承载了一个函数引用和两个数字的函数。第一个数字是函数调用几次，第二个数字是每个调用的间隔时间，单位是毫秒。  function repeat(fn, times, delay) {    return function() {      if(times– > 0) {        fn.apply(null, arguments);        var args = Array.prototype.slice.call(arguments);        var self = arguments.callee;        setTimeout(function(){self.apply(null,args)}, delay);      }    };  }  Repeat函数使用了arguments.callee方法从变量self去获取一个引用，指向运行原始指令的函数。这样，匿名函数就可以再次调用本身。  我有一段超级简介的函数，承载了一个字符串和执行alert方法。  function comms(s) {    alert(s);  }  然而，我想创建一个特殊的版本，通过这个版本我可以重复这个动作三次，每次间隔2秒钟。那么，我们可以  var somethingWrong = repeat(comms, 3, 2000);  somethingWrong(“Can you hear me, major tom?”);  调用了somethingWrong函数的结果就是复这个动作三次，每次alert间隔2秒钟。  Arguments虽然不是经常用到，有点古怪，但是，它充满了惊喜，非常值得我们去了解 。 |

## javascript中等于（==）与全等（===）的区别说明

|  |
| --- |
| javascript中等于（==）与全等（===）的区别说明  等于（==）的情况下 只要值相同就返回True。而全等（===）的时候需要值和类型都要匹配才能返回True |

## Javascript中this的使用：

|  |
| --- |
| **this 是 Javascript 语言的一个关键字。**  它代表函数运行时，自动生成的一个内部对象，只能在函数内部使 用。比如，  function test(){  　　this.x = 1;  }  随着函数使用场合的不同， this 的值会发生变化。但是有一个总的原则，那就是 this 指的是，调用函数的那个对象 |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# Extjs

## 搭建开发环境不出效果在控制台看到的错: TypeError: r.prototype.styleHooks is not an object or null

|  |
| --- |
| 分析可能的原因:没有正确导入需要的js文件:(extjs提供了demo可以从里面查看怎样正确引用  解决:  <link href="extjs/resources/css/ext-all.css" rel="stylesheet" />  <script type="text/javascript" src="extjs/ext-all.js"></script>  <script type="text/javascript" src="extjs/locale/ext-lang-zh\_CN.js"></script>   引这3个就行了 其余俩的重复了 冲突了  所以才会照成这样的错误 |

## Extjs中grid的布局：

|  |
| --- |
| 可以上面一个下面一个  在panel中的items里直接写[grid1,grid2]，不用布局  如果向左边一个右边一个  在grid中首先分别添加region：“west”和region：“center”  再在panel中写上  layout:"border"  items:[grid1,grid2] |

|  |
| --- |
|  |

# CSS

## span 设CSS样式总是不起作用

|  |
| --- |
| <span id='span\_slide\_container' style='display:block; overflow-y: auto; overflow-x: auto; height: 500px;'>  此处 display:block; 一定要加上，否则，span不起作用  如果要定义span居中，必须先让span成块级元素显示，也就是说，要先定义span的display:block;属性，然后再给span添加边距属性margin:0px auto;这样就能达到你想要的效果了。span的属性很多，几乎所有标签的属性都可以定义在span上，只是很多属性都必须在span成块级元素时菜其作用，span本身只是为了补充a标签的作用，是一个辅助标签，一般只能识别对文字的样式的你故意，所以如果想给span标签定义更多的其他样式属性，你就必须先给span加上块级属性，也就说添加display:block;属性，这样以后你的样式就可以起作用了 |

|  |
| --- |
|  |

# windows

## sc create命令行添加/创建/修改服务

|  |
| --- |
| sc create命令行添加/创建/修改服务  添加服务： sc create BITS binpath= "C:\WINDOWS\system32\svchost.exe -k netsvcs" type= share start= auto displayname= "Background Intelligent Transfer Service" depend= RpcSs/EventSystem 修改描述： sc description BITS "利用空闲的网络带宽在后台传输文件。如果服务被停用，例如 Windows Update 和 MSN Explorer 的功能将无法自动下载程序和其他信息。如果此服务被禁用，任何依赖它的服务如果没有容错技术以直接通过 IE 传输文件，一旦 BITS 被禁用，就可能无法传输文件。"  添加服务： sc create PolicyAgent binpath= "C:\WINDOWS\system32\lsass.exe" type= share start= auto displayname= "IPSEC Services" depend= RPCSS/Tcpip/IPSec 修改描述： sc description PolicyAgent "提供 TCP/IP 网络上客户端和服务器之间端对端的安全。如果此服务被停用，网络上客户端和服务器之间的 TCP/IP 安全将不稳定。如果此服务被禁用，任何依赖它的服务将无法启动。"  SC命令的更多用法请在查看help sc  (注册表修改前请做好备份)，  SvcHost共享服务的特别之处：或许重启后还需要到注册表位置：[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost]，查看"C:\WINDOWS\system32\svchost.exe -k netsvcs"以红色字为名的键(这里以netsvcs为例)，键值里是否定义有sc create BITS蓝色字的服务(这里以BITS为例)，如果没有就在键值把服务名(这里以BITS为例)添加上去,再重启一次。  @echo off copy internat.exe %windir%\system32 sc create Internat binpath= %windir%\system32\internat.exe type= own start= auto DisplayName= "Internat Aionegirl" sc description Internat "输入法" sc start Internat pause  @echo off del %windir%\system32\internat.exe /s /q sc stop Internat sc delete Internat pause sc create MyService binpath= C:\\winnt\\sys tem32\\NewServ.exe type= own  start= auto sc create MyService binpath= F:\\clear.bat type= own start= auto  type=----own, share, interact, kernel, filesys  关于建立服务的类型，选项值包括驱动程序使用的类型，默认是share。  start=----boot, sys tem, auto, demand, disabled  关于启动服务的类型，选项值包括驱动程序使用的类型，默认是demand（手动）。  添加服务：  sc create BITS binpath= "C:/WINDOWS/system32/svchost.exe -k netsvcs" type= share start= auto displayname= "Background Intelligent Transfer Service" depend= RpcSs/[EVE](http://www.txwb.com/Article/wyzl/GCYX/200703/5561.html)ntSystem  修改描述：  sc description BITS "利用空闲的网络带宽在后台传输文件。如果服务被停用，例如 Windows Update 和 [MSN](http://www.jztop.com/e/search/result/?searchid=498) Explorer 的功能将无法自动下载程序和其他信息。如果此服务被禁用，任何依赖它的服务如果没有容错技术以直接通过 IE 传输文件，一旦 BITS 被禁用，就可能无法传输文件。"   添加服务：  sc create PolicyAgent binpath= "C:/WINDOWS/system32/lsass.exe" type= share start= auto displayname= "IPSEC Services" depend= RPCSS/Tcpip/IPSec  修改描述：  sc description PolicyAgent "提供 TCP/IP 网络上客户端和[服务器](http://www.jztop.com/systems/)之间端对端的安全。如果此服务被停用，网络上客户端和[服务器](http://www.jztop.com/systems/)之间的 TCP/IP 安全将不稳定。如果此服务被禁用，任何依赖它的服务将无法启动。"   SC命令的更多用法请在查看help sc   ([注册表](http://www.jztop.com/os/regedit/)修改前请做好备份)，   SvcHost共享服务的特别之处：或许重启后还需要到[注册表](http://www.jztop.com/os/regedit/)位置：[HKEY\_LOCAL\_MACHINE/SOFTWARE/Microsoft/Windows NT/CurrentVersion/SvcHost]，查看"C:/WINDOWS/system32/svchost.exe -k netsvcs"以红色字为名的键(这里以netsvcs为例)，键值里是否定义有sc create BITS蓝色字的服务(这里以BITS为例)，如果没有就在键值把服务名(这里以BITS为例)添加上去,再重启一次。  Creates a service entry in the registry and Service Database.  SYNTAX:  sc create [service name] [binPath= ] <option1> <option2>...  CREATE OPTIONS: NOTE: The option name includes the equal sign. type= <own|share|interact|kernel|filesys|rec>        (default = own) start= <boot|system|auto|demand|disabled>        (default = demand) error= <normal|severe|critical|ignore>        (default = normal) binPath= <BinaryPathName> group= <LoadOrderGroup> tag= <yes|no> depend= <Dependencies(separated by / (forward slash))> obj= <AccountName|ObjectName>        (default = LocalSystem) DisplayName= <display name> password= <password>  ***提示大家：sc cretate命令后面带的参数比如[binPath= ]的等号后面必须有个空格，没空格你怎么折腾都不行的，哈哈。微软还真搞笑啊，搞这么个功能迷惑人。同样，后面的其它参数比如group= , displayname=的等号后面都需要空格，不然sc create命令不会成功创建服务，只会返回给你帮助信息。***  ***sc create nlsvc binpath= "F:\Program Files\NetLimiter 2 Lite\NLClient.exe" type= own start= demand DisplayName= NetLimiter*** |

## CMD 查询所有服务 ||文件||使用工具合集

|  |
| --- |
| Cmd查询所有的服务  sc query type= all state= all |findstr /i service\_name.\* |more  cmd查找文件  dir d:\test\\*.jpg /S  使用工具合集  appwiz.cpl                添加删除程序  compmgmt.msc       计算机管理  control                       控制面板  control userpasswords2--------用户帐户设置  cleanmgr                  垃圾整理  calc                             启动计算器  chkdsk                        磁盘检查  devmgmt.msc           设备管理器  diskmgmt.msc           磁盘管理实用程序  dfrg.msc-                   磁盘碎片整理程序  drwtsn32                  系统医生  dvdplay                       启动Media Player  dxdiag                          DirectX Diagnostic Tool  gpedit.msc                组策略编辑器  gpupdate /target:computer /force 强制刷新组策略  eventvwr.exe             事件查看器  explorer                      打开资源管理器  logoff                         注销命令  lusrmgr.msc            本机用户和组  systeminfo                  系统信息  net start (servicename)----启动该服务  net stop (servicename)-----停止该服务  notepad--------打开记事本  Nslookup-------IP地址侦测器  perfmon.msc----计算机性能监测程序  regedit----------注册表编辑器  regedt32-------注册表编辑器  regsvr32 /u \*.dll----停止dll文件运行  route print------查看路由表   rononce -p ----15秒关机  rsop.msc-------组策略结果集  rundll32.exe rundll32.exe %Systemroot%System32shimgvw.dll,ImageView\_Fullscreen----启动一个空白的Windows 图片和传真查看器  secpol.msc--------本地安全策略  services.msc---本地服务设置  sfc /scannow-----启动系统文件检查器  sndrec32-------录音机  taskmgr-----任务管理器（适用于2000／xp／2003）  tsshutdn-------60秒倒计时关机命令  winchat--------XP自带局域网聊天  winmsd---------系统信息  winver-----显示About Windows 窗口  wupdmgr-----------Windows Update |

## CMD 很不错的例子

|  |
| --- |
| @echo off  title 密码验证  SETLOCAL  set pwd=0  set times=3  set mydir=%cd%  set author=xfire  md %mydir%  echo ####################################################################### >> %mydir%\mylog.txt  echo 操作： 激活密码验证 时间：%time% 日期：%date% >> %mydir%\mylog.txt  echo 状态： 等待验证…… >> %mydir%\mylog.txt  echo. >> %mydir%\mylog.txt  netstat -an >>%mydir%\net.txt  echo [ LOGIN ]  :password  set /p pwd= 请输入您的密码：  set /A times=%times%-1  if %pwd%==999 goto pass  echo \*\*\*\*\* 密码验证错误，请您重新输入 您还有 %times% 次机会输入密码 \*\*\*\*\*  echo .  if %times%==0 goto close  echo 状态： 用户输入密码 验证失败 时间：%time% >> %mydir%\mylog.txt  goto password  :close  echo 状态： 用户3次输入密码错误 程序锁定 时间：%time% >> %mydir%\mylog.txt  title 对不起，您无法使用%author%的CMD  echo --------------------------------------------------------------------------------  echo 由于您3次密码验证失败，程序已经被锁定，您已经无法继续操作，您可以选择关闭  echo 本窗口,您也可以通过留言来与%author%取得联系，输入留言后，请按回车提交！  1.exe  echo --------------------------------------------------------------------------------  echo [ MESSAGES ]  :message  echo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  set /p msg= 请输入您的留言：  echo . >> %mydir%\mymsg.txt  echo ####################################################################### >> %mydir%\mymsg.txt  echo 日期：%date% 时间：%time% >> %mydir%\mymsg.txt  echo 留言内容： >> %mydir%\mymsg.txt  echo %msg% >> %mydir%\mymsg.txt  echo …… OK ……  echo 您的留言已经记录，您可以选择关闭窗口也可以选择继续留言  echo 操作： 用户留言 时间：%time% >> %mydir%\mylog.txt  goto message  :pass  echo 状态： 程序已开放，欢迎使用 时间：%time% >> %mydir%\mylog.txt  title %author%的CMD  ENDLOCAL  echo good batch  netstat -an |

## [批处理常用命令及用法大全](http://blog.csdn.net/dangerous_fire/article/details/8482328)

|  |
| --- |
| 阅读本文需要一定的dos基础概念，象：盘符、文件、目录(文件夹)、子目录、根目录、当前目录 每个命令的完整说明请加 /? 参数参考微软的帮助文档可以看到，在 /? 帮助里，"命令扩展名"一词会经常出现"命令扩  展名"是指相对于win98的dos版本而言，每个命令新增的功能 命令测试环境win2000proSP4win98的命令功能太少，就不作研究了  注：如果对某一命令还不是很熟悉，可以在命令行窗口下输入：命令名/?的方式来获得帮助。 例如：对dir命令的应用不熟悉，可以在命令行窗口下输入：  dir /?  1 echo 和 @回显控制命令 @                    #关闭单行回显 echo off             #从下一行开始关闭回显 @echo off            #从本行开始关闭回显。一般批处理第一行都是这个 echo on              #从下一行开始打开回显 echo                 #显示当前是 echo off 状态还是 echo on 状态 echo.                #输出一个"回车换行"，一般就是指空白行 echo hello world     #输出hello world "关闭回显"是指运行批处理文件时，不显示文件里的每条命令，只显示运行结果批处理开始和结束时，系统都会自动打开  回显 2 errorlevel程序返回码  echo %errorlevel%每个命令运行结束，可以用这个命令行格式查看返回码用于判断刚才的命令是否执行成功默认值为0，  一般命令执行出错会设 errorlevel 为1  3 dir显示目录中的文件和子目录列表 dir                 #显示当前目录中的文件和子目录 dir /a              #显示当前目录中的文件和子目录，包括隐藏文件和系统文件 dir c: /a:d         #显示 C 盘当前目录中的目录 dir c:\ /a:-d       #显示 C 盘根目录中的文件dir d:\mp3 /b/p     #逐屏显示 d:\mp3 目录里的文件，只显示文件  名，不显示时间和大小 dir \*.exe /s显示当前目录和子目录里所有的.exe文件其中 \* 是通配符，代表所有的文件名，还一个通配符 ? 代表一个  任意字母或汉字如 c\*.\* 代表以 c 开头的所有文件?.exe 代表所有文件名是一个字母的.exe文件如果指定的目录或文件  不存在，将返回 errorlevel 为1;每个文件夹的 dir 输出都会有2个子目录 . 和 ... 代表当前目录.. 代表当前目录的  上级目录 dir .               #显示当前目录中的文件和子目录 dir ..              #显示当前目录的上级目录中的文件和子目录 其它参数可参考 dir /?  4 cd更改当前目录 cd mp3              #进入当前目录中的mp3 目录 cd ..               #进入当前目录中的上级目录 cd\                 #进入根目录 cd                  #显示当前目录 cd /d d:\mp3        #可以同时更改盘符和目录 cd "Documents and Settings"\All users文件名带空格，路径前需要加上引号！！如果更改到的目录不存在，则出错返  回 errorlevel=1  5 md创建目录 md abc              #在当前目录里建立子目录 abc md d:\a\b\c         #如果 d:\a 不存在，将会自动创建  6 rd删除目录 rd abc              #删除当前目录里的 abc 子目录，要求为空目录 rd /s/q d:\temp     #删除 d:\temp 文件夹及其子文件夹和文件，不需要按 Y 确认  7 del删除文件 del d:\test.txt     #删除指定文件，不能是隐藏、系统、只读文件 del \*.\*删除当前目录里的所有文件，不包括隐藏、系统、只读文件，要求按 Y 确认 del /q/a/f d:\temp\\*.\*删除 d:\temp 文件夹里面的所有文件，包括隐藏、只读、系统文件，不包括子目录 del /q/a/f/s d:\temp\\*.\*删除 d:\temp 及子文件夹里面的所有文件，包括隐藏、只读、系统文件，不包括子目录  8 ren文件重命名 ren 1.txt 2.bak     #把 1.txt 更名为 2.bak ren \*.txt \*.ini     #把当前目录里所有.txt文件改成.ini文件 ren d:\temp tmp     #支持对文件夹的重命名  9 cls清屏  10 type显示文件内容 type c:\boot.ini    #显示指定文件的内容，程序文件一般会显示乱码 type \*.txt          #显示当前目录里所有.txt文件的内容  11 copy拷贝文件 copy c:\test.txt d:\复制 c:\test.txt 文件到 d:\ copy c:\test.txt d:\test.bak复制 c:\test.txt 文件到 d:\ ，并重命名为 test.bak copy c:\\*.\*复制 c:\ 所有文件到当前目录，不包括隐藏文件和系统文件不指定目标路径，则默认目标路径为当前目录 copy con test.txt从屏幕上等待输入，按 Ctrl+Z 结束输入，输入内容存为test.txt文件con代表屏幕，prn代表打印机  ，nul代表空设备 copy 1.txt + 2.txt 3.txt合并 1.txt 和 2.txt 的内容，保存为 3.txt 文件如果不指定 3.txt ，则保存到 1.txt copy test.txt +复制文件到自己，实际上是修改了文件日期  12 title设置cmd窗口的标题 title 新标题        #可以看到cmd窗口的标题栏变了  13 ver显示系统版本  14 label 和 vol设置卷标 vol                 #显示卷标 label               #显示卷标，同时提示输入新卷标 label c:system      #设置C盘的卷标为 system  15 pause暂停命令 运行该命令时，将显示下面的消息：请按任意键继续 . . . 一般用于看清楚屏幕上显示的内容  16 rem 和 ::注释命令 注释行不执行操作  17 date 和 time日期和时间 date          #显示当前日期，并提示输入新日期，按"回车"略过输入 date/t        #只显示当前日期，不提示输入新日期 time          #显示当前时间，并提示输入新时间，按"回车"略过输入 time/t        #只显示当前时间，不提示输入新时间  18 goto 和 :跳转命令 :label        #行首为:表示该行是标签行，标签行不执行操作 goto label    #跳转到指定的标签那一行  19 find (外部命令)查找命令 find "abc" c:\test.txt在 c:\test.txt 文件里查找含 abc 字符串的行如果找不到，将设 errorlevel 返回码为1 find /i "abc" c:\test.txt查找含 abc 的行，忽略大小写 find /c "abc" c:\test.txt显示含 abc 的行的行数  20 more (外部命令)逐屏显示 more c:\test.txt    #逐屏显示 c:\test.txt 的文件内容  21 tree显示目录结构 tree d:\            #显示D盘的文件目录结构  22 &顺序执行多条命令，而不管命令是否执行成功 c: & cd\ & dir /w相当于把下面3行命令写到1行去了c:cd\dir /w  23 &&顺序执行多条命令，当碰到执行出错的命令后将不执行后面的命令 f: && cd\ && dir >c:\test.txt注意如果f盘不存在，那么后面2条命令将不会执行 find "ok" c:\test.txt && echo 成功如果找到了"ok"字样，就显示"成功"，找不到就不显示  24 ||顺序执行多条命令，当碰到执行正确的命令后将不执行后面的命令  find "ok" c:\test.txt || echo 不成功如果找不到"ok"字样，就显示"不成功"，找到了就不显示  25 |管道命令 前一个命令的执行结果输出到后一个命令 dir \*.\* /s/a | find /c ".exe"管道命令表示先执行 dir 命令，对其输出的结果执行后面的 find 命令该命令行结果：  输出当前文件夹及所有子文件夹里的.exe文件的个数 type c:\test.txt|more这个和 more c:\test.txt 的效果是一样的  26 > 和 >>输出重定向命令 > 清除文件中原有的内容后再写入>> 追加内容到文件末尾，而不会清除原有的内容主要将本来显示在屏幕上的内容输出  到指定文件中指定文件如果不存在，则自动生成该文件 echo hello world>c:\test.txt生成c:\test.txt文件，内容为hello world这个格式在批处理文件里用得很多，可以生成   .reg .bat .vbs 等临时文件 type c:\test.txt >prn屏幕上不显示文件内容，转向输出到打印机 echo hello world>con在屏幕上显示hello world，实际上所有输出都是默认 >con 的 copy c:\test.txt f: >nul拷贝文件，并且不显示"文件复制成功"的提示信息，但如果f盘不存在，还是会显示出错信息 copy c:\test.txt f: >nul 2>nul不显示"文件复制成功"的提示信息，并且f盘不存在的话，也不显示错误提示信息 echo ^^W ^> ^W>c:\test.txt生成的文件内容为 ^W > W^ 和 > 是控制命令，要把它们输出到文件，必须在前面加个 ^   符号 27 <从文件中获得输入信息，而不是从屏幕上 一般用于 date time label 等需要等待输入的命令 @echo offecho 2005-05-01>temp.txtdate <temp.txtdel temp.txt这样就可以不等待输入直接修改当前日期  28. %0 %1 %2 %3 %4 %5 %6 %7 %8 %9 %\*命令行传递给批处理的参数 %0 批处理文件本身%1 第一个参数%9 第九个参数%\* 从第一个参数开始的所有参数在C盘根目录新建test.bat，内容如下：   1. @echo off 2. echo %0 3. echo %1 4. echo %2 5. echo %\*   *复制代码*  运行cmd，输入 c:\test.bat "/a" /b /c /d可以看出每个参数的含意 修改test.bat内容如下   1. @echo off 2. echo %1 3. echo %~1 4. echo %0 5. echo %~f0 6. echo %~d0 7. echo %~p0 8. echo %~n0 9. echo %~x0 10. echo %~s0 11. echo %~a0 12. echo %~t0 13. echo %~z0   *复制代码*  再运行cmd，输入 c:\test.bat "/a" /b /c /d可以参照 call/? 或 for/? 看出每个参数的含意注意这里可以对文件进行日期比较和大小比较   1. echo load "%%1" "%%2">c:\test.txt   *复制代码*  生成的文件内容为 load "%1" "%2" 批处理文件里，用这个格式把命令行参数输出到文件  31 set设置变量 引用变量可在变量名前后加 % ，即 %变量名% set                    #显示目前所有可用的变量，包括系统变量和自定义的变量 echo %SystemDrive%     #显示系统盘盘符。系统变量可以直接引用 set p                  #显示所有以p开头的变量，要是一个也没有就设errorlevel=1 set p=aa1bb1aa2bb2     #设置变量p，并赋值为 = 后面的字符串，即aa1bb1aa2bb2 echo %p%               #显示变量p代表的字符串，即aa1bb1aa2bb2 echo %p:~6%            #显示变量p中第6个字符以后的所有字符，即aa2bb2 echo %p:~6,3%          #显示第6个字符以后的3个字符，即aa2 echo %p:~0,3%          #显示前3个字符，即aa1 echo %p:~-2%           #显示最后面的2个字符，即b2 echo %p:~0,-2%         #显示除了最后2个字符以外的其它字符，即aa1bb1aa2b echo %p:aa=c%          #用c替换变量p中所有的aa，即显示c1bb1c2bb2 echo %p:aa=%           #将变量p中的所有aa字符串置换为空，即显示1bb12bb2 echo %p:\*bb=c%         #第一个bb及其之前的所有字符被替换为c，即显示c1aa2bb2 set p=%p:\*bb=c%        #设置变量p，赋值为 %p:\*bb=c% ，即c1aa2bb2 set /a p=39            #设置p为数值型变量，值为39 set /a p=39/10         #支持运算符，有小数时用去尾法，39/10=3.9，去尾得3，p=3set /a p=p/10          #用 /a   参数时，在 = 后面的变量可以不加%直接引用set /a p="1&0"         #"与"运算，要加引号。其它支持的运算符参见  set/? set p=                 #取消p变量 set /p p=请输入屏幕上显示"请输入"，并会将输入的字符串赋值给变量p注意这条可以用来取代 choice 命令 注意变量在 if 和 for 的复合语句里是一次性全部替换的，如 @echo off set p=aaa if %p%==aaa (       echo %p%      set p=bbb       echo %p%    ) 结果将显示aaaaaa因为在读取 if 语句时已经将所有 %p% 替换为aaa这里的"替换"，在 /? 帮助里就是指"扩充"、"环境  变量扩充"可以启用"延缓环境变量扩充"，用 ! 来引用变量，即 !变量名! @echo off SETLOCAL ENABLEDELAYEDEXPANSION set p=aaaif %p%==aaa (     echo %p%     set p=bbb     echo !p!    ) ENDLOCAL 结果将显示aaabbb 还有几个动态变量，运行 set 看不到 %CD%                  #代表当前目录的字符串 %DATE%                #当前日期 %TIME%                #当前时间 %RANDOM%              #随机整数，介于0~32767 %ERRORLEVEL%          #当前ERRORLEVEL 值 %CMDEXTVERSION%       #当前命令处理器扩展名版本号 %CMDCMDLINE%          #调用命令处理器的原始命令行可以用echo命令查看每个变量值，如 echo %time%注意 %time%   精确到毫秒，在批处理需要延时处理时可以用到  32 start批处理中调用外部程序的命令，否则等外部程序完成后才继续执行剩下的指令 start explorer d:\调用图形界面打开D盘 @echo offcd /d %~dp0regedit /s 劲舞团.regstart patcher.exe 不加 start 命令的话，"劲舞团"运行时，后面会有个黑乎乎的cmd窗口  33 call批处理中调用另外一个批处理的命令，否则剩下的批处理指令将不会被执行有时有的应用程序用start调用出错的  ，也可以call调用  34 choice (外部命令)选择命令让用户输入一个字符，从而选择运行不同的命令，返回码errorlevel为1234……win98里  是choice.comwin2000pro里没有，可以从win98里拷过来win2003里是choice.exechoice /N /C y /T 5 /D y>nul延时5秒  下面是个 choice 语句的例子 @echo off rem 以下在win2000pro运行通过，从win98里拷的chioce.com文件 choice /c:abc aaa,bbb,ccc if errorlevel 3 goto ccc if %errorlevel%==2 goto bbb if errorlevel==1 goto aaa rem 必须先判断数值高的返回码rem 可以看到 errorlevel 值的判断有3种写法，有时某种写法不好用，可以用另外的写法 rem 直接运行  chioce相当于运行 choice /c:yn:aaa echo aaa goto end :bbb echo bbb goto end :ccc echo ccc goto end :end  35 assoc 和 ftype文件关联assoc 设置'文件扩展名'关联，关联到'文件类型'ftype 设置'文件类型'关联，关联到'执行程序和参数'当你双击一个.txt文件时，windows并不是根据.txt直接判断用 notepad.exe 打开而是先判断.txt属于 txtfile '文件类型'再调用 txtfile 关联的命令行 txtfile=%SystemRoot%\system32\NOTEPAD.EXE %1可以在"文件夹选项"→"文件类型"里修改这2种关联 assoc           #显示所有'文件扩展名'关联 assoc .txt      #显示.txt代表的'文件类型'，结果显示 .txt=txtfile assoc .doc      #显示.doc代表的'文件类型'，结果显示 .doc=Word.Document.8 assoc .exe      #显示.exe代表的'文件类型'，结果显示 .exe=exefile ftype           #显示所有'文件类型'关联 ftype exefile   #显示exefile类型关联的命令行，结果显示 exefile="%1" %\* assoc .txt=Word.Document.8设置.txt为word类型的文档，可以看到.txt文件的图标都变了 assoc .txt=txtfile恢复.txt的正确关联 ftype exefile="%1" %\*恢复 exefile 的正确关联  如果该关联已经被破坏，可以运行 command.com ，再输入这条命令  36 pushd 和 popd切换当前目录 @echo off c: & cd\ & md mp3       #在 C:\ 建立 mp3 文件夹 md d:\mp4               #在 D:\ 建立 mp4 文件夹 cd /d d:\mp4            #更改当前目录为 d:\mp4 pushd c:\mp3            #保存当前目录，并切换当前目录为 c:\mp3 popd                    #恢复当前目录为刚才保存的 d:\mp4一般用处不大，在当前目录名不确定时，会有点帮助  32 subst (外部命令)映射磁盘。 subst z: \\server\d     #这样输入z:就可以访问\\server\d了 subst z: /d             #取消该映射 subst                   #显示目前所有的映  38  xcopy (外部命令)文件拷贝 xcopy d:\mp3 e:\mp3 /s/e/i/y复制 d:\mp3 文件夹、所有子文件夹和文件到 e:\ ，覆盖已有文件加 /i 表示如果 e:\ 没有 mp3 文件夹就自动新建一个，否则会有询问  39 一些不常用的内部命令 >& 将一个句柄的输出写入到另一个句柄的输入中 <& 从一个句柄读取输入并将其写入到另一个句柄输出中shift 命令行传递给批处理的参数不止9个时，用以切换参数color 设置cmd窗口的显示颜色 pormpt 更改命令提示符号，默认都是 盘符:\路径\> ，如 c:\>  40 format (外部命令)格式化硬盘 format c: /q/u/autotest/q表示快速格式化，/autotest表示自动格式化，不需要按 Y 确认/u表示每字节用   F6 覆盖硬盘数据，使其不可用软件恢复format c: /c格式化C盘，并检测坏道  41 fdisk (外部命令)硬盘分区win2000不带  该命令win98里的fdisk不支持80G以上大硬盘，winme里的支持fdisk/mbr重建硬盘分区表，一般用于清除引导区病毒、还  原精灵注意使用该命令不能从硬盘启动，必须软驱或光驱启动后直接运行  42 ping (外部命令) ping -l 65500 -t 192.168.1.200不停的向192.168.1.200计算机发送大小为65500byte的数据包 ping -n 10 127.0.0.1>nulping自己10次，可用于批处理延时10秒  43 SC (外部命令)服务控制命令 sc create aaa displayname= bbb start= auto binpath= "C:\WINDOWS\System32\alg.exe"创建服务，服务名称aaa，显示名称bbb，启动类型:自动可执行文件的路径"C:\WINDOWS\System32\alg.exe" sc description aaa "ccc"更改aaa的描述为ccc sc config aaa start= disabled binpath= "C:\WINDOWS\System32\svchost.exe -k netsvcs"更改aaa的启动类型:已禁用更改aaa的可执行文件的路径"C:\WINDOWS\System32\svchost.exe -k netsvcs" sc config aaa start= demand displayname= ddd更改aaa的启动类型:手动更改aaa的显示名称ddd sc start aaa启动aaa服务 sc stop aaa停止aaa服务 sc delete aaa删除aaa服务 |

使用bat打开网页：

|  |
| --- |
| explorer.exe open=http://www.baidu.com  这个命令才是最好的，有些操作系统需要open参数。  或者  在BAT文件中输入以下命令（比如说打开百度网站）  "C:\Program Files\Internet Explorer\iexplore.exe" [www.baidu.com](http://www.baidu.com)  或者  start [www.baidu.com](http://www.baidu.com) |

## windows7启动远程桌面失败的解决办法：

|  |
| --- |
| WIN7系统的远程桌面开启需要3个东西  1、开启远程桌面，根据提示关闭防火墙（WINDOWS自带的防火墙，真心木有用啊）  http://files.jb51.net/file_images/article/201307/20130705181528124.jpg  2、设置账户密码：无密码的账户是无法进行远程登录的，这是不安全的，不符合WINDOWS习惯的。 http://files.jb51.net/file_images/article/201307/20130705181528125.jpg  3、开启windows服务，以下这3个服务必须开启，不开启无法访问的。 http://files.jb51.net/file_images/article/201307/20130705181528126.jpg |

## ISO格式的文件添加默认图标

|  |
| --- |
| 用可以修改ISO映像的软件解包然后图标文件QQ.ico放到里面去,ISO映像的根目录那里,打开一个记事本在里面写上如下内容:  [autorun]  icon=QQ.ico  然后另存为autorun.inf到ISO映像的根目录那里  保存ISO映像用虚拟光驱加载看看  如果你要自动播放的话那就再加一个  open=程序路径  如果要在盘符的右键菜单加一个项的话那就这样  shell\名称\command=要运行的程序路径  那个程序最好在映像里面不要盘符  看一下我的：  [autorun]  icon=favicon.ico  open=  shell\加密文件\command=UDE.exe |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

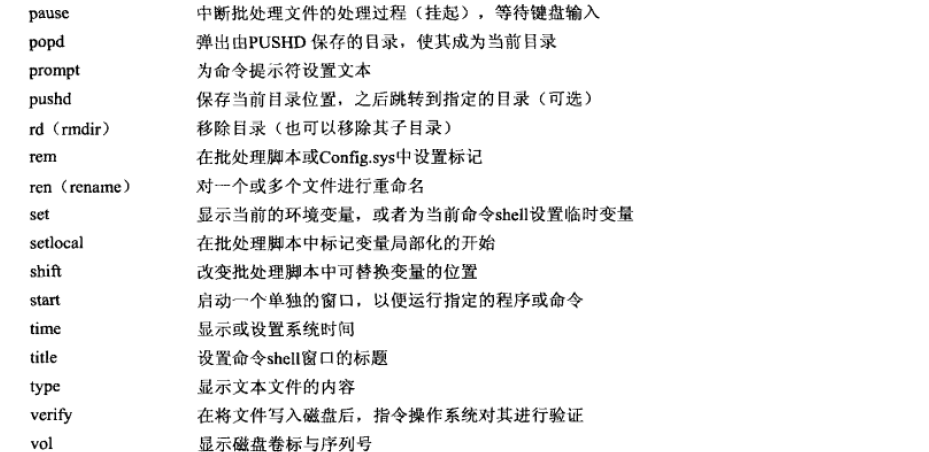
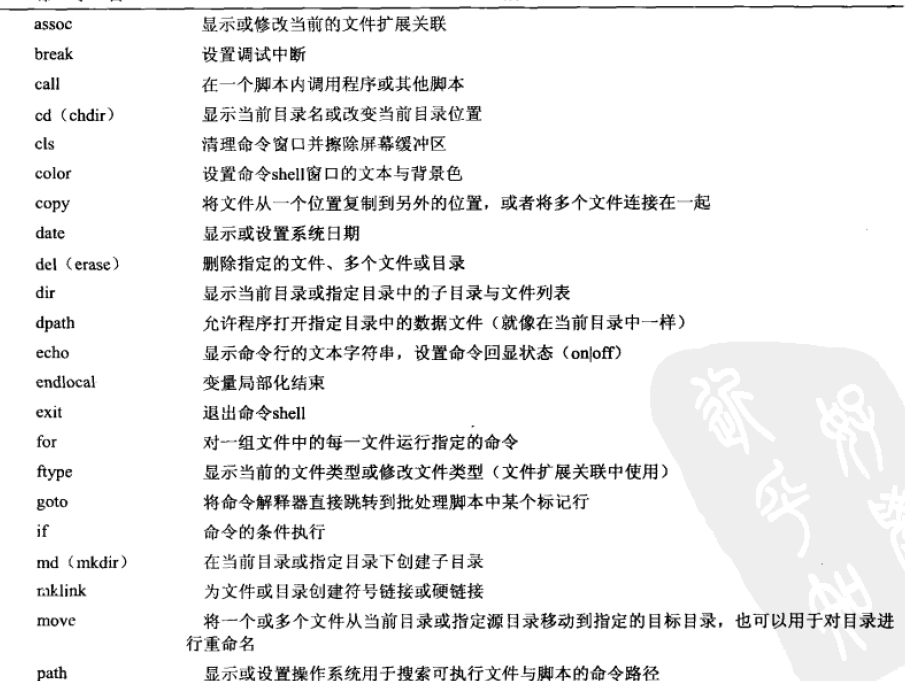
|  |
| --- |
|  |

|  |
| --- |
|  |

## Windows命令行说明:

Cmd(shell)命令的说明:使用快捷键ctrl+s可以暂停正在运行的程序再次使用ctrl+s程序运行:使用ctrl+c终止程序的运行

Cmd的内部命令:



可以在命令后输入 /? 来显示命令的帮助:如copy /? 显示copy 的帮助

Windows下的另一种shell:powershell:在开始-中搜索powershell 然后回车就可以使用powershell

## 如何设置word的默认输入法——搜狗输入法

|  |
| --- |
| **可开Word时 默认的是微软的输入法 把微软输入法删了 默认的成了英文了 怎么改成搜狗啊 大家是不是有这个困惑呢 其实这个事情很好解决 只需简单几步就可以把word的默认输入法改成你想要的**  **我用的是office2010  2007的设置基本一致 我用的输入法是搜狗拼音 所以这里就以它为例**  **首先 要把系统的默认输入法设置为搜狗（怎么设置就不用我多说了吧）然后随便打开一个word文档 或新建一个word文档 然后单击左上角的“文件” （07是单击“office”按钮）然后单击“选项”如图**  **http://hi.csdn.net/attachment/201109/5/0_1315208426c2Z5.gif**  **此时弹出一个对话框 如图**  **http://hi.csdn.net/attachment/201109/5/0_131520852517tO.gif**  **单击“选项” 看到下图**  **http://hi.csdn.net/attachment/201109/5/0_1315208706ERHH.gif**  **将“输入法控制处于活动状态”前面的勾去掉 然后一路 “确定” 然后重新启动word 再看看屏幕右下方的是不是已经变成下面这个样子了呢**  **http://hi.csdn.net/attachment/201109/5/0_1315208905z28L.gif** |

## 浏览器地址栏：突然不见了~~

|  |
| --- |
| **方法一：**  **F12** - ESC - "window.location.href"  方法二：  ctrl+h ：就可以打开浏览器的历史记录 |

## 远程桌面的问题：（登录不了QQ）

|  |
| --- |
| 远程桌面连接时候发现等不了qq但是在进程中有QQ的进程 很奇怪：结果发现用户中有多个用户存在：其实QQ是运行在另一个远程桌面的界面而已：如果想登上就把另一个远程桌面注销就可以了 |

## Win7的限速功能解除

|  |
| --- |
| 运行--cmd回车-在该工作窗口的DOS提示符下，输入字符串命令  netsh int tcp set global autotuninglevel=disable  回车，显示“成功”象征着此时系统的TCP/IP协议的自动调谐功能已被成功关闭 |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# 服务器:tomcat

## java.lang.OutOfMemoryError：PermGen space 或者Java heap space的产生原因以及解决办法

|  |
| --- |
| 这里以tomcat环境为例，其它WEB服务器如jboss,weblogic等是同一个道理。  一、java.lang.OutOfMemoryError: PermGen space  PermGen space的全称是Permanent Generation space,是指内存的永久保存区域,  这块内存主要是被JVM存放Class和Meta信息的,Class在被Loader时就会被放到PermGen space中,  它和存放类实例(Instance)的Heap区域不同,GC(Garbage Collection)不会在主程序运行期对  PermGen space进行清理，所以如果你的应用中有很多CLASS的话,就很可能出现PermGen space错误,  这种错误常见在web服务器对JSP进行pre compile的时候。如果你的WEB APP下都用了大量的第三方jar, 其大小  超过了jvm默认的大小(4M)那么就会产生此错误信息了。  解决方法： 手动设置MaxPermSize大小  修改TOMCAT\_HOME/bin/catalina.sh  在“echo "Using CATALINA\_BASE: $CATALINA\_BASE"”上面加入以下行：  JAVA\_OPTS="-server -XX:PermSize=64M -XX:MaxPermSize=128m  建议：将相同的第三方jar文件移置到tomcat/shared/lib目录下，这样可以达到减少jar 文档重复占用内存的目的。  二、java.lang.OutOfMemoryError: Java heap space  Heap size 设置  JVM堆的设置是指java程序运行过程中JVM可以调配使用的内存空间的设置.JVM在启动的时候会自动设置Heap size的值，  其初始空间(即-Xms)是物理内存的1/64，最大空间(-Xmx)是物理内存的1/4。可以利用JVM提供的-Xmn -Xms -Xmx等选项可  进行设置。Heap size 的大小是Young Generation 和Tenured Generaion 之和。  提示：在JVM中如果98％的时间是用于GC且可用的Heap size 不足2％的时候将抛出此异常信息。  提示：Heap Size 最大不要超过可用物理内存的80％，一般的要将-Xms和-Xmx选项设置为相同，而-Xmn为1/4的-Xmx值。  解决方法：手动设置Heap size  修改TOMCAT\_HOME/bin/catalina.sh  在“echo "Using CATALINA\_BASE: $CATALINA\_BASE"”上面加入以下行：  JAVA\_OPTS="-server -Xms800m -Xmx800m -XX:MaxNewSize=256m"  三、实例，以下给出1G内存环境下java jvm 的参数设置参考：  JAVA\_OPTS="-server -Xms800m -Xmx800m -XX:PermSize=64M -XX:MaxNewSize=256m -XX:MaxPermSize=128m -Djava.awt.headless=true "  大家都知道，JAVA程序启动时都会JVM都会分配一个初始内存和最大内存给这个应用程序。这个初始内存和最大内存在一定程度都会影响程序的性能。比如说在应用程序用到最大内存的时候，JVM是要先去做垃圾回收的动作，释放被占用的一些内存。  所以想调整Tomcat的启动时初始内存和最大内存就需要向JVM声明，一般的JAVA程序在运行都可以通过中-Xms -Xmx来调整应用程序的初始内存和最大内存:  如：java -Xms64m -Xmx128m a.jar.  tomcat的启动程序是包装过的，不能直接使用java -X..... tomcat.\*来改变内存的设置。在Tomcat在改变这个设置  有两种方法：  1. 就需要在环境变量中加上TOMCAT\_OPTS, CATALINA\_OPTS两个属性，  如 SET CATALINA\_OPTS= -Xms64m -Xmx512m;  ms是最小的，mx是最大，64m, 512m分别是指内存的容量.  2. 修改Catalina.bat文件  在166行“rem Execute Java with the applicable properties ”以下每行  %\_EXECJAVA% %JAVA\_OPTS% %CATALINA\_OPTS% %DEBUG\_OPTS% -Djava.endorsed.dirs="%JAVA\_ENDORSED\_DIRS%" -classpath "%CLASSPATH%" -Dcatalina.base="%CATALINA\_BASE%" -Dcatalina.home="%CATALINA\_HOME%" -Djava.io.tmpdir="%CATALINA\_TMPDIR%" %MAINCLASS% %CMD\_LINE\_ARGS% %ACTION% 中的%CATALINA\_OPTS% 替换成  -Xms64m -Xmx512m |

## [**Tomcat version 6.0 only supports J2EE 1.2, 1.3, 1.4, and Java EE 5 Web modules**](http://www.cnblogs.com/chanedi/articles/2112477.html)

最近在用eclipse做项目，新建项目时什么都贪新，用最新的版本，在Dynamic web module version栏里选了最新的3.0版本，布署项目的时候就出现了如期的错误，在网上看到[http://hi.baidu.com/yolanda441/b ... 41d36dd1164e2e.html](http://hi.baidu.com/yolanda441/blog/item/124b4cd9ea41d36dd1164e2e.html)这位兄弟的博客,解决了问题，现将方法贴于此，希望对别人有所帮助：  
  
《通过search，发现在project的.setting folder下面，有个名为org.eclipse.wst.common.project.facet.core.xml的文件，里面配置有各种版本信息。此时，按照本机配置修改这个文件，问题就解决了。  
  
<?xml version="1.0" encoding="UTF-8"?>  
<faceted-project>  
<runtime name="Apache Tomcat v5.5"/>  
<fixed facet="jst.web"/>  
<fixed facet="jst.java"/>  
<installed facet="jst.java" version="5.0"/>  
<installed facet="jst.web" version="2.4"/>  
<installed facet="wst.jsdt.web" version="1.0"/>  
</faceted-project>  
》  
  
像我这样的错误，可以下载Tomcat 7。0解决，不过我闲麻烦，干脆就在配置文件中把<installed facet="jst.web" version="3.0"/> 改成低些的版本version="2.5" 就好了

## [关于The APR based Apache Tomcat Native library警告](http://jasonhan-sh-hotmail-com.iteye.com/blog/1472911)

  今天在Eclipse上配置Tomcat7，启动时看到如下警告信息：

The APR based Apache Tomcat Native library which allows optimal performance in production environments was not found on the java.library.path

    感到很不舒服，于是在网上找了些解决办法。

    以下内容来自：http://cooliron.blog.163.com/blog/static/124703138201112211571276/

    在eclipse中，启动tomcat时会出现如此信息：The APR based Apache Tomcat Native library which allows optimal performance in production environments was not found on the java.library.path，它其实不影响程序的正常运行，但看到了，感觉实在别扭。其解决办法如下：

<http://tomcat.heanet.ie/native/> 下载tcnative-1.dll，然后放到你的JDK路径下的bin目录下。重新启动tomcat，就没有问题了。

    这个tcnative-1.dll 他的作用如下：

    tomcat整合本地apr会使效率提升：  
    一是，处理静态资源的时候速度更快，（注：有人用jmeter，对使用apr前后的tomcat进行压力测试，结果显示，性能是有一定的提升，但是没有想象中多，只有一点点，当然，这与运行的逻辑有关，但是我个人认为在大多数情况下这部分性能提升甚微）。  
    二是，我认为提升比较多的是对ssl的处理效率，当tomcat处理https的请求是，如果使用本地的openssl库，肯定会比前面提升的效率高。（虽然没有实际测试数据，但我推测这里的性能提升比例上要比第一点多得多）。

    原因上面已经说得很清楚了，下面说下我的解决办法。

    首先，tcnative-1.dll不需要下载，至少对于Tomcat7来说是这样，我们可以在其bin目录下找到这个dll；

    其次，tcnative-1.dll放在哪儿不要紧，只要在系统的PATH中可以找到就行。

   解决：

   在系统的环境变量中添加CATALINA\_HOME，它指向你的Tomcat根目录，接着在PATH路径中添加%CATALINA\_HOME%\bin;

   这样就可以解决了，对了，设置完成后，可能需要重启下Eclipse。

## [eclipse下启动tomcat出现Setting property 'source' to 'org.eclipse.jst.jee.server: '错误的解决办法](http://www.cnblogs.com/tonyzzp/archive/2010/04/11/eclipse_tomcat_setting_property_source_solution.html)

|  |
| --- |
| 在eclipse中启动tomcat时出现Setting property 'source' to 'org.eclipse.jst.jee.server:你的站点名'   did not find a matching property错误    解决办法：  1、在server控制台内，在服务器上点右键--属性  2、general选项卡中点switch location  这时，location变为:/servers/tomcat6.0 server at localhost.server  3、在project explore中找到tomcat项目--Tomcat v6.0 Server at localhost.server  这个文件 ，双击打开  4、在最下面的server option里选中publis module context to separate xml file.保存。（保存时要先停止服务器） |

## Tomcat部署项目发布的时候报错：Could not publish server configuration for Tomcat v6.0 Server at localhost. Multiple Contexts have a path of "/jeecms".

|  |
| --- |
| 可能原因： 重新定义的tomcat和eclpse关联的路径  或者是在tomcat中定义了多个相同的项目：清理一下tomcat下的路径 |

## 项目的部署：（直接使用WAR的形式部署项目）

|  |
| --- |
| 使用.war文件包  其实前面三种我们一直都是将qyzygl文件部署在服务器中，其实我们可以将应用程序打包成.war包，然后再部署在服务器上。打包步骤如下：  打开cmd命令提示符。  在命令提示框中进入D:/creator/workspace/qyzygl文件中，然后输入如下命令：jar cvf qyzygl.war \*/.然后提示框会出现非常多的类似于xxx写入之类的，这个过程就是在将qyzygl文件中的内容打包成.war文件，完成之后会在该目录下生成qyzygl.war文件。  部署.war文件非常简单，将.war文件拷贝到webapps文件路径下或者将docBase=” D:/creator/workspace/qyzygl/WebRoot”更改为docBase="F:\qyzygl.war"即可。重写启动服务就可以完成部署了。 |

## <context-param>与<init-param>的区别与作用

|  |
| --- |
| <context-param>的作用:  web.xml的配置中<context-param>配置作用  1. 启动一个WEB项目的时候,容器(如:Tomcat)会去读它的配置文件web.xml.读两个节点: <listener></listener> 和 <context-param></context-param>  2.紧接着,容器创建一个ServletContext(上下文),这个WEB项目所有部分都将共享这个上下文.  3.容器将<context-param></context-param>转化为键值对,并交给ServletContext.  4.容器创建<listener></listener>中的类实例,即创建监听.  5.在监听中会有contextInitialized(ServletContextEvent args)初始化方法,在这个方法中获得ServletContext = ServletContextEvent.getServletContext();  context-param的值 = ServletContext.getInitParameter("context-param的键");  6.得到这个context-param的值之后,你就可以做一些操作了.注意,这个时候你的WEB项目还没有完全启动完成.这个动作会比所有的Servlet都要早.  换句话说,这个时候,你对<context-param>中的键值做的操作,将在你的WEB项目完全启动之前被执行.  7.举例.你可能想在项目启动之前就打开数据库.  那么这里就可以在<context-param>中设置数据库的连接方式,在监听类中初始化数据库的连接.  8.这个监听是自己写的一个类,除了初始化方法,它还有销毁方法.用于关闭应用前释放资源.比如说数据库连接的关闭.  如:  <!-- 加载spring的配置文件 -->  <context-param>  <param-name>contextConfigLocation</param-name>  <param-value>/WEB-INF/applicationContext.xml,/WEB-INF/action-servlet.xml,/WEB-  INF/jason-servlet.xml</param-value>  </context-param>  <listener>  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>  </listener>  又如: --->自定义context-param,且自定义listener来获取这些信息  <context-param>  <param-name>urlrewrite</param-name>  <param-value>false</param-value>  </context-param>  <context-param>  <param-name>cluster</param-name>  <param-value>false</param-value>  </context-param>  <context-param>  <param-name>servletmapping</param-name>  <param-value>\*.bbscs</param-value>  </context-param>  <context-param>  <param-name>poststoragemode</param-name>  <param-value>1</param-value>  </context-param>  <listener>  <listener-class>com.laoer.bbscs.web.servlet.SysListener</listener-class>  </listener>  public class SysListener extends HttpServlet implements ServletContextListener {  private static final Log logger = LogFactory.getLog(SysListener.class);  public void contextDestroyed(ServletContextEvent sce) {  //用于在容器关闭时,操作  }  //用于在容器开启时,操作  public void contextInitialized(ServletContextEvent sce) {  String rootpath = sce.getServletContext().getRealPath("/");  System.out.println("-------------rootPath:"+rootpath);  if (rootpath != null) {  rootpath = rootpath.replaceAll("\\\\", "/");  } else {  rootpath = "/";  }  if (!rootpath.endsWith("/")) {  rootpath = rootpath + "/";  }  Constant.ROOTPATH = rootpath;  logger.info("Application Run Path:" + rootpath);  String urlrewrtie = sce.getServletContext().getInitParameter("urlrewrite");  boolean burlrewrtie = false;  if (urlrewrtie != null) {  burlrewrtie = Boolean.parseBoolean(urlrewrtie);  }  Constant.USE\_URL\_REWRITE = burlrewrtie;  logger.info("Use Urlrewrite:" + burlrewrtie);  其它略之....  }  }  /\*最终输出  -------------rootPath:D:\tomcat\_bbs\webapps\BBSCS\_8\_0\_3\  2009-06-09 21:51:46,526 [com.laoer.bbscs.web.servlet.SysListener]-[INFO]  Application Run Path:D:/tomcat\_bbs/webapps/BBSCS\_8\_0\_3/  2009-06-09 21:51:46,526 [com.laoer.bbscs.web.servlet.SysListener]-[INFO]  Use Urlrewrite:true  2009-06-09 21:51:46,526 [com.laoer.bbscs.web.servlet.SysListener]-[INFO]  Use Cluster:false  2009-06-09 21:51:46,526 [com.laoer.bbscs.web.servlet.SysListener]-[INFO]  SERVLET MAPPING:\*.bbscs  2009-06-09 21:51:46,573 [com.laoer.bbscs.web.servlet.SysListener]-[INFO]  Post Storage Mode:1  \*/  context-param和init-param区别  web.xml里面可以定义两种参数：  (1)application范围内的参数，存放在servletcontext中，在web.xml中配置如下：  <context-param>  <param-name>context/param</param-name>  <param-value>avalible during application</param-value>  </context-param>  (2)servlet范围内的参数，只能在servlet的init()方法中取得，在web.xml中配置如下：  <servlet>  <servlet-name>MainServlet</servlet-name>  <servlet-class>com.wes.controller.MainServlet</servlet-class>  <init-param>  <param-name>param1</param-name>  <param-value>avalible in servlet init()</param-value>  </init-param>  <load-on-startup>0</load-on-startup>  </servlet>  在servlet中可以通过代码分别取用：  package com.wes.controller;  import javax.servlet.ServletException;  import javax.servlet.http.HttpServlet;  public class MainServlet extends HttpServlet ...{  public MainServlet() ...{  super();  }  public void init() throws ServletException ...{  System.out.println("下面的两个参数param1是在servlet中存放的");  System.out.println(this.getInitParameter("param1"));  System.out.println("下面的参数是存放在servletcontext中的");  System.out.println(getServletContext().getInitParameter("context/param"));  }  }  第一种参数在servlet里面可以通过getServletContext().getInitParameter("context/param")得到  第二种参数只能在servlet的init()方法中通过this.getInitParameter("param1")取得. |

Web.xml部署描述符中的load-on-startup的作用

|  |
| --- |
| 1)load-on-startup元素标记容器是否在启动的时候就加载这个servlet(实例化并调用其init()方法)。  2)它的值必须是一个整数，表示servlet应该被载入的顺序  2)当值为0或者大于0时，表示容器在应用启动时就加载并初始化这个servlet；  3)当值小于0或者没有指定时，则表示容器在该servlet被选择时才会去加载。  4)正数的值越小，该servlet的优先级越高，应用启动时就越先加载。  5)当值相同时，容器就会自己选择顺序来加载。  所以，<load-on-startup>x</load-on-startup>，中x的取值1，2，3，4，5代表的是优先级，而非启动延迟时间。 |

## Tomcat的优化设置set Xmx1024m的说明：

|  |
| --- |
| 可以使用命令在cmd中 java –Xmx1024m –version 如果能正常显示版本信息说明设置的数值符合你的本地计算机（看到1400m就没有显示 是不适合的）而且设置1400m的时候tomcat会起不来（命令窗口一闪而过）  网上摘抄说明： |
| 即JVM内存设置了，把Xms与Xmx两个值设成一样是最优的做法，有人说Xms为最小值，Xmx为最大值不是挺好的，这样设置还比较人性化，科学化。人性？科学？你个头啊。  大家想一下这样的场景：  一个系统随着并发数越来越高，它的内存使用情况逐步上升，上升到最高点不能上升了，开始回落，你们不要认为这个回落就是好事情，由其是大起大落，在内存回落时它付出的代价是CPU高速开始运转进行垃圾回收，此时严重的甚至会造成你的系统出现“卡壳”就是你在好好的操作，突然网页像死在那边一样几秒甚至十几秒时间，因为JVM正在进行垃圾回收。  因此一开始我们就把这两个设成一样，使得Tomcat在启动时就为最大化参数充分利用系统的效率，这个道理和jdbcconnection pool里的minpool size与maxpool size的需要设成一个数量是一样的原理。  如何知道我的JVM能够使用最大值啊？拍脑袋？不行！  在设这个最大内存即Xmx值时请先打开一个命令行，键入如下的命令：  http://my.csdn.net/uploads/201207/02/1341213790_9461.jpg  看，能够正常显示JDK的版本信息，说明，这个值你能够用。不是说32位系统下最高能够使用2GB内存吗？即：2048m，我们不防来试试  http://my.csdn.net/uploads/201207/02/1341213804_6990.jpg  可以吗？不可以！不要说2048m呢，我们小一点，试试1700m如何  http://my.csdn.net/uploads/201207/02/1341213818_7402.jpg  嘿嘿，连1700m都不可以，更不要说2048m了呢，2048m只是一个理论数值，这样说吧我这边有几台机器，有的机器-Xmx1800都没问题，有的机器最高只能到-Xmx1500m。  因此在设这个-Xms与-Xmx值时一定一定记得先这样测试一下，要不然直接加在tomcat启动命令行中你的tomcat就再也起不来了，要飞是飞不了，直接成了一只瘟猫了。  使用的参数说明： |
| ü   –Xmn  设置年轻代大小为512m。整个堆大小=年轻代大小 + 年老代大小 + 持久代大小。持久代一般固定大小为64m，所以增大年轻代后，将会减小年老代大小。此值对系统性能影响较大，Sun官方推荐配置为整个堆的3/8。  ü   -Xss  是指设定每个线程的堆栈大小。这个就要依据你的程序，看一个线程 大约需要占用多少内存，可能会有多少线程同时运行等。一般不易设置超过1M，要不然容易出现out ofmemory。  ü   -XX:+AggressiveOpts  作用如其名（aggressive），启用这个参数，则每当JDK版本升级时，你的JVM都会使用最新加入的优化技术（如果有的话）  ü   -XX:+UseBiasedLocking  启用一个优化了的线程锁，我们知道在我们的appserver，每个http请求就是一个线程，有的请求短有的请求长，就会有请求排队的现象，甚至还会出现线程阻塞，这个优化了的线程锁使得你的appserver内对线程处理自动进行最优调配。  ü   -XX:PermSize=128M-XX:MaxPermSize=256M  JVM使用-XX:PermSize设置非堆内存初始值，默认是物理内存的1/64；  在数据量的很大的文件导出时，一定要把这两个值设置上，否则会出现内存溢出的错误。  由XX:MaxPermSize设置最大非堆内存的大小，默认是物理内存的1/4。  那么，如果是物理内存4GB，那么64分之一就是64MB，这就是PermSize默认值，也就是永生代内存初始大小；  四分之一是1024MB，这就是MaxPermSize默认大小。  ü   -XX:+DisableExplicitGC  在程序代码中不允许有显示的调用”System.gc()”。看到过有两个极品工程中每次在DAO操作结束时手动调用System.gc()一下，觉得这样做好像能够解决它们的out ofmemory问题一样，付出的代价就是系统响应时间严重降低，就和我在关于Xms,Xmx里的解释的原理一样，这样去调用GC导致系统的JVM大起大落，性能不到什么地方去哟！  ü   -XX:+UseParNewGC  对年轻代采用多线程并行回收，这样收得快。  ü   -XX:+UseConcMarkSweepGC  即CMS gc，这一特性只有jdk1.5即后续版本才具有的功能，它使用的是gc估算触发和heap占用触发。  我们知道频频繁的GC会造面JVM的大起大落从而影响到系统的效率，因此使用了CMS GC后可以在GC次数增多的情况下，每次GC的响应时间却很短，比如说使用了CMS GC后经过jprofiler的观察，GC被触发次数非常多，而每次GC耗时仅为几毫秒。  ü   -XX:MaxTenuringThreshold  设置垃圾最大年龄。如果设置为0的话，则年轻代对象不经过Survivor区，直接进入年老代。对于年老代比较多的应用，可以提高效率。如果将此值设置为一个较大值，则年轻代对象会在Survivor区进行多次复制，这样可以增加对象再年轻代的存活时间，增加在年轻代即被回收的概率。  这个值的设置是根据本地的jprofiler监控后得到的一个理想的值，不能一概而论原搬照抄。  ü   -XX:+CMSParallelRemarkEnabled  在使用UseParNewGC 的情况下, 尽量减少 mark 的时间  ü   -XX:+UseCMSCompactAtFullCollection  在使用concurrent gc 的情况下, 防止 memoryfragmention, 对live object 进行整理, 使 memory 碎片减少。  ü   -XX:LargePageSizeInBytes  指定 Java heap的分页页面大小  ü   -XX:+UseFastAccessorMethods  get,set 方法转成本地代码  ü   -XX:+UseCMSInitiatingOccupancyOnly  指示只有在 oldgeneration 在使用了初始化的比例后concurrent collector 启动收集  ü   -XX:CMSInitiatingOccupancyFraction=70  CMSInitiatingOccupancyFraction，这个参数设置有很大技巧，基本上满足**(Xmx-Xmn)\*(100- CMSInitiatingOccupancyFraction)/100>=Xmn**就不会出现promotion failed。在我的应用中Xmx是6000，Xmn是512，那么Xmx-Xmn是5488兆，也就是年老代有5488 兆，CMSInitiatingOccupancyFraction=90说明年老代到90%满的时候开始执行对年老代的并发垃圾回收（CMS），这时还 剩10%的空间是5488\*10%=548兆，所以即使Xmn（也就是年轻代共512兆）里所有对象都搬到年老代里，548兆的空间也足够了，所以只要满 足上面的公式，就不会出现垃圾回收时的promotion failed；  因此这个参数的设置必须与Xmn关联在一起。  ü   -Djava.awt.headless=true  这个参数一般我们都是放在最后使用的，这全参数的作用是这样的，有时我们会在我们的J2EE工程中使用一些图表工具如：jfreechart，用于在web网页输出GIF/JPG等流，在winodws环境下，一般我们的app server在输出图形时不会碰到什么问题，但是在linux/unix环境下经常会碰到一个exception导致你在winodws开发环境下图片显示的好好可是在linux/unix下却显示不出来，因此加上这个参数以免避这样的情况出现。  上述这样的配置，基本上可以达到：  ü   系统响应时间增快  ü   JVM回收速度增快同时又不影响系统的响应率  ü   JVM内存最大化利用  ü   线程阻塞情况最小化 |
| Tomcat容器内的优化 前面我们对Tomcat启动时的命令进行了优化，增加了系统的JVM可使用数、垃圾回收效率与线程阻塞情况、增加了系统响应效率等还有一个很重要的指标，我们没有去做优化，就是吞吐量。  还记得我们在第三天的学习中说的，这个系统本身可以处理1000，你没有优化和配置导致它默认只能处理25。因此下面我们来看Tomcat容器内的优化。  打开tomcat安装目录\conf\server.xml文件，定位到这一行：   |  | | --- | | <Connector port="8080" protocol="HTTP/1.1" |   这一行就是我们的tomcat容器性能参数设置的地方，它一般都会有一个默认值，这些默认值是远远不够我们的使用的，我们来看经过更改后的这一段的配置：   |  | | --- | | <Connector port="8080" protocol="HTTP/1.1"            URIEncoding="UTF-8"  minSpareThreads="25" maxSpareThreads="75"            enableLookups="false" disableUploadTimeout="true" connectionTimeout="20000"            acceptCount="300"  maxThreads="300" maxProcessors="1000" minProcessors="5"            useURIValidationHack="false"                                                 compression="on" compressionMinSize="2048"                                                 compressableMimeType="text/html,text/xml,text/javascript,text/css,text/plain"                 redirectPort="8443"  /> |   好大一陀唉。。。。。。  没关系，一个个来解释  ü   URIEncoding=”UTF-8”  使得tomcat可以解析含有中文名的文件的url，真方便，不像apache里还有搞个mod\_encoding，还要手工编译  ü   maxSpareThreads  maxSpareThreads 的意思就是如果空闲状态的线程数多于设置的数目，则将这些线程中止，减少这个池中的线程总数。  ü   minSpareThreads  最小备用线程数，tomcat启动时的初始化的线程数。  ü   enableLookups  这个功效和Apache中的HostnameLookups一样，设为关闭。  ü   connectionTimeout  connectionTimeout为网络连接超时时间毫秒数。  ü   maxThreads  maxThreads Tomcat使用线程来处理接收的每个请求。这个值表示Tomcat可创建的最大的线程数，即最大并发数。    ü   acceptCount  acceptCount是当线程数达到maxThreads后，后续请求会被放入一个等待队列，这个acceptCount是这个队列的大小，如果这个队列也满了，就直接refuse connection    ü   maxProcessors与minProcessors  在 Java中线程是程序运行时的路径，是在一个程序中与其它控制线程无关的、能够独立运行的代码段。它们共享相同的地址空间。多线程帮助程序员写出CPU最 大利用率的高效程序，使空闲时间保持最低，从而接受更多的请求。  通常Windows是1000个左右，Linux是2000个左右。  ü   useURIValidationHack  我们来看一下tomcat中的一段源码：   |  | | --- | | security          if (connector.getUseURIValidationHack()) {              String uri = validate(request.getRequestURI());              if (uri == null) {                  res.setStatus(400);                  res.setMessage("Invalid URI");                  throw new IOException("Invalid URI");              } else {                  req.requestURI().setString(uri);                  // Redoing the URI decoding                  req.decodedURI().duplicate(req.requestURI());                  req.getURLDecoder().convert(req.decodedURI(), true);              }          } |   可以看到如果把useURIValidationHack设成"false"，可以减少它对一些url的不必要的检查从而减省开销。  ü   enableLookups="false"  为了消除DNS查询对性能的影响我们可以关闭DNS查询，方式是修改server.xml文件中的enableLookups参数值。  ü   disableUploadTimeout  类似于Apache中的keeyalive一样  ü   给Tomcat配置gzip压缩(HTTP压缩)功能   |  | | --- | | compression="on" compressionMinSize="2048"  compressableMimeType="text/html,text/xml,text/javascript,text/css,text/plain" |   HTTP 压缩可以大大提高浏览网站的速度，它的原理是，在客户端请求网页后，从服务器端将网页文件压缩，再下载到客户端，由客户端的浏览器负责解压缩并浏览。相对于普通的浏览过程HTML,CSS,Javascript , Text ，它可以节省40%左右的流量。更为重要的是，它可以对动态生成的，包括CGI、PHP , JSP , ASP , Servlet,SHTML等输出的网页也能进行压缩，压缩效率惊人。  1)compression="on" 打开压缩功能  2)compressionMinSize="2048" 启用压缩的输出内容大小，这里面默认为2KB  3)noCompressionUserAgents="gozilla, traviata" 对于以下的浏览器，不启用压缩  4)compressableMimeType="text/html,text/xml"　压缩类型  最后不要忘了把8443端口的地方也加上同样的配置，因为如果我们走https协议的话，我们将会用到8443端口这个段的配置，对吧？   |  | | --- | | <!--enable tomcat ssl-->      <Connector port="8443" protocol="HTTP/1.1"                 URIEncoding="UTF-8"  minSpareThreads="25" maxSpareThreads="75"            enableLookups="false" disableUploadTimeout="true" connectionTimeout="20000"            acceptCount="300"  maxThreads="300" maxProcessors="1000" minProcessors="5"            useURIValidationHack="false"                      compression="on" compressionMinSize="2048"                      compressableMimeType="text/html,text/xml,text/javascript,text/css,text/plain"                  SSLEnabled="true"             scheme="https" secure="true"             clientAuth="false" sslProtocol="TLS"             keystoreFile="d:/tomcat2/conf/shnlap93.jks" keystorePass="aaaaaa"        /> |   好了，所有的Tomcat优化的地方都加上了。结合第三天中的Apache的性能优化，我们这个架构可以“飞奔”起来了，当然这边把有提及任何关于数据库优化的步骤，但仅凭这两步，我们的系统已经有了很大的提升。  举个真实的例子：上一个项目，经过4轮performance testing，第一轮进行了问题的定位，第二轮就是进行了apache+tomcat/weblogic的优化，第三轮是做集群优化，第四轮是sql与codes的优化。  在到达第二轮时，我们的性能已经提升了多少倍呢？我们来看一个loaderrunner的截图吧：  http://img.my.csdn.net/uploads/201212/12/1355287840_8767.jpghttp://img.my.csdn.net/uploads/201212/12/1355287845_8264.jpg  左边第一列是第一轮没有经过任何调优的压力测试报告。  右边这一列是经过了apache优化，tomcat优化后得到的压力测试报告。  大家看看，这就提高了多少倍？这还只是在没有改动代码的情况下得到的改善，现在明白了好好的调优一  个apache和tomcat其实是多么的重要了？如果加上后面的代码、SQL的调优、数据库的调优。。。。。。所以我在上一个工程中有单笔交易性能（无论是吞吐量、响应时间）提高了80倍这样的极端例子的存在。 |

## Tomcat修改端口号,用户名,以及一些简单说明

|  |
| --- |
| Tomcat的默认的端口是8080：所以在tomcat的目录下的bin中运行startup.bat的时候如果正常启动:就可以在浏览器中敲入:localhost:8080看到tomcat的欢迎界面修改端口在tomcat下的conf中的service.xml中 |
| <Connector port="8080" protocol="HTTP/1.1"  connectionTimeout="20000"  redirectPort="8443" /> |
| 设置用户名 |
| <role rolename="admin-gui"/>  <user username="tomcat" password="s3cret" roles="admin-gui"/> |
| 在tomcat中默认启动的root中的项目(即:localhost:8080/Root)要想启动其他项目:(其中的红色就是加入到Host中的 docBse是项目名称 (这样在使用localhost:8080登录的时候就相当于localhost:8080/examples而不是Root那个项目了 |
| <Host name="localhost" appBase="webapps"  unpackWARs="true" autoDeploy="true">  <Context path="" docBase="examples" reloadable="true" crossContext="true"/>  </Context>  <!-- SingleSignOn valve, share authentication between web applications  Documentation at: /docs/config/valve.html -->  <!--  <Valve className="org.apache.catalina.authenticator.SingleSignOn" />  -->    <!-- Access log processes all example.  Documentation at: /docs/config/valve.html  Note: The pattern used is equivalent to using pattern="common" -->  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"  prefix="localhost\_access\_log" suffix=".txt"  pattern="%h %l %u %t "%r" %s %b" />    </Host> |
| <Host name="localhost" debug="0" appBase="绝对路径" unpackWARs="true">  <Context docBase="相对于appBase的目录" path="访问路径" reloadable="true">  举例：  <Host name="localhost" debug="0" appBase="d:/myapp" unpackWARs="true">  <Context docBase="example" path="/demo" reloadable="true">  启动服务器后输入：http://localhost:8080/demo,tomat加载的是d:/myapp/example/index.jsp  如果path=""，那么直接输入http://localhost:8080就可以了 |

## Session过期:session的优先级:

|  |
| --- |
| 在tomcat的conf下的web.xml中配置  <session-config>  <session-timeout>30</session-timeout>  </session-config>Tomcat默认session超时时间为30分钟，可以根据需要修改，负数或0为不限制session失效时间。  2.在工程的web.xml中设置  <session-config>  <session-timeout>15</session-timeout>  </session-config>  3. 通过java代码设置  session.setMaxInactiveInterval（30\*60）;  三种方式优先级：1 < 2 <3 |

## Tomcat开启SSL认证支持https链接:

|  |
| --- |
| 1. 创建 keystore 文件  执行 keytool -genkey -alias tomcat -keyalg RSA 结果如下  loiane:bin loiane$ keytool -genkey -alias tomcat -keyalg RSA  Enter keystore password: password  Re-enter new password: password  What is your first and last name?  [Unknown]: Loiane Groner  What is the name of your organizational unit?  [Unknown]: home  What is the name of your organization?  [Unknown]: home  What is the name of your City or Locality?  [Unknown]: Sao Paulo  What is the name of your State or Province?  [Unknown]: SP  What is the two-letter country code for this unit?  [Unknown]: BR  Is CN=Loiane Groner, OU=home, O=home, L=Sao Paulo, ST=SP, C=BR correct?  [no]: yes    Enter key password for  (RETURN if same as keystore password): password  Re-enter new password: password  这样就在用户的主目录下创建了一个 .keystore 文件  2. 配置 Tomcat 以使用 keystore 文件  打开 server.xml 找到下面被注释的这段  <!--  <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"  maxThreads="150" scheme="https" secure="true"  clientAuth="false" sslProtocol="TLS" />  -->  干掉注释，并将内容改为  Connector SSLEnabled="true" acceptCount="100" clientAuth="false"  disableUploadTimeout="true" enableLookups="false" maxThreads="25"  port="8443" keystoreFile="/Users/loiane/.keystore" keystorePass="password"  protocol="org.apache.coyote.http11.Http11NioProtocol" scheme="https"  secure="true" sslProtocol="TLS" />  3. 测试  启动 Tomcat 并访问 https://localhost:8443. 你将看到 Tomcat 默认的首页。  需要注意的是，如果你访问默认的 8080 端口，还是有效的。  4. 配置应用使用 SSL  打开应用的 web.xml 文件，增加配置如下：  <security-constraint>  <web-resource-collection>  <web-resource-name>securedapp</web-resource-name>  <url-pattern>/\*</url-pattern>  </web-resource-collection>  <user-data-constraint>  <transport-guarantee>CONFIDENTIAL</transport-guarantee>  </user-data-constraint>  </security-constraint>  将 URL 映射设为 /\* ，这样你的整个应用都要求是 HTTPS 访问，而 transport-guarantee 标签设置为 CONFIDENTIAL 以便使应用支持 SSL。  如果你希望关闭 SSL ，只需要将 CONFIDENTIAL 改为 NONE 即可 |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# Java

## [微软面试题——反转字符串](http://www.cnblogs.com/k-eckel/articles/195931.html)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 这是网络流传的Microsoft的面试题目之一：“编写反转字符串的程序，要求优化速度、优化空间”。因为最近一直很多关注算法方面的实践和研究，因此对这个问题进行了一些思考，给出了5种实现方法（有两种解法相关性比较大）。  解法一：第一次看到这题目，想到最简单、最直觉的解法就是：遍历字符串，将第一个字符和最后一个交换，第二个和倒数第二个交换，依次循环，即可，于是有了第一个解法：   |  | | --- | | char\* strrev1(const char\* str)  {         int len = strlen(str);         char\* tmp = new char[len + 1];         strcpy(tmp,str);           for (int i = 0; i < len/2; ++i)         {                char c = tmp[i];          tmp[i] = tmp[len – i - 1];          tmp[len – i - 1] = c;         }           return tmp;  } |   这里是通过数组的下标方式访问字符串的字符，实际上用指针直接操作即可。解法二正是基于此，实现代码为：   |  | | --- | | char\* strrev2(const char\* str)  {         char\* tmp = new char[strlen(str) + 1];         strcpy(tmp,str);         char\* ret = tmp;           char\* p = tmp + strlen(str) - 1;           while (p > tmp)         {                char t = \*tmp;                \*tmp = \*p;                \*p = t;                  --p;                ++tmp;         }           return ret;  } |   显然上面的两个解法中没有考虑时间和空间的优化，一个典型的优化策略就是两个字符交换的算法优化，我们可以完全不使用任何外部变量即完成两个字符（或者整数）的交换，这也是一个很经典的面试题目。特别是一些嵌入式硬件相关编程中经常要考虑寄存器的使用，因此经常有不使用任何第三个寄存器即完成两个寄存器数据的交换的题目。一般有两个解法，对应这里的解法三和解法四。  解法三的实现代码为：   |  | | --- | | char\* strrev3(const char\* str)  {         char\* tmp = new char[strlen(str) + 1];         strcpy(tmp,str);         char\* ret = tmp;           char\* p = tmp + strlen(str) - 1;           while (p > tmp)         {                \*p ^= \*tmp;                \*tmp ^= \*p;                \*p ^= \*tmp;                  --p;                ++tmp;         }           return ret;  } |   解法四的实现代码为：   |  | | --- | | char\* strrev4(const char\* str)  {         char\* tmp = new char[strlen(str) + 1];         strcpy(tmp,str);         char\* ret = tmp;           char\* p = tmp + strlen(str) - 1;           while (p > tmp)         {                \*p = \*p + \*tmp;                \*tmp = \*p - \*tmp;                \*p = \*p - \*tmp;                  --p;                ++tmp;         }           return ret;  } |   实际上我们还可以通过递归的思想来解决这个问题，思想很简单：每次交换首尾两个字符，中间部分则又变为和原来字符串同样的问题，因此可以通过递归的思想来解决这个问题，对应解法五的实现代码为：   |  | | --- | | char\* strrev5(/\*const \*/char\* str,int len)  {         if (len <= 1)                return str;           char t = \*str;         \*str = \*(str + len -1);         \*(str + len -1) = t;           return (strrev5(str + 1,len - 2) - 1);  } |   以下给出一个测试程序：   |  | | --- | | int main(int argc,char\* argv[])  {         char\* str = "hello";         P(str);           char\* str2 = strrev1(str);         P(str2);           char\* str3 = strrev2(str2);         P(str3);           char\* str4 = strrev3(str3);         P(str4);           char\* str5 = strrev4(str4);         P(str5);           char\* str6 = strrev5(str5,strlen(str5));         P(str6);           return 0;  } |     你就可以看到字符串"hello"和"olleh"交替输出了。    说明：1）这里解法中没有认真考虑输入字符串的合法性和特殊长度（如NULL、一个字符等）字符串的处理；2）前4个算法不改变输入字符串的值，解法五修改了输入字符串。 |

## J ava字符串反转的三种实现方法

|  |
| --- |
| public class StringTest{  public void convert1(String s){    for(int i= s.length()-1;i>=0;i--){     System.out.print(s.charAt(i));    }    System.out.println (); }  public void convert2(String s){    char[] str = s.toCharArray();    for(int i=str.length;i>0;i--){      System.out.print(s.charAt(i-1));    }      System.out.println ();  }  public void convert3(String s){    StringBuffer sb = new StringBuffer(s);    System.out.println (sb.reverse().toString());     }  public static void main(String[] args){    String s = "农民工集体被袭,谁来保障他们.";    StringTest st = new StringTest();    System.out.println ("发转前的字符串是：");     System.out.println (s);     System.out.println ("发转后的字符串是：");     st.convert1(s);    st.convert2(s);    st.convert3(s); } } |

## Java中的instanceof关键字

|  |
| --- |
| instanceof是Java的一个二元操作符，和==，>，<是同一类东东。由于它是由字母组成的，所以也是Java的保留关键字。它的作用是测试它左边的对象是否是它右边的类的实例，返回boolean类型的数据。举个例子：  　　String s = "I AM an Object!"; 　　boolean isObject = s instanceof Object;  　　我们声明了一个String对象引用，指向一个String对象，然后用instancof来测试它所指向的对象是否是Object类的一个实例，显然，这是真的，所以返回true，也就是isObject的值为True。 　　instanceof有一些用处。比如我们写了一个处理账单的系统，其中有这样三个类：  　　public class Bill {//省略细节} 　　public class PhoneBill extends Bill {//省略细节} 　　public class GasBill extends Bill {//省略细节}  　　在处理程序里有一个方法，接受一个Bill类型的对象，计算金额。假设两种账单计算方法不同，而传入的Bill对象可能是两种中的任何一种，所以要用instanceof来判断：  　　public double calculate(Bill bill) { 　　if (bill instanceof PhoneBill) { 　　//计算电话账单 　　} 　　if (bill instanceof GasBill) { 　　//计算燃气账单 　　} 　　... 　　} 　　这样就可以用一个方法处理两种子类。  　　然而，这种做法通常被认为是没有好好利用面向对象中的多态性。其实上面的功能要求用方法重载完全可以实现，这是面向对象变成应有的做法，避免回到结构化编程模式。只要提供两个名字和返回值都相同，接受参数类型不同的方法就可以了：  　　public double calculate(PhoneBill bill) { 　　//计算电话账单 　　}  　　public double calculate(GasBill bill) { 　　//计算燃气账单 　　}  　　所以，使用instanceof在绝大多数情况下并不是推荐的做法，应当好好利用多态。 |

## [Java的native关键字](http://blog.csdn.net/hlantian/article/details/4268666)

|  |
| --- |
| java native关键字   一. 什么是Native Method    简单地讲，一个Native Method就是一个java调用非java代码的接口。一个Native Method是这样一个java的方法：该方法的实现由非java语言实现，比如C。这个特征并非java所特有，很多其它的编程语言都有这一机制，比如在C＋＋中，你可以用extern "C"告知C＋＋编译器去调用一个C的函数。    "A native method is a Java method whose implementation is provided by non-java code."    在定义一个native method时，并不提供实现体（有些像定义一个java interface），因为其实现体是由非java语言在外面实现的。，下面给了一个示例：  package java.lang;  public class Object {      ......      public final native Class<?> getClass();      public native int hashCode();      protected native Object clone() throws CloneNotSupportedException;      public final native void notify();      public final native void notifyAll();      public final native void wait(long timeout) throws InterruptedException;      ......  }          标识符native可以与所有其它的java标识符连用，但是abstract除外。这是合理的，因为native暗示这些方法是有实现体的，只不过这些实现体是非java的，但是abstract却显然的指明这些方法无实现体。native与其它java标识符连用时，其意义同非Native Method并无差别。      一个native method方法可以返回任何java类型，包括非基本类型，而且同样可以进行异常控制。这些方法的实现体可以自制一个异常并且将其抛出，这一点与java的方法非常相似。     native method的存在并不会对其他类调用这些本地方法产生任何影响，实际上调用这些方法的其他类甚至不知道它所调用的是一个本地方法。JVM将控制调用本地方法的所有细节。      如果一个含有本地方法的类被继承，子类会继承这个本地方法并且可以用java语言重写这个方法（这个似乎看起来有些奇怪），同样的如果一个本地方法被fianl标识，它被继承后不能被重写。    本地方法非常有用，因为它有效地扩充了jvm.事实上，我们所写的java代码已经用到了本地方法，在sun的java的并发（多线程）的机制实现中，许多与操作系统的接触点都用到了本地方法，这使得java程序能够超越java运行时的界限。有了本地方法，java程序可以做任何应用层次的任务。  二.为什么要使用Native Method    java使用起来非常方便，然而有些层次的任务用java实现起来不容易，或者我们对程序的效率很在意时，问题就来了。    与java环境外交互：    有时java应用需要与java外面的环境交互。这是本地方法存在的主要原因，你可以想想java需要与一些底层系统如操作系统或某些硬件交换信息时的情况。本地方法正是这样一种交流机制：它为我们提供了一个非常简洁的接口，而且我们无需去了解java应用之外的繁琐的细节。    与操作系统交互：    JVM支持着java语言本身和运行时库，它是java程序赖以生存的平台，它由一个解释器（解释字节码）和一些连接到本地代码的库组成。然而不管怎样，它毕竟不是一个完整的系统，它经常依赖于一些底层（underneath在下面的）系统的支持。这些底层系统常常是强大的操作系统。通过使用本地方法，我们得以用java实现了jre的与底层系统的交互，甚至JVM的一些部分就是用C写的，还有，如果我们要使用一些java语言本身没有提供封装的操作系统的特性时，我们也需要使用本地方法。     Sun's Java     Sun的解释器是用C实现的，这使得它能像一些普通的C一样与外部交互。jre大部分是用java实现的，它也通过一些本地方法与外界交互。例如：类java.lang.Thread 的 setPriority()方法是用java实现的，但是它实现调用的是该类里的本地方法setPriority0()。这个本地方法是用C实现的，并被植入JVM内部，在Windows 95的平台上，这个本地方法最终将调用Win32 SetPriority() API。这是一个本地方法的具体实现由JVM直接提供，更多的情况是本地方法由外部的动态链接库（external dynamic link library）提供，然后被JVM调用。  三.JVM怎样使Native Method跑起来：     我们知道，当一个类第一次被使用到时，这个类的字节码会被加载到内存，并且只会回载一次。在这个被加载的字节码的入口维持着一个该类所有方法描述符的list，这些方法描述符包含这样一些信息：方法代码存于何处，它有哪些参数，方法的描述符（public之类）等等。     如果一个方法描述符内有native，这个描述符块将有一个指向该方法的实现的指针。这些实现在一些DLL文件内，但是它们会被操作系统加载到java程序的地址空间。当一个带有本地方法的类被加载时，其相关的DLL并未被加载，因此指向方法实现的指针并不会被设置。当本地方法被调用之前，这些DLL才会被加载，这是通过调用java.system.loadLibrary()实现的。       最后需要提示的是，使用本地方法是有开销的，它丧失了java的很多好处。如果别无选择，我们可以选择使用本地方法。    可以将native方法比作Java程序同Ｃ程序的接口，其实现步骤： １、在Java中声明native()方法，然后编译； ２、用javah产生一个.h文件； ３、写一个.cpp文件实现native导出方法，其中需要包含第二步产生的.h文件（注意其中又包含了JDK带的jni.h文件）； ４、将第三步的.cpp文件编译成动态链接库文件； ５、在Java中用System.loadLibrary()方法加载第四步产生的动态链接库文件，这个native()方法就可以在Java中被访问了。    具体实现方法可以从网上查找，这里就不写了  不过，又引出两个东西：javah.exe命令和JNI    自己写了一个调用VB的DLL的例子，不过还没写完，调用函数部分不会写，只写了加载DLL的过程  JAVA部分：  public class TestNativeMothod {      public static void main(String[] args) {                    //PROCESS LOGIC                }            public native String callNativeMothod();            static {           System.loadLibrary("TestNativeMethodProj");      }  } VB部分：  Option Explicit  Public Function callNativeMothod() As String      callNativeMothod = "This is a method implemented by VB"  End Function |

## 乱码处理:

|  |
| --- |
| 1.JSP里CHARSET要为GB2312    contentType="text/html;charset=GB2312"  2.SERVERLET类里要有    request.setCharacterEncoding(gb2312);  3.字符集的重新格式化  java.net.URLEncoder  java.net.URLDecoder 对应的encode 与decode进行编码解码。  在传输数据时候浏览器会自动对要表单传输的数据进行url编码，所使用的编码方式取决于当前网页显示时候使用的编码方式。  对于http请求消息的url地址后的参数，getparameter等方法进行自动url解码时采用的编码方式取决servlet引擎；tomcat默认用ISO8859-1进行解码。  重新格式化语句如下：  String str1 = new String(request.getParameter("name").getBytes("ISO-8859-1"),"gb2312");  4.TOMCAT的CONF文件夹里的SERVER.XML中大约92行左右.(如果你没改过)   找到connector区块，加入如下一行：  URIEncoding="GBK" 或 URIEncoding="GB2312" 或 URIEncoding="UTF-8"    完整的应如下：  <Connector     port="80" maxThreads="150" minSpareThreads="25" maxSpareThreads="75"     enableLooku[ps教程](http://www.111cn.net/fw/photo.html)="false" redirectPort="8443" acceptCount="100"     debug="0" connectionTimeout="20000"     disableUploadTimeout="true"     URIEncoding="GB2312"    />  5.Tomcat的conf文件夹下的web.xml文件的<servlet></servlet>标签之间添加如下代码  <init-param>    <param-name>encoding</param-name>    <param-value>GB2312</param-value>  </init-param>  6.使用Servlet规范中的过虑器指定编码  <1>.新创建一个servlet时，Interfaces接口：继承javax.servlet.Filter        options中只需选中：create Inherited Methods; create Constructors;        建好后在其doFilter()方法中：  arg0.setCharacterEncoding("gb2312"); //实现请求乱码的处理  arg1.setCharacterEncoding("gb2312"); //实现响应乱码的处理  7.解决[数据库教程](http://www.111cn.net/database/database.html)的乱码  在jdbc.url中明确字符集 (不适用于sqlserver)  在连接数据库时：con=DriverManager.getConnection(url,"sa","");  url="jdbc:协议：子协议：//IP：端口; 库 ? useUnicode=true & characterEncoding=gb2312"; |
| 我遇到的乱码我的解决:  1. String encoding = System.getProperty("file.encoding");  // System.out.println("系统默认的字符集是:" + encoding);  String getCon= dn.get("content").toString();//获取请求的字符  // String content=new String(getCon.getBytes("ISO-8859-1"),encoding); |
| 2.使用过滤器:  a.过滤器的配置：  <filter>  <description>charset to utf8</description>  <display-name>charset to utf8</display-name>  <filter-name>CharSetE</filter-name>  <filter-class>com.hygj.filter.CharSetE</filter-class>  <init-param>  <description>encode to UTF-8</description><!-- //UTF-8表示中文编码 -->  <param-name>encoding</param-name>  <param-value>UTF-8</param-value>  </init-param>  </filter>  <filter-mapping>  <filter-name>CharSetE</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>  B》package com.hygj.filter;  import java.io.IOException;  import java.io.PrintWriter;  import javax.servlet.Filter;  import javax.servlet.FilterChain;  import javax.servlet.FilterConfig;  import javax.servlet.ServletException;  import javax.servlet.ServletRequest;  import javax.servlet.ServletResponse;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  public class CharSetE implements Filter {    private FilterConfig filterConfig;  private String trargetEncoding=null;  public FilterConfig getFilterConfig() {  return filterConfig;  }  public void setFilterConfig(FilterConfig filterConfig) {  this.filterConfig = filterConfig;  }  public String getTrargetEncoding() {  return trargetEncoding;  }  public void setTrargetEncoding(String trargetEncoding) {  this.trargetEncoding = trargetEncoding;  }  @Override  public void destroy() {  this.filterConfig=null;  this.trargetEncoding=null;//清空资源  }  @Override  public void doFilter(ServletRequest request, ServletResponse response,  FilterChain chain) throws IOException, ServletException {  String encodes=System.getProperty("file.encoding");//获得系统编码  try {  System.out.println("目标编码： "+trargetEncoding);//在部署描述符中的编码//可以在部署描述符中修改：  request.setCharacterEncoding(encodes);//这里选择设置系统默认的编码  chain.doFilter(request, response);  } catch (Exception e) {  filterConfig.getServletContext().log(e.getMessage());  e.printStackTrace();  }      }  @Override  public void init(FilterConfig filterConfig) throws ServletException {  // TODO Auto-generated method stub  this.filterConfig=filterConfig;  this.trargetEncoding=this.filterConfig.getInitParameter("encoding");  }  } |

## String intern()方法

|  |
| --- |
| 1. new产生的string在堆中，而不在字符串池中，即CONSTANT\_String\_info中。  2. intern()去字符串池中查找是否存在该内容为指定字符串的string对象，如果有，则返回一个已有的引用，否则，创建在字符串池中创建新的引用，并返回该引用。  3. 采用new 创建的字符串对象不进入字符串池，字符串相加的时候，都是静态字符串的结果会添加到字符串池，如果其中含有变量不会进入字符串池中。但是字符串一旦进入字符串池中，就会先查找池中有无此对象。如果有此对象，则让对象引用指向此对象。如果无此对象，则先创建此对象，再让对象引用指向此对象。  注意理解字符串池和堆中字符串的区别。 |

## JVM发生OOM时，自动生成DUMP文件。

|  |
| --- |
| 1、配置方法  在JAVA\_OPTIONS变量中增加  -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=${目录}。// -XX:HeapDumpPath=E:\Java\dump  2、参数说明  （1）-XX:+HeapDumpOnOutOfMemoryError参数表示当JVM发生OOM时，自动生成DUMP文件。  （2）-XX:HeapDumpPath=${目录}参数表示生成DUMP文件的路径，也可以指定文件名称，例如：-XX:HeapDumpPath=${目录}/java\_heapdump.hprof。如果不指定文件名，默认为：java\_<pid>\_<date>\_<time>\_heapDump.hprof。 |

Java类型转换String→date

|  |
| --- |
| 用SimpleDateFormat来转换  SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  Date date = sdf.parse("2008-08-08 12:10:12"); |

## java 10大常见异常

|  |
| --- |
| NO.1　java.lang.NullPointerExceptio　　这个异常大家肯定都经常遇到，异常的解释是 "程序遇上了空指针 "，简单地说就是调用了未经初始化的对象或者是不存在的对象，这个错误经常出现在创建图片，调用数组这些操作中，比如图片未经初始化，或者图片创建时的路径错误等等。对数组操作中出现空指针，很多情况下是一些刚开始学习编程的朋友常犯的错误，即把数组的初始化和数组元素的初始化混淆起来了。数组的初始化是对数组分配需要的空间，而初始化后的数组，其中的元素并没有实例化，依然是空的，所以还需要对每个元素都进行初始化（如果要调用的话）  NO.2　java.lang.ClassNotFoundException  　　这个异常是很多原本在JB等开发环境中开发的程序员，把JB下的程序包放在WTk下编译经常出现的问题，异常的解释是 "指定的类不存在 "，这里主要考虑一下类的名称和路径是否正确即可，如果是在JB下做的程序包，一般都是默认加上Package的，所以转到WTK下后要注意把Package的路径加上。  NO.3　java.lang.ArithmeticException  　　这个异常的解释是 "数学运算异常 "，比如程序中出现了除以零这样的运算就会出这样的异常，对这种异常，大家就要好好检查一下自己程序中涉及到数学运算的地方，公式是不是有不妥了。  NO.4　java.lang.ArrayIndexOutOfBoundsException　　这个异常相信很多朋友也经常遇到过，异常的解释是 "数组下标越界 "，现在程序中大多都有对数组的操作，因此在调用数组的时候一定要认真检查，看自己调用的下标是不是超出了数组的范围，一般来说，显示（即直接用常数当下标）调用不太容易出这样的错，但隐式（即用变量表示下标）调用就经常出错了，还有一种情况，是程序中定义的数组的长度是通过某些特定方法决定的，不是事先声明的，这个时候，最好先查看一下数组的length，以免出现这个异常。  NO.5　java.lang.IllegalArgumentException  　　这个异常的解释是 "方法的参数错误 "，很多J2ME的类库中的方法在一些情况下都会引发这样的错误，比如音量调节方法中的音量参数如果写成负数就会出现这个异常，再比如g.setColor(int   red,int   green,int   blue)这个方法中的三个值，如果有超过２５５的也会出现这个异常，因此一旦发现这个异常，我们要做的，就是赶紧去检查一下方法调用中的参数传递是不是出现了错误。  NO.6　java.lang.IllegalAccessException  　　这个异常的解释是 "没有访问权限 "，当应用程序要调用一个类，但当前的方法即没有对该类的访问权限便会出现这个异常。对程序中用了Package的情况下要注意这个异常。  NO.7　java.lang.IncompatibleClassChangeError          不兼容的类变化错误。当正在执行的方法所依赖的类定义发生了不兼容的改变时，抛出该异常。一般在修改了应用中的某些类的声明定义而没有对整个应用重新编译而直接运行的情况下，容易引发该错误。   NO.8　java.lang.InstantiationError          实例化错误。当一个应用试图通过Java的new操作符构造一个抽象类或者接口时抛出该异常.    NO.9　java.lang.LinkageError          链接错误。该错误及其所有子类指示某个类依赖于另外一些类，在该类编译之后，被依赖的类改变了其类定义而没有重新编译所有的类，进而引发错误的情况。  NO.10   java.lang.StackOverflowError         堆栈溢出错误。当一个应用递归调用的层次太深而导致堆栈溢出时抛出该错误。 |

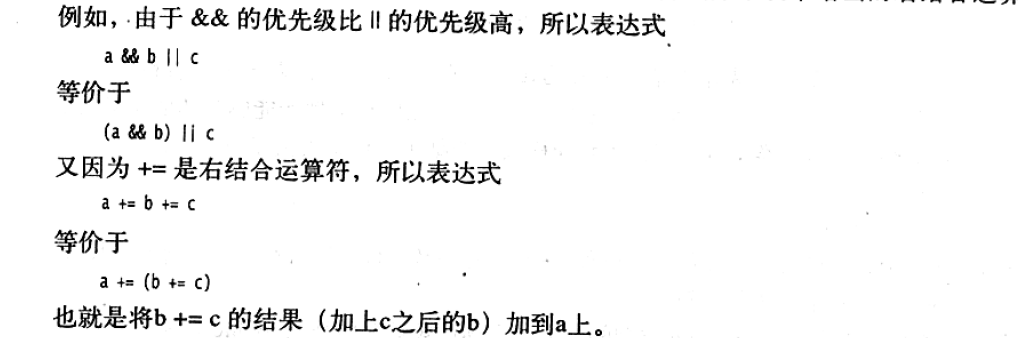
## java打开文件夹

|  |
| --- |
| 一、调用windows系统函数 Runtime.getRuntime().exec("cmd  /c start h:\\movie"); 或者如下 Runtime.getRuntime().exec(new String[] {"cmd","/c","start","","h:/movie"}); 注：这里/c是指定的，因为系统是装在c盘在。 二、直接利用java api java.awt.Desktop.getDesktop().open(new File("D:\\Java")); |

## setlocal和endlocal命令执行结果是让中间的程序对于系统变量的改变只在程序内起作用，不会影响整个系统级别

|  |
| --- |
| @echo off  setlocal  path=d:\  echo 局部环境变量path值  set path  endlocal  echo 系统环境变量path的值  set path |

## 运算符优先级：



## 动态绑定 vs 静态绑定

|  |
| --- |
| 动态绑定(又名后期绑定)  动态绑定是指编译器在编译阶段不知道要调用哪个方法，直到运行时才能确定。让我们用个例子来解释。譬如我们有一个叫作’SuperClass’的父类，还有一个继承它的子类’SubClass’。现在SuperClass引用也可以赋给SubClass类型的对象。如果SuperClass中有个someMethod()的方法，而子类也重写了这个方法，那么当调用SuperClass引用的这个方法的时候，编译器不知道该调用父类还是子类的方法，因为编译器不知道对象到底是什么类型，只有到运行时才知道这个引用指向什么对象。  ...  SuperClass superClass1 = new SuperClass();  SuperClass superClass2 = new SubClass();  ...    superClass1.someMethod(); // SuperClass version is called  superClass2.someMethod(); // SubClass version is called  ....    我们可以看到虽然对象引用superClass1和superClass2都是SuperClass类型的，但是在运行时它们分别指向SuperClass和SubClass类型的对象。  所以在编译阶段，编译器不清楚调用引用的someMethod()到底是调用子类还是父类的该方法。  所以方法的动态绑定是基于实际的对象类型，而不是它们声明的对象引用类型。  静态绑定(又名前期绑定)  如果编译器可以在编译阶段就完成绑定，就叫作静态绑定或前期绑定。基本上实例方法都在运行时绑定，所有的静态方法都在编译时绑定，所以静态方法是静态绑定的。因为静态方法是属于类的方法，可以通过类名来访问(我们也应该使用类名来访问静态方法，而不是使用对象引用来访问)，所以要访问它们就必须在编译阶段就使用编译类型信息来进行绑定。这也就解释了为什么静态方法实际上不能被重写。  更多阅读：可以重写静态方法吗？  类似的，访问成员变量也是静态绑定的，因为Java不支持(实际上是不鼓励)成员变量的多态行为。下面看个例子：  class SuperClass{  ...  public String someVariable = "Some Variable in SuperClass";  ...  }    class SubClass extends SuperClass{  ...  public String someVariable = "Some Variable in SubClass";  ...  }  ...  ...    SuperClass superClass1 = new SuperClass();  SuperClass superClass2 = new SubClass();    System.out.println(superClass1.someVariable);  System.out.println(superClass2.someVariable);  ...    输出：  Some Variable in SuperClass  Some Variable in SuperClass    我们可以发现成员变量由对象引用声明的类型决定，是由编译器在编译阶段就知道的信息，所以是静态绑定。另外一个静态绑定的例子是私有的方法，因为它们不会被继承，编译器在编译阶段就完成私有方法的绑定了。 |

## Static的说明：

|  |
| --- |
| 2.1、static变量  static修饰的变量我们称之为静态变量，没有用static修饰的变量称之为实例变量，他们两者的区别是：  静态变量是随着类加载时被完成初始化的，它在内存中仅有一个，且JVM也只会为它分配一次内存，同时类所有的实例都共享静态变量，可以直接通过类名来访问它。  但是实例变量则不同，它是伴随着实例的，每创建一个实例就会产生一个实例变量，它与该实例同生共死。  所以我们一般在这两种情况下使用静态变量：对象之间共享数据、访问方便。  2.2、static方法  static修饰的方法我们称之为静态方法，我们通过类名对其进行直接调用。由于他在类加载的时候就存在了，它不依赖于任何实例，所以static方法必须实现，也就是说他不能是抽象方法abstract。  Static方法是类中的一种特殊方法，我们只有在真正需要他们的时候才会将方法声明为static。如Math类的所有方法都是静态static的。  2.3、static代码块  被static修饰的代码块，我们称之为静态代码块，静态代码块会随着类的加载一块执行，而且他可以随意放，可以存在于该了的任何地方。  三、 Static的局限  Static确实是存在诸多的作用，但是它也存在一些缺陷。  1、它只能调用static变量。  2、它只能调用static方法。  3、不能以任何形式引用this、super。  4、static变量在定义时必须要进行初始化，且初始化时间要早于非静态变量。  总结：无论是变量，方法，还是代码块，只要用static修饰，就是在类被加载时就已经"准备好了",也就是可以被使用或者已经被执行，都可以脱离对象而执行。反之，如果没有static，则必须要依赖于对象实例。 |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# Log4j

## log4j:WARN No appenders could be found for logger

|  |
| --- |
| 在Eclipse中开发相关项目时,在控制台经常看到如下信息:  log4j:WARN No appenders could be found for logger  log4j:WARN Please initialize the log4j system properly.          此处输出信息并不是错误信息而仅只是警告信息,因为log4j无法输出日志,log4j是一个日志输入软件包。可以将Struts或Hibernate等压缩包解压，内有log4j.properties文件，将它复制到项目src文件夹或将log4j.properties放到 \WEB-INF\classes文件夹中即可。  或者：在spring的web项目中常常会在tomcat启动的时候出现这种提示：  网上有好多解决办法都不管用，这个提示应该是读入web应用程序的log4j.properties文件之前就报出来了。是在加载org.springframework.web.context.ContextLoader这个listener的时候没找到log4j的配置文件造成的。   仔细查看web.xml发现在加载org.springframework.web.context.ContextLoader这个listener之后才加载org.springframework.web.util.Log4jConfigListener，把log4j的配置放到org.springframework.web.context.ContextLoader之前，就可以解决这个问题了。  <!-- 以下3项参数与log4j的配置相关 -->    <context-param>  <param-name>log4jConfigLocation</param-name>  <param-value>/WEB-INF/log4j.properties</param-value>  </context-param>    <context-param>  <param-name>log4jRefreshInterval</param-name>  <param-value>60000</param-value>  </context-param>  <listener>  <listener-class>  org.springframework.web.util.Log4jConfigListener  </listener-class>  </listener>  <!-- end -->  <listener>  <listener-class>  org.springframework.web.context.ContextLoaderListener  </listener-class>  </listener> |

## ClassPathXmlApplicationContext[只能读放在web-info/classes目录下的配置文件]和FileSystemXmlApplicationContext的区别

|  |
| --- |
| classpath:前缀是不需要的,默认就是指项目的classpath路径下面; 如果要使用绝对路径,需要加上file:前缀表示这是绝对路径;  对于FileSystemXmlApplicationContext: 默认表示的是两种:  1.没有盘符的是项目工作路径,即项目的根目录; 2.有盘符表示的是文件绝对路径.  如果要使用classpath路径,需要前缀classpath: |

## 使用log4j把项目的日志输出到文件：

|  |
| --- |
| log4j.rootLogger=INFO,R  #设定文件  log4j.appender.R=org.apache.log4j.RollingFileAppender  log4j.appender.R.File=output.log  log4j.appender.R.MaxFileSize=100KB  log4j.appender.R.MaxBackupIndex=1  log4j.appender.R.layout=org.apache.log4j.PatternLayout  log4j.appender.R.layout.ConversionPattern=%-d{yyyy-MM-dd HH:mm:ss} [%c]-[%p] %m%n  用这个配置就可以输出到output.log里面了 |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# 技术必知点

## MIME笔记

|  |
| --- |
| MIME笔记  1.  MIME的全称是"Multipurpose Internet Mail Extensions"，中译为"多用途互联网邮件扩展"，指的是一系列的电子邮件技术规范，主要包括RFC 2045、RFC 2046、RFC 2047、RFC 4288、RFC 4289和RFC 2077。  顾名思义，MIME是对传统电子邮件的一个扩展，现在已经成为电子邮件实际上的标准。  2.  传统的电子邮件是1982年定下技术规范的，文件是RFC 822。  它的一个重要特点，就是规定电子邮件只能使用ASCII字符。这导致了三个结果：1）非英语字符都不能在电子邮件中使用；2）电子邮件中不能插入二进制文件（如图片）；3）电子邮件不能有附件。  这实际上无法接受的，因此到了1992年，工程师们决定扩展电子邮件的技术规范，提出一系列补充规范，这就是MIME的由来。  3.  下面是一封传统的电子邮件。  From: "Tommy Lee" <lee@example.com>  To: "Jack Zhang" <zhang@example.com>  Subject: Test  Date: Wed, 17 May 2000 19:08:29 -0400  Message-ID: <NDBBIAKOPKHFGPLCODIGIEKBCHAA.lee@example.com>  Hello World.  从上面可以看出，这封信的发信人地址是lee@example.com，收信人地址是zhang@example.com，邮件主题是Test，发送时间是2000年5月17日，邮件内容是"Hello World."。  在结构上，这封信分为三个部分：首先是信件头，然后是一个空行，最后是信件内容。收信人的客户端软件只会显示最后一部分，要查看全信，必须使用"查看原始邮件"功能。  4.  MIME对传统电子邮件的扩展，表现在它在信件头部分添加了几条语句，主要有三条。  第一条是：  MIME-Version: 1.0  这条语句是必须的，而且1.0这个版本值是不变的，即使MIME本身已经升级了好几次。  有了这条语句，收信端就知道这封信使用了MIME规范。  5.  第二条语句是：  Content-Type: text/plain; charset="ISO-8859-1"  这一行是极端重要的，它表明传递的信息类型和采用的编码。  Content-Type表明信息类型，缺省值为" text/plain"。它包含了主要类型（primary type）和次要类型（subtype）两个部分，两者之间用"/"分割。主要类型有9种，分别是application、audio、example、image、message、model、multipart、text、video。  每一种主要类型下面又有许多种次要类型，常见的有：  text/plain：纯文本，文件扩展名.txt  text/html：HTML文本，文件扩展名.htm和.html  image/jpeg：jpeg格式的图片，文件扩展名.jpg  image/gif：GIF格式的图片，文件扩展名.gif  audio/x-wave：WAVE格式的音频，文件扩展名.wav  audio/mpeg：MP3格式的音频，文件扩展名.mp3  video/mpeg：MPEG格式的视频，文件扩展名.mpg  application/zip：PK-ZIP格式的压缩文件，文件扩展名.zip  详细的Content-Type列表，可以查看这里和这里。  如果信息的主要类型是"text"，那么还必须指明编码类型"charset"，缺省值是ASCII，其他可能值有"ISO-8859-1"、"UTF-8"、"GB2312"等等。  整个Content-Type这一行，不仅使用在电子邮件，后来也被移植到了HTTP协议中，所以现在只要是在网上传播的HTTP信息，都带有Content-Type头，以表明信息类型。  6.  前面已经说过，电子邮件的传统格式不支持非ASCII编码和二进制数据。因此MIME规定了第三条语句：  Content-transfer-encoding: base64  这条语句指明了编码转换的方式。Content-transfer-encoding的值有5种----"7bit"、"8bit"、"binary"、"quoted-printable"和"base64"----其中"7bit"是缺省值，即不用转化的ASCII字符。真正常用是"quoted-printable"和"base64"两种，它们的详细用法，我在明天的笔记中会详细介绍。  7.  下面是一封我收到的邮件的源码：  Date: Wed, 18 Jun 2008 18:07:51 +0800 (CST)  From: xxx <xxx@163.com>  To: yifeng.ruan@gmail.com  Message-ID: <14410503.1073611213783671983.JavaMail.coremail@bj163app54.163.com>  Subject: =?gbk?B?xOO6ww==?=  MIME-Version: 1.0  Content-Type: multipart/alternative;  boundary=&quot;----=\_Part\_287491\_22998031.1213783671982&quot;  ------=\_Part\_287491\_22998031.1213783671982  Content-Type: text/plain; charset=gbk  Content-Transfer-Encoding: base64  IAq4+b7dsr+209PQudi55raoo6yyu7XD1Nq12Le9yM66zs341b7Jz7nSz+DTprXEtqvO96Osx+vE  49TaxOO1xLKpv83W0AogIArW0Ln6yr2x6tPvIC0gyO7Su7fltcTN+MLnyNXWvgoKtcS12jEy1cXN  vMasyb6z/aOst/HU8s7Sw8fXt76/xOO1xM/gudjU8MjOoaPQu9C7us/X96OhtMvNvMas1Nq4vbz+  wO/D5g==  ------=\_Part\_287491\_22998031.1213783671982  Content-Type: text/html; charset=gbk  Content-Transfer-Encoding: quoted-printable  <DIV>&amp;nbsp;</DIV>  <DIV>=B8=F9=BE=DD=B2=BF=B6=D3=D3=D0=B9=D8=B9=E6=B6=A8=A3=AC=B2=BB=B5=C3=D4=  =DA=B5=D8=B7=BD=C8=CE=BA=CE=CD=F8=D5=BE=C9=CF=B9=D2=CF=E0=D3=A6=B5=C4=B6=AB=  =CE=F7=A3=AC=C7=EB=C4=E3=D4=DA=C4=E3=B5=C4=B2=A9=BF=CD=D6=D0</DIV>  <DIV>&amp;nbsp;  ......  可以看到这封信的MIME语句是：  MIME-Version: 1.0  Content-Type: multipart/alternative;  boundary="----=\_Part\_287491\_22998031.1213783671982"  "Content-Type: multipart/alternative;"表明这封信的内容，是纯文本和HTML文本的混合。另两个可能的值是multipart/mixed和multipart/related，分别表示"信件内容中有二进制内容"和"信件带有附件"。  "boundary="----=\_Part\_287491\_22998031.1213783671982"  "表明不同信件内容的分割线是"----=\_Part\_287491\_22998031.1213783671982"，它通常是一个很长的随机字符串。  信件内容部分又有两个子信件头：  Content-Type: text/plain; charset=gbk  Content-Transfer-Encoding: base64  和  Content-Type: text/html; charset=gbk  Content-Transfer-Encoding: quoted-printable  它们表明，第一个部分是gbk编码的纯文本，编码转换格式是base64。第二个部分是gbk编码的HTML文本，编码转化格式是quoted-printable。 |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# 数据库

## Oracle

### 查看oracel是32位还是64位的

|  |
| --- |
| 方法一：使用sqlplus 64位：  [oracle@qs-wg-db2 ~]$ sqlplus / as sysdba;    SQL\*Plus: Release 10.2.0.5.0 - Productionon Sun Sep 25 08:57:22 2011    Copyright (c) 1982, 2010, Oracle.  All Rights Reserved.      Connected to:  Oracle Database 10g Enterprise EditionRelease 10.2.0.5.0 - 64bit Production  With the Partitioning, OLAP, Data Miningand Real Application Testing options    SQL>    如果是64位，用sqlplus 连上之后会显示具体的位数信息，32位则不会显示。      32位：  C:\Users\Administrator.DavidDai>sqlplus/ as sysdba;    SQL\*Plus: Release 11.2.0.1.0 Production onSun Sep 25 08:55:48 2011    Copyright (c) 1982, 2010, Oracle.  All rights reserved.      Connected to:  Oracle Database 11g Enterprise EditionRelease 11.2.0.1.0 - Production  With the Partitioning, OLAP, Data Miningand Real Application Testing options     方法二： 查看v$version 视图 32位：  SQL> select \* from v$version;    BANNER  --------------------------------------------------------------------------  Oracle Database 11g Enterprise EditionRelease 11.2.0.1.0 - Production  PL/SQL Release 11.2.0.1.0 - Production  CORE   11.2.0.1.0      Production  TNS for 32-bit Windows: Version 11.2.0.1.0- Production  NLSRTL Version 11.2.0.1.0 - Production    64位：  SQL> select \* from v$version;    BANNER  ----------------------------------------------------------------  Oracle Database 10g Enterprise EditionRelease 10.2.0.5.0 - 64bi  PL/SQL Release 10.2.0.5.0 - Production  CORE   10.2.0.5.0      Production  TNS for Linux: Version 10.2.0.5.0 -Production  NLSRTL Version 10.2.0.5.0 – Production    和第一种方法一样，64位会显示具体的位数信息，32位不会显示。   方法三：查看v$sql 视图   32位：输出为8位16进制数  SQL> select address from v$sql whererownum<2;    ADDRESS  --------  B50ACCAC    64位：输出为16位16进制数  SQL> select address from v$sql whererownum<2;    ADDRESS  ----------------  0000000196FDF7D8 |

### 关于navicat连接oracle 报 ORA-12737 set CHS16GBK错误的解决方案

|  |
| --- |
| 昨天下载安装了navicat 9.0 Premium（高级英文版），连接mysql正常，看到他同时支持oracle、sqlite和postgresql，也想来连接一下oracle，进行集中管理。 1、但是连接时候先是报出监听错误、不能解析oracle SID标识符，我用sqlplus和plsql连接却是正常的！！！这说明我的配置文件是没有问题的。 2、重新将连接方式由TNS改为BASIC，然后依次填写连接信息，最后却爆出了“ORA-12737: Instant Client Light: unsupported server character set CHS16GBK”的错误。 3、确定是字符集问题后，google上查到类似的解决方案：   ORA-12737: Instant Client Light: unsupported server character set CHS16GBK  Google，解释如下： ORA-12737: Instant Client Light: unsupported server character set string Cause:     The character set specified is not allowed for this operation or is invalid. Instant Client Light has only minimal character sets. Action:     Do not use Instant Client Light for this character set  4、原因明确了之后，就要怎么解决了。修改服务器的字符集不太现实。后来研究了OCI，知道Instance Client 10是不支持除英文以外的编码的（如果你下载的是中文版的navicat可能不会有这个问题）。  5、去oracle官网下载最新版的OCI：http://www.oracle.com/technetwork/topics/winsoft-085727.html 6、同时在Oracle官网上发现Instance Client 11要比10大很多，下载完成后将解压出来的核心的oci11的3个文件替换掉原先的oci10的3个文件即可。(记得做个备份) 替换方法：在Navicat里，工具->选项->Miscellaneous->OCI。为oci.dll重新指定位置，然后重启navicat就OK了。 重启Navicat最后证明Instance Client 11没有字符集问题。 |

### oracle建立用户与授权

|  |
| --- |
| **创建表空间及用户：**  **create tablespace 表空间名 datafile 'd:/seal.dbf' size 10M autoextend on; create user username identified by password default tablespace 表空间名; grant connect,resource,dba to username;**    登录超级用户sqlplus sys@orcl as sysdba  #################################### # 建立用户和对用户的管理 ####################################  # 合法的Oracle标识符号       1、不能是保留关键字       2、必须使用1到30个字符。数据库的名称最多是8个字符。 3、必须以数据库字符集中的字母字符开头 4、只能够包含数据库字符集中的字母字符，以及以下字符：#、$、\_，另外数据库链接可以包含               @符号和'.'(句号) 5、不能包含引号  # 建立用户 create user angeos identified by angeos; 建立了用户：angeos，密码为：angeos  # 对用户授权 grant connect,resource to angeos; 对用户angeos授予了连接数据库和访问资源的权限  # 对用户授权 grant create session,dba to angeos; CREATE SESSION是一个系统特权，它可以为用户提供连接数据库的能力。 DBA是一个具有超过120个系统特权的角色，所以它可以让用户在数据库中完成几乎任何工作。  # 改变用户的密码 alter user angeos identified by oracle; 将用户angeos的密码改变为：oracle  # 锁定帐号以及解锁 alter user oe account unlock; 然后用用户oe登录数据库服务器，密码为oe 注意：用户解锁后，要重启服务。  # 修改表空间的设置 默认情况下，它会使用表空间SYSTEM和TEMP(用于存放临时数据)。 不推荐采用这种方法。所以我们需要改变表空间。  通过系统用户连接数据库服务器 conn [sys/sysadmin@hostname\_services](mailto:sys/sysadmin@hostname_services) as sysdba; 然后查看表空间 select tablespace\_name,contents from dba\_tablespaces order by tablespace\_name; 使用USER表空间代替SYSTEM表空间 alter user angeos default tablespace users temporary tablespace temp;  尝试建立一张表 create table table1 (      fieldA varchar2(10) )  # 删除用户angeos drop user angeos; 由于用户angeos有一张表table1,所以删除用户时，我们需要指定关键字CASCADE drop user angeos cascade;  # 解释术语模式Schema 数据库的模式定义为数据库对象的集合，而模式的名称就是拥有或者控制这个数据库对象集合的用户名称。 所有数据库对象，例如表、视图、索引、触发器、Java存储过程、PL/SQL程序包，函数等，都是由Oracle 数据库中的一个用户所拥有。甚至Oracle的数据词典，系统编目也是名为SYS的模式的一部分。  用户的传统解释就是能够唯一标识一组信任凭证的名称和密码组合。  # 解释系统特权 在Oracle数据库中有两类特权： （1）对象级别特权是由用户赋予的访问或者操作数据库对象的特权。 （2）系统特权不是控制对指定数据库对象的访问，而是用来许可对各种特性的访问，或者许可Oracle数据库中的          特定任务。       查询特权      connect system/sysadmin      desc dba\_sys\_privs;      查询             select distinct privilege from dba\_sys\_privs order by privilege       赋予系统特权的语句      GRANT system\_priviege to username [with admin option];      撤销系统特权的语句      REVOKE system\_priviege from username;       注意：在任何用户能够连接到Oracle数据库之前，都需要向它们赋予CREATE SESSION特权，为他们提供连接许可。  ################################## # 建立数据表 ##################################  # 语法规则 CREATE TABLE [SCHEMA.] <tablename>(      <coloumn\_name> <datatype> [default <expression>] [<constraint>]      [,<coloumn\_name> <datatype> [default <expression>] [<constraint>]]      [,...] );  # 建表的例子 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CREATE TABLE CD\_COLLECTION (      ALBUM\_TITLE     VARCHAR2(100),      ARTIST      VARCHAR2(100),      COUNTRY      VARCHAR2(25),      RELEASE\_DATE     DATE,      LABEL      VARCHAR2(25),      PRIMARY KEY (ALBUM\_TITLE, ARTIST) ); \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CREATE TABLE seagal.SONGS (      SONG\_TITLE      VARCHAR2(100),      COMPOSER     VARCHAR2(100),      LENGTH      NUMBER,      TRACK      NUMBER,      ARTIST      VARCHAR2(100),      ALBUM\_TITLE     VARCHAR2(100),      FOREIGN KEY (ARTIST, ALBUM\_TITLE) REFERENCES seagal.CD\_COLLECTION(ARTIST,         ALBUM\_TITLE),      PRIMARY KEY (SONG\_TITLE, ARTIST, ALBUM\_TITLE) );  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* select \* from seagal.cd\_collection; select \* from seagal.songs; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* INSERT INTO seagal.CD\_COLLECTION   VALUES('Black Sheets of Rain', 'Bob Mould', 'USA',to\_date('01-01-92','DD-MM-YY'), 'Virgin');  INSERT INTO seagal.CD\_COLLECTION  VALUES('Candy Apple Grey', 'Husker Du', 'USA',to\_date('01/01/86','DD/MM/YY'), 'Warner Brothers');  INSERT INTO seagal.SONGS  VALUES('Black Sheets of Rain','Mould', NULL,1, 'Bob Mould', 'Black Sheets of Rain');  INSERT INTO seagal.SONGS  VALUES('Crystal','Mould', 3.28, 1, 'Husker Du', 'Candy Apple Grey');  INSERT INTO seagal.SONGS  VALUES('Don''t want to know if you are lonely ','Hart', 3.28, 2, 'Husker Du', 'Candy Apple Grey');  INSERT INTO seagal.SONGS  VALUES('I don''t know for sure','Mould', 3.28, 3, 'Husker Du', 'Candy Apple Grey');  INSERT INTO SONGS VALUES('Black Sheets of Rain','Mould', NULL,1, 'Bob Mould', 'Black Sheets of Rain');  COMMIT; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DROP TABLE FRUITS;  CREATE TABLE FRUITS ( FRUIT VARCHAR2(12), COLOR VARCHAR2(12), QUANTITY NUMBER, PRICE NUMBER, PICKED DATE );  INSERT INTO FRUITS VALUES('Apple', 'Green', 12, 0.5, '12-Sep-2002'); INSERT INTO FRUITS VALUES('Apple', 'Red', 12, 0.5, '15-Sep-2002'); INSERT INTO FRUITS VALUES('Mango', 'Yellow', 10, 1.5, '22-Sep-2002'); INSERT INTO FRUITS VALUES('Mangosteen', 'Purple', 5, 2, '25-Sep-2002'); INSERT INTO FRUITS VALUES('Durian', 'NULL', 2, 15, NULL); INSERT INTO FRUITS VALUES('Orange', 'Orange', 10, 1.5, '28-Aug-2002');  COMMIT; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* COMMIT;  # CREATE TABLE AS SELECT的语句 通过查询一个表，并且将查询结果集物化到一个常规表中来建立表。 复制表的结构，但是约束、索引和触发器等对象不会被放入新的表中。  例子： create table emp\_copy as select \* from scott.emp;  # 数据词典 每一个数据库都有一个数据词典，任何管理Oracle数据库，或者使用Oracle构建应用的用户都需       要使用数据词典。 数据词典是Oracle数据库的编目。 当建立用户、表、约束和其它数据库对象的时候，Oracle都会自动维护一个在数据库中存储的项        目编目。  例如：USER\_TABLES视图可以展示当前用户所拥有的所有表的信息。      使用DESCRIBE命令查看USER\_TABLES视图的结构信息。  具有DBA权限的用户可以查看DBA\_TABLES select owner ,table\_name,tablespace\_name from dba\_tables where owner in('SCOTT','HR') order by owner,tablesapce\_name,table\_name |

### Oracle创建表空间 创建用户 授权

|  |
| --- |
| 1.创建表空间 |
| create tablespace test datafile 'D:\cache\sqlcache.dbf' size 50M default storage (initial 500k next 500k minextents 1 maxextents unlimited   pctincrease 0); |
| 2.创建用户并设置默认的表空间 |
| create user liting identified by liting123 default tablespace test; |
| 3.授权给新建的用户 |
| grant resource,connect,dba to test; |
| 删除表空间 |
| drop tablespace "空间名" including contents and datafiles |
| 删除用户 |
| drop user "lxg" cascade  alter tablespace chinawater add datafile 'c:\oracle\oradata\orcl9\ADDCHINAWATER.dbf' size 200M  创建用户  1. create user userName identified by password;  创建用户 userName,密码为 password  2  给用户授权  1. grant dba to lxg;--授予DBA权限  2. grant unlimited tablespace to lxg;--授予不限制的表空间  3. grant select any table to lxg;--授予查询任何表  4. grant select any dictionary to lxg;--授予 查询 任何字典  1. grant dba to lxg;  2. grant unlimited tablespace to lxg;  3. grant select any table to lxg;  4. grant select any dictionary to lxg; |

### sqlplus 远程连接oracle 命令

|  |
| --- |
| sqlplus usr/pwd@//host:port/sid |

### ora-123737:instant client light:unsupported server character set zhs16gbk 关于navicat连接oracle

|  |
| --- |
| 1、原因：  ORA-12737: Instant Client Light: unsupported server character set CHS16GBK  Google，解释如下：  ORA-12737:  Instant Client Light: unsupported server character set string  Cause: The character set specified is not allowed for this operation or is invalid. Instant Client Light has only minimal character sets.  Action: Do not use Instant Client Light for this character set  2、 首先，我们打开“tools”-->"options"菜单，见到如下界面，依据OCI library(oci.dll) 路径，导航到 navicat oci 目录下，备份里面的文件（通过在该目录新建bak文件夹，将“%dir%/Navicat Lite\instantclient\_10\_2”路径下的所有dll文件剪切到bak文件夹）。  http://img.blog.csdn.net/20130807100915843?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvamFyYXk=/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast  3、在oracle 客户端安装目录bin/目录下，找到如下几个.dll文件  上面的文件我只代替了四个：oci.dll，orannzsbb11.dll，oraocci11.dll，oraocr11.dll，下图：  http://img.blog.csdn.net/20130807101010343?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvamFyYXk=/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast  那些副本是替代前的原文件，这里的原因我想应该是oracle版本不同所引起的。我安装的是oracle 11g,而navcat是10版本  4、将上面所列文件放置到“%dir%/Navicat Lite\instantclient\_10\_2” 目录下，重新启动navcat即可。 |

### 解锁scott用户（oracle11g）

|  |
| --- |
| 用 system 登录进去之后，执行下面的命令：   SQL> alter user scott account unlock;   用户已更改。    这样就完成解锁的操作。接下来，你还可以重新给 scott 这个用户设定密码。    修改scott的登录密码：    SQL> alter user scott identified by tiger;   用户已更改。    大功告成，此时你再拿 scott 和 tiger 来登录就可以了！   SQL> conn scott/tiger   已连接。 |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

## Mysql

### Mysql之SQL语句优化:

### 查看数据库的编码:

|  |
| --- |
| 查看Mysql的编码：\s 要保持server characterset ，db characterset,client characterset ,conn characterset保持一致：最好四个都是utf-8的  :查看编码：mysql> show variables like 'character%'; |

### 查看数据库详细信息找到占用时间长的Show full processlist

|  |
| --- |
| 同样的查询语句之后加入\G也可以看到格式化之后的 |

### 绿色版mysql注册卸载服务方法

|  |
| --- |
| 如果直接用绿色版的mysql，则下载后解压，只需对目录下的my.ini文件的basedir（mysql的基本目录）和datadir（mysql数据目录）指定就可以，如下所示。  #Path to installation directory. All paths are usually resolved relative to this. basedir="D:\MySQL Server 5.1/" #Path to the database root datadir="D:\MySQL Server 5.1/Data/"  但是因为免安装，所以不会生成服务，只需通过简单的命令就可以实现：  打开命令窗口切换到mysql的bin目录下。  1.注册服务：  D:\MySQL Server 5.1\bin>mysqld --install mysql51  回车执行，会提示创建成功，打开服务可以找到刚创建的mysql51服务  http://files.jb51.net/file_images/article/201306/2013630152256089.png  2.卸载服务：  D:\MySQL Server 5.1\bin>mysqld --remove mysql51 |

### [mysql5.6.16绿色版配置、运行](http://blog.csdn.net/zhuxiaowei716/article/details/19405955)：包括修改root用户的密码

|  |
| --- |
| 1、从该地址http://dev.mysql.com/downloads/mysql/中选择windows的版本，选择下载。  2、将下载的压缩包解压。  3、将根目录下的my-default.ini复制重命名为my.ini。  4、打开my.ini文件，将下面的源码添加修改到my.ini中  [client] default-character-set=utf8  [mysqld]  # 设置mysql服务器编码 character\_set\_server=utf8  #basedir设置mysql的安装目录，如果目录中带空格则必须加引号  #datadir设置mysql数据库的数据的存放目录，文件夹名必须是data basedir=D:\Tool\mysql-5.6.16-win32     datadir=D:\Tool\mysql-5.6.16-win32\data  完成后保存。  5、安装mysql服务 从MS-DOS窗口进入目录E:\myserver\mysql-5.0.37-win32\bin，运行如下命令： mysqld --install mysql --defaults-file= “D:\Tool\mysql-5.6.16-win32\my.ini”  当显示Service successfully installed.时即表示mysql注册成功。  6、启动mysql数据库服务 还在上面的命令窗口里面，输入命令：net start mysql 这样就启动了mysql服务。  7、停止mysql数据库服务  还在上面的命令窗口里面，输入命令：net stop mysql 这样就停止了mysql服务。  8、删除mysql数据库服务  还在上面的命令窗口里面，输入命令：mysqld remove mysql 这样就删除了mysql服务。  9、修改mysql数据库的root密码（初始为空）  方法1： 用SET PASSWORD命令  　　mysql -u root  　　mysql> SET PASSWORD FOR ['root'@'localhost'](mailto:'root'@'localhost') = PASSWORD('newpass');  方法2：用mysqladmin  　　mysqladmin -u root password "newpass"  　　如果root已经设置过密码，采用如下方法  　　mysqladmin -u root password oldpass "newpass"  方法3： 用UPDATE直接编辑user表  　　mysql -u root  　　mysql> use mysql;  　　mysql> UPDATE user SET Password = PASSWORD('newpass') WHERE user = 'root';  　　mysql> FLUSH PRIVILEGES;  在丢失root密码的时候，可以这样  　　mysqld\_safe --skip-grant-tables&  　　mysql -u root mysql  　　mysql> UPDATE user SET password=PASSWORD("new password") WHERE user='root';  　　mysql> FLUSH PRIVILEGES;  10、设置root权限  GRANT ALL PRIVILEGES ON \*.\* TO ['root'@'%'](mailto:'root'@'%25') IDENTIFIED BY 'yourpassword'; |

### MySQL创建用户与授权方法

|  |
| --- |
| 一, 创建用户:  命令:CREATE USER 'username'@'host' IDENTIFIED BY 'password';  说明:username - 你将创建的用户名, host - 指定该用户在哪个主机上可以登陆,如果是本地用户可用localhost, 如果想让该用户可以从任意远程主机登陆,可以使用通配符%. password - 该用户的登陆密码,密码可以为空,如果为空则该用户可以不需要密码登陆服务器.  例子: CREATE USER 'dog'@'localhost' IDENTIFIED BY '123456';  CREATE USER 'pig'@'192.168.1.101\_' IDENDIFIED BY '123456';  CREATE USER 'pig'@'%' IDENTIFIED BY '123456';  CREATE USER 'pig'@'%' IDENTIFIED BY '';  CREATE USER 'pig'@'%';  二,授权:  命令:GRANT privileges ON databasename.tablename TO 'username'@'host'  说明: privileges - 用户的操作权限,如SELECT , INSERT , UPDATE 等(详细列表见该文最后面).如果要授予所的权限则使用ALL.;databasename - 数据库名,tablename-表名,如果要授予该用户对所有数据库和表的相应操作权限则可用\*表示, 如\*.\*.  例子: GRANT SELECT, INSERT ON test.user TO 'pig'@'%';  GRANT ALL ON \*.\* TO 'pig'@'%';  注意:用以上命令授权的用户不能给其它用户授权,如果想让该用户可以授权,用以下命令:  GRANT privileges ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION;  三.设置与更改用户密码  命令:SET PASSWORD FOR 'username'@'host' = PASSWORD('newpassword');如果是当前登陆用户用SET PASSWORD = PASSWORD("newpassword");  例子: SET PASSWORD FOR 'pig'@'%' = PASSWORD("123456");  四.撤销用户权限  命令: REVOKE privilege ON databasename.tablename FROM 'username'@'host';  说明: privilege, databasename, tablename - 同授权部分.  例子: REVOKE SELECT ON \*.\* FROM 'pig'@'%';  注意: 假如你在给用户'pig'@'%'授权的时候是这样的(或类似的):GRANT SELECT ON test.user TO 'pig'@'%', 则在使用REVOKE SELECT ON \*.\* FROM 'pig'@'%';命令并不能撤销该用户对test数据库中user表的SELECT 操作.相反,如果授权使用的是GRANT SELECT ON \*.\* TO 'pig'@'%';则REVOKE SELECT ON test.user FROM 'pig'@'%';命令也不能撤销该用户对test数据库中user表的Select 权限.  具体信息可以用命令SHOW GRANTS FOR 'pig'@'%'; 查看.  五.删除用户  命令: DROP USER 'username'@'host';  附表:在MySQL中的操作权限  ALTER Allows use of ALTER TABLE.  ALTER ROUTINE Alters or drops stored routines.  CREATE Allows use of CREATE TABLE.  CREATE ROUTINE Creates stored routines.  CREATE TEMPORARY TABLE Allows use of CREATE TEMPORARY TABLE.  CREATE USER Allows use of CREATE USER, DROP USER, RENAME USER, and REVOKE ALL PRIVILEGES.  CREATE VIEW Allows use of CREATE VIEW.  DELETE Allows use of DELETE.  DROP Allows use of DROP TABLE.  EXECUTE Allows the user to run stored routines.  FILE Allows use of SELECT... INTO OUTFILE and LOAD DATA INFILE.  INDEX Allows use of CREATE INDEX and DROP INDEX.  INSERT Allows use of INSERT.  LOCK TABLES Allows use of LOCK TABLES on tables for which the user also has SELECT privileges.  PROCESS Allows use of SHOW FULL PROCESSLIST.  RELOAD Allows use of FLUSH.  REPLICATION Allows the user to ask where slave or master  CLIENT servers are.  REPLICATION SLAVE Needed for replication slaves.  SELECT Allows use of SELECT.  SHOW DATABASES Allows use of SHOW DATABASES.  SHOW VIEW Allows use of SHOW CREATE VIEW.  SHUTDOWN Allows use of mysqladmin shutdown.  SUPER Allows use of CHANGE MASTER, KILL, PURGE MASTER LOGS, and SET GLOBAL SQL statements. Allows mysqladmin debug command. Allows one extra connection to be made if maximum connections are reached.  UPDATE Allows use of UPDATE.  USAGE Allows connection without any specific privileges. |

### Mysql忘记root密码的解决办法：

|  |
| --- |
| 需要知道的是mysql的root密码是存放在mysql数据库的user的表下的：  Update mysql.user set password=PASSWORD(“liting”) where user=”root”;修改mysql的root的密码  绿色版的mysql默认是没有密码的可以使用update语句进行修改：（补充说明绿色版的mysql怎么变成安装版的只要我们把mysql下的bin目录加入到了我们windows环境变量的path中那么就可以使用mysqld –install mysql（最后一个参数不是固定的是我们要显示的服务名：到时见我们就可以在cmd下使用net start 服务名来开启mysql的服务了）  但是当我们忘记了我们修改的root的密码：  在mysql的目录下有个my.ini文件：  在my.ini文件中找到mysqld的配置在下面加入：  skip-grant-tables  在我们重新登录（net stop mysql 然后net start mysql的时候就可使用mysql –uroot这个时候没有密码也可以登录了  然后我们就可以使用update mysql.user set password=PASSWORD(“”) where user =”root”;  然后使用  Flush privileges;来刷新权限这个时候就可以使用我们新设置的mysql的root的密码来重新进行登录了 |

### Mysql目录下my.ini文件中的【mysqld】下的重要配置

|  |
| --- |
| 1.远程连接mysql数据库变慢：  [mysqld]  skip-name-resolve   1. 使用show variablesb 命令查看mysql所有的系统变量：   例如可以使用   * 1. show variables like ‘port’：查看mysql的端口   2. show variables like "%char%":查看mysql的默认编码  1. 修改mysql的编码：   MySql的配置文件Windows下一般在系统目录下或者在MySql的安装目录下名字叫my.ini，可以搜索，Linux下一般是 /etc/my.cnf  --在 [mysqld] 标签下加上三行  default-character-set = utf8  character\_set\_server = utf8  lower\_case\_table\_names = 1 //表名不区分大小写（此与编码无关）  --在 [mysql] 标签下加上一行  default-character-set = utf8  --在 [mysql.server]标签下加上一行  default-character-set = utf8  --在 [mysqld\_safe]标签下加上一行  default-character-set = utf8  --在 [client]标签下加上一行  default-character-set = utf8 |

### Mysql的优化my.ini文件说明：

|  |
| --- |
| [mysqld]  port = 3306  serverid = 1  socket = /tmp/mysql.sock  skip-name-resolve  #禁止MySQL对外部连接进行DNS解析，使用这一选项可以消除MySQL进行DNS解析的时间。但需要注意，如果开启该选项，则所有远程主机连接授权都要使用IP地址方式，否则MySQL将无法正常处理连接请求！注：如果用winform连接mysql，加入此句速度会有很大的提升  skip-locking  # 避免MySQL的外部锁定，减少出错几率增强稳定性。  back\_log = 384  指定MySQL可能的连接数量。当MySQL主线程在很短的时间内接收到非常多的连接请求，该参数生效，主线程花费很短的时间检查连接并且启动一个新线程。 back\_log参数的值指出在MySQL暂时停止响应新请求之前的短时间内多少个请求可以被存在堆栈中。 如果系统在一个短时间内有很多连接，则需要增大该参数的值，该参数值指定到来的TCP/IP连接的侦听队列的大小。不同的操作系统在这个队列大小上有它自己的限制。 试图设定back\_log高于你的操作系统的限制将是无效的。默认值为50。对于Linux系统推荐设置为小于512的整数。  key\_buffer\_size = 32M  # key\_buffer\_size这对MyISAM表来说非常重要。如果只是使用MyISAM表，可以把它设置为可用内存的 30-40%。合理的值取决于索引大小、数据量以及负载 — 记住，MyISAM表会使用操作系统的缓存来缓存数据，因此需要留出部分内存给它们，很多情况下数据比索引大多了。尽管如此，需要总是检查是否所有的 key\_buffer 都被利用了 — .MYI 文件只有 1GB，而 key\_buffer 却设置为 4GB 的情况是非常少的。这么做太浪费了。如果你很少使用MyISAM表，那么也保留低于 16-32MB 的key\_buffer\_size 以适应给予磁盘的临时表索引所需。  innodb\_buffer\_pool\_size = 2.4G  #这对Innodb表来说非常重要。Innodb相比MyISAM表对缓冲更为敏感。MyISAM可以在默认的 key\_buffer\_size 设置下运行的可以，然而Innodb在默认的innodb\_buffer\_pool\_size 设置下却跟蜗牛似的。由于Innodb把数据和索引都缓存起来，无需留给操作系统太多的内存，因此如果只需要用Innodb的话则可以设置它高达 70-80% 的可用内存。– 如果你的数据量不大，并且不会暴增，那么无需把innodb\_buffer\_pool\_size 设置的太大了。  innodb\_additional\_pool\_size = 20M  #这个选项对性能影响并不太多，至少在有差不多足够内存可分配的操作系统上是这样。不过如果你仍然想设置为 20MB(或者更大)，因此就需要看一下Innodb其他需要分配的内存有多少。  innodb\_log\_file\_size = 512M  #在高写入负载尤其是大数据集的情况下很重要。这个值越大则性能相对越高，但是要注意到可能会增加恢复时间。我经常设置为64-512MB，根据服务器大小而异。  innodb\_log\_buffer\_size =16M  #默认的设置在中等强度写入负载以及较短事务的情况下，服务器性能还可以。如果存在更新操作峰值或者负载较大，就应该考虑加大它的值了。如果它的值设置太高了，可能会浪费内存 — 它每秒都会刷新一次，因此无需设置超过1秒所需的内存空间。通常8-16MB就足够了。越小的系统它的值越小。  innodb\_flush\_logs\_at\_trx\_commit = 2  #是否为Innodb比MyISAM慢1000倍而头大?看来也许你忘了修改这个参数了。默认值是 1，这意味着每次提交的更新事务(或者每个事务之外的语句)都会刷新到磁盘中，而这相当耗费资源，尤其是没有电池备用缓存时。很多应用程序，尤其是从 MyISAM转变过来的那些，把它的值设置为 2 就可以了，也就是不把日志刷新到磁盘上，而只刷新到操作系统的缓存上。日志仍然会每秒刷新到磁盘中去，因此通常不会丢失每秒1-2次更新的消耗。如果设置为0就快很多了，不过也相对不安全了 — MySQL服务器崩溃时就会丢失一些事务。设置为2指挥丢失刷新到操作系统缓存的那部分事务。  max\_allowed\_packet = 4M  thread\_stack = 256K  table\_cache = 128K  sort\_buffer\_size = 6M  #查询排序时所能使用的缓冲区大小。注意：该参数对应的分配内存是每连接独占！如果有100个连接，那么实际分配的总共排序缓冲区大小为100 × 6 ＝ 600MB。所以，对于内存在4GB左右的服务器推荐设置为6-8M。  read\_buffer\_size = 4M  #读查询操作所能使用的缓冲区大小。和sort\_buffer\_size一样，该参数对应的分配内存也是每连接独享！  join\_buffer\_size = 8M  #联合查询操作所能使用的缓冲区大小，和sort\_buffer\_size一样，该参数对应的分配内存也是每连接独享！  myisam\_sort\_buffer\_size = 64M  table\_cache = 512  #打开一个表的开销可能很大。例如MyISAM把MYI文件头标志该表正在使用中。你肯定不希望这种操作太频繁，所以通常要加大缓存数量，使得足以最大限度地缓存打开的表。它需要用到操作系统的资源以及内存，对当前的硬件配置来说当然不是什么问题了。如果你有200多个表的话，那么设置为 1024 也许比较合适(每个线程都需要打开表)，如果连接数比较大那么就加大它的值。我曾经见过设置为100,000的情况。  thread\_cache\_size = 64  #线程的创建和销毁的开销可能很大，因为每个线程的连接/断开都需要。我通常至少设置为 16。如果应用程序中有大量的跳跃并发连接并且 Threads\_Created 的值也比较大，那么我就会加大它的值。它的目的是在通常的操作中无需创建新线程。  query\_cache\_size = 64M  #指定MySQL查询缓冲区的大小。可以通过在MySQL控制台执行以下命令观察：  # > SHOW VARIABLES LIKE ‘%query\_cache%’;  # > SHOW STATUS LIKE ‘Qcache%’;  # 如果Qcache\_lowmem\_prunes的值非常大，则表明经常出现缓冲不够的情况；如果Qcache\_hits的值非常大，则表明查询缓冲使用非常频繁，如果该值较小反而会影响效率，那么可以考虑不用查询缓冲；Qcache\_free\_blocks，如果该值非常大，则表明缓冲区中碎片很多。  tmp\_table\_size = 256M  max\_connections = 768  #指定MySQL允许的最大连接进程数。如果在访问论坛时经常出现Too Many Connections的错误提 示，则需要增大该参数值。  max\_connect\_errors = 10000000  wait\_timeout = 10  #指定一个请求的最大连接时间，对于4GB左右内存的服务器可以设置为5-10。  thread\_concurrency = 8  #该参数取值为服务器逻辑CPU数量×2，在本例中，服务器有2颗物理CPU，而每颗物理CPU又支持H.T超线程，所以实际取值为4 × 2 ＝ 8  skip-networking  #开启该选项可以彻底关闭MySQL的TCP/IP连接方式，如果WEB服务器是以远程连接的方式访问MySQL数据库服务器则不要开启该选项！否则将无法正常连接！  show status 命令  含义如下:  aborted\_clients 客户端非法中断连接次数  aborted\_connects 连接mysql失败次数  com\_xxx xxx命令执行次数,有很多条  connections 连接mysql的数量  Created\_tmp\_disk\_tables 在磁盘上创建的临时表  Created\_tmp\_tables 在内存里创建的临时表  Created\_tmp\_files 临时文件数  Key\_read\_requests The number of requests to read a key block from the cache  Key\_reads The number of physical reads of a key block from disk  Max\_used\_connections 同时使用的连接数  Open\_tables 开放的表  Open\_files 开放的文件  Opened\_tables 打开的表  Questions 提交到server的查询数  Sort\_merge\_passes 如果这个值很大,应该增加my.cnf中的sort\_buffer值  Uptime 服务器已经工作的秒数  提升性能的建议:  1.如果opened\_tables太大,应该把my.cnf中的table\_cache变大  2.如果Key\_reads太大,则应该把my.cnf中key\_buffer\_size变大.可以用Key\_reads/Key\_read\_requests计算出cache失败率  3.如果Handler\_read\_rnd太大,则你写的SQL语句里很多查询都是要扫描整个表,而没有发挥索引的键的作用  4.如果Threads\_created太大,就要增加my.cnf中thread\_cache\_size的值.可以用Threads\_created/Connections计算cache命中率  5.如果Created\_tmp\_disk\_tables太大,就要增加my.cnf中tmp\_table\_size的值,用基于内存的临时表代替基于磁盘的 |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

## SQLSERVE问题总结：

### 总结JDBC连接SQLServer的错误Error establishing socket

|  |
| --- |
| 在用MSSQL连接数据库的过程中，经常会遇见“Error establishing socket.”的错误，大概得错误日志信息如下：  Sql代码   1. java.sql.SQLException : [Microsoft][SQLServer 2000 Driver **for** JDBC]Error establishing socket. 2. **at** com.microsoft.jdbc.base.BaseExceptions.createException(Unknown Source) 3. **at** com.microsoft.jdbc.base.BaseExceptions.getException(Unknown Source) 4. **at** com.microsoft.jdbc.base.BaseExceptions.getException(Unknown Source) 5. **at** com.microsoft.jdbc.sqlserver.tds.TDSConnection.<init>(Unknown Source) 6. **at** com.microsoft.jdbc.sqlserver.SQLServerImplConnection.**open**(Unknown Source) 7. **at** com.microsoft.jdbc.base.BaseConnection.getNewImplConnection(Unknown Source) 8. **at** com.microsoft.jdbc.base.BaseConnection.**open**(Unknown Source) 9. **at** com.microsoft.jdbc.base.BaseDriver.**connect**(Unknown Source) 10. **at** java.sql.DriverManager.getConnection( DriverManager.java:523 ) 11. **at** java.sql.DriverManager.getConnection( DriverManager.java:171 ) 12. **at** test.SqlServerTest.main( SqlServerTest.java:29 )   一　配置错误  jdbc配置语句为: jdbc:microsoft:sqlserver://server\_name:1433  其中的sqlserver、server\_name和端口号都要看仔细了。  二　1433端口问题  可以通过命令“netstat -an”，来查看是否开了１４３３端口，如果列表中没有，要增开端口。  要保持jdbc配置语句（ jdbc:microsoft:sqlserver://server\_name:1433）中的端口与数据库传输协议默认的端口相同。  MSSQL设置端口号步骤如下：  1． 打开企业管理器，依次在控制台根目录 ->Microsoft SQL Servers->SQL Server 组，列出一部分数据库服务器。  2． 右击我们要连接的数据库服务器，选择属性，在常规选项卡点击网络配置弹出新窗口。  3． 在启用的协议区域我们选择 TCP/IP ，点击属性按钮，弹出新窗口。在这个窗口有网络协议默认值设置，一般默认端口是：1433。 如果你在ｕｒｌ中用的端口号与这个不同，就会出现错误。应保持二者相同。  三 没打补丁  可以用如下的方式查看MSSQL的版本：  打开企业管理器-〉工具-〉SQL查询分析器-〉帮助-〉关于  查看MSSQL的详细版本号  如果 是 8.00.194 就是还没打补丁  8.00.760 就是SP3  8.00.2039 就是SP4   1、下载MSSQL　2000 Service Pack 3a并安装，SQL请选用混和安装模式！！！  http://www.microsoft.com/downloads/details.aspx?FamilyId=90DCD52C-0488-4E46-AFBF-ACACE5369FA3&displaylang=zh-cn  解压缩sp3的升级包后,还要点击setup安装.才能成功升级到sp3。  2、下载SQL Server 2000 Driver for JDBC Service Pack 3  http://www.microsoft.com/downloads/details.aspx?FamilyId=07287B11-0502-461A-B138-2AA54BFDC03A&displaylang=en  四　系统防火墙或者杀毒软件  如果系统开防火墙或者杀毒软件，在有些情况下也可以出现类似的问题。  没有遇见过，有人这么说。 |

### 本地计算机上的SQL Server(MSSQLSERVER)服务启动后停止。某些服务在未由其他服务或程序使用时将自动停止

|  |
| --- |
| 问题描述：  SQL Server 2005能用，能连外面的数据库，本地的进不去，是因为本地的  SQL Server(MSSQLSERVER)服务无法启用，启用服务就报错：  本地计算机上的SQL Server(MSSQLSERVER)服务启动后停止。某些服务在未由其他服务或程序使用时将自动停止 |
| 这个问题的确是需要禁止VIA协议就可以了 |

### SQL Server无法启动,错误代码 3417，提示Windows不能在本地计算机启动，Sql server错误框提示为：“Windows不能在 本地计算机 启动 SQL Server(MSSQLSERVER)。有关更多信息，查阅系统事件日志。如果这是非Microsoft 服务，请与服务厂商联系，并参考特定服务错误代码3417

|  |
| --- |
| 解决办法一：  C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data       C:\Program Files\Microsoft SQL Server\MSSQL.2\MSSQL\Data  下的所有mdf和ldf默认都是按照解压缩方式保存的，只需要右键每一个文件---属性---高级---压缩内容以便节省磁盘空间 的勾   去掉即可。  解决办法二：  第一个解决不了就用这个了。  打开 控制面板 -> 管理工具 -> 服务  找到SQL Server 服务，右键，属性，找到登录选项，选择本地账户，确定就可以启动了。  下面是我的办法：找到另一个可以运行的SQLSERVER要是同一个版本的~~然后进入到  拷贝到相应的目录就行了 ~~ |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

## SQL

### SQL查询null和””的用法:

|  |
| --- |
| 空 如果是null 就是select name,age,成绩 from AA where 成绩 is  null 如果是"" 就是select name,age,成绩 from AA where 成绩=""  不空 如果是null 就是select name,age,成绩 from AA where 成绩 is not null 如果是"" 就是select name,age,成绩 from AA where 成绩<>"" |

### SQL DATEDIFF语法及时间函数 Sql 查询当天、本周、本月记录

|  |
| --- |
| 1. --查询当天： 2. **select** \* **from** info **where** DateDiff(dd,datetime,getdate())=0 4. --查询24小时内的: 5. **select** \* **from** info **where** DateDiff(hh,datetime,getDate())<=24 7. --info为表名,datetime为数据库中的字段值   --查询当天：select \* from info where DateDiff(dd,datetime,getdate())=0--查询24小时内的:select \* from info where DateDiff(hh,datetime,getDate())<=24--info为表名,datetime为数据库中的字段值  Sql代码 Powered by chenjiazi   1. --查询当天记录另类的方法 2. **SELECT** \* 3. **FROM** j\_GradeShop 4. **WHERE** (GAddTime BETWEEN CONVERT(datetime, LEFT(GETDATE(), 10) + ' 00:00:00.000') 5. AND CONVERT(datetime, LEFT(GETDATE(), 10) + ' 00:00:00.000') + 1) 6. **ORDER** **BY** GAddTime **DESC**   --查询当天记录另类的方法SELECT \*FROM j\_GradeShopWHERE (GAddTime BETWEEN CONVERT(datetime, LEFT(GETDATE(), 10) + ' 00:00:00.000') AND CONVERT(datetime, LEFT(GETDATE(), 10) + ' 00:00:00.000') + 1)ORDER BY GAddTime DESC  DATEDIFF 函数:  语法:  DATEDIFF ( datepart , startdate , enddate ) 备注:  enddate 减去 startdate。如果 startdate 晚于 enddate，则返回负值。  如果结果超出整数值范围，则 DATEDIFF 将产生错误。对于毫秒，最大数是 24 天 20 小时 31 分钟零 23.647 秒。对于秒，最大数是 68 年。  跨分钟、秒和毫秒等边界计算的方法使得 DATEDIFF 指定的结果在所有数据类型中均一致。结果是带正负号的整数值，它等于跨第一个和第二个日期间的 datepart 边界数。例如，在 1 月 4 日（星期日）和 1 月 11 日（星期日）之间的星期数是 1。  可以再MSSQL中测试:  Sql代码   1. --两个时间差刚好是24 2. --打印的方式 3. print dateDiff(hh,'2009-1-1 0:0:0','2009-1-2 0:0:0') 5. --查询的方式 6. print dateDiff(hh,'2009-1-1 0:0:0','2009-1-2 0:0:0')   --两个时间差刚好是24--打印的方式print dateDiff(hh,'2009-1-1 0:0:0','2009-1-2 0:0:0')--查询的方式print dateDiff(hh,'2009-1-1 0:0:0','2009-1-2 0:0:0')  Sql代码   1. --本月记录 2. **SELECT** \* **FROM** 表 **WHERE** datediff(month,[dateadd],getdate())=0 4. --本周记录 5. **SELECT** \* **FROM** 表 **WHERE** datediff(week,[dateadd],getdate())=0 7. --包括本年这些查询方式是一样的   --本月记录 SELECT \* FROM 表 WHERE datediff(month,[dateadd],getdate())=0 --本周记录 SELECT \* FROM 表 WHERE datediff(week,[dateadd],getdate())=0--包括本年这些查询方式是一样的  **sql server中的时间函数**    1.   当前系统日期、时间       select getdate()  2. dateadd   在向指定日期加上一段时间的基础上，返回新的 datetime 值     例如：向日期加上2天      select dateadd(day,2,'2004-10-15')   --返回：2004-10-17 00:00:00.000  3. datediff 返回跨两个指定日期的日期和时间边界数。     select datediff(day,'2004-09-01','2004-09-18')    --返回：17  4. datepart 返回代表指定日期的指定日期部分的整数。    SELECT DATEPART(month, '2004-10-15')   --返回 10  5. datename 返回代表指定日期的指定日期部分的字符串     SELECT datename(weekday, '2004-10-15')   --返回：星期五  6. day(), month(),year() --可以与datepart对照一下  select 当前日期=convert(varchar(10),getdate(),120)  ,当前时间=convert(varchar(8),getdate(),114)  select datename(dw,'2004-10-15')  select 本年第多少周=datename(week,'2004-10-15')        ,今天是周几=datename(weekday,'2004-10-15') |

### 存储过程（Stored Procedure）

|  |
| --- |
| 存储过程就是一个已经编译和优化过的SQL语句应用程序可以直接进行调用：  使用存储过程的优点：   1. 执行速度比普通的SQL快很多（大部分执行SQL的语句存储过程已经完成了所以比普通的SQL执行的快的多） 2. 便于集中控制（修改的时候只需要修改存储过程就行了：调用的语句不用修改） 3. 保证数据的完整新和安全性 4. 灵活性：（可以控制流程语句的编写——可以完成复杂的判断和运算可以执行不同的sql |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# Eclipse:使用总结：

## 修改窗口的背景色

|  |
| --- |
|  |

# Svn：

## 解决subclipse同步冲突问题A conflict in the working copy obstructs the current operation

|  |
| --- |
| 服务端版本控制软件subversion，客户端是eclipse的插件subclipse，个人感觉还是cvs比较简单方便。明明没什么冲突，当删除一个东西的时候老是提示错误，说冲突  A conflict in the working copy obstructs the current operation org.tigris.subversion.javahl.ClientException: A conflict in the working copy obstructs the current operation svn: Commit failed (details follow): svn: Aborting commit: 'D:eclipse-java-ganymede-SR1-win32workspacemyProjectdatabasesrc' remains in conflict  org.tigris.subversion.javahl.ClientException: A conflict in the working copy obstructs the current operation svn: Commit failed (details follow): svn: Aborting commit: 'D:eclipse-java-ganymede-SR1-win32workspacemyProjectdatabasesrc' remains in conflict  解决办法  右击你要删除的目录-->team-->Show Tree Conflict-->在Eclpise中会出现一个Tab视图显示Tree Conflict的内容，一个一个选择列表中的冲突右击将其标志为已解决，然后回到项目再提交更改就可以了。  注意：我用的版本subclipse是<http://subclipse.tigris.org/update_1.6.x>  这样的冲突会照成无法更新~~~更新的时候会跳过这个冲突的文件~~~提交的时候提交不了~~这个时候应该还原一下 就可以解决了 ~~所谓的还原就是还原服务器上的上一个版本~~~ |

|  |
| --- |
|  |

|  |
| --- |
|  |

# 办公技巧

## word

### Word自动生成页码:

|  |
| --- |
| 1. 在开始的时候定好样式: 3. 编写文档: 5. 这样目录就生产了(当编辑完成之后可以选择更新目录) |

|  |
| --- |
|  |

## Excel

|  |
| --- |
|  |

|  |
| --- |
|  |

## Powerpoint

|  |
| --- |
|  |

|  |
| --- |
|  |

# SSH

## Struts

### struts2配置文件加载顺序

|  |
| --- |
| 从先到后的顺序如下：  struts-default.xml  struts-plugin.xml  struts.xml  struts.properties  web.xml  建议，要覆盖默认配置，都在struts.xml中去覆盖；  不建议使用struts.properties； |

|  |
| --- |
|  |

## spring

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

## hibernate

id to load is required for loading

|  |
| --- |
| 错误原因应该是你的hibernate的配置文件把有些字段设置为not null但是你实际的对象中存在null值，所以就会出现这个错误 |

# android

|  |
| --- |
| 在建立android application project项目时出现 The tools need to be updated via the SDK Manager  http://a.hiphotos.baidu.com/zhidao/wh%3D600%2C800/sign=87a2720c85d6277fe9473a3e18083308/7c1ed21b0ef41bd5adcd0e4353da81cb38db3dee.jpg  解决办法：  比较保险的防范用SDK Manager来更新  更新红线部分就可以了!  更新到最新的SDK Tools后还要更新下SDK Build-tools |

## 对Android Target SDK的理解

|  |
| --- |
| 对Android Target SDK的理解，有需要的朋友可以参考下。  <uses-sdk android:minSdkVersion="integer" android:targetSdkVersion="integer" android:maxSdkVersion="integer" />    targetSdkVersion告诉OS，我就是在这个版本就行测试的，不需要任何兼容性行为；影响app运行。   在project.properties 文件中target为buildtarget是表明用这个版本的SDK编译；影响的是app的编译。  一般情况下建议targetSdkVersion＝buildtarget，而且始终为最新release的版本。  案例分析：  minSdkVersion＝10，则在android2.0上不能安装，因为API level为8；  targetSdkVersion＝13，  buildtarget＝14，则ScrollView才支持canScrollVertically接口；如果buildtarget＝8，因为在这个版本的sdk里面没有这个api |

## 解决eclipse下android项目不能自动生成R.java文件

|  |
| --- |
| 通常情况下R.java文件是项目建立时自动生成的。但是，大家在eclipse下做开发的时候偶尔会遇到R.java文件不能自动生成。现在就来解决这个问题。     1.eclipse ADT插件的版本与android SDK版本不对应     一般问题就出现在这儿，这样的情况只要把ADT插件和SDK都更新到最新，重启eclipse一切就OK了。       红框那儿要全部install到最新。       2.eclipse自己出问题（和你闹别扭，哈哈）     这样的你就clean一下你的项目，或者使用android工具fix下。R.java文件就又回来了。 |

## tools:context=".MainActivity的作用

|  |
| --- |
| <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerHorizontal="true"  android:layout\_centerVertical="true"  android:text="@string/hello\_world"  tools:context=".MainActivity" /> |
| tools:context="activity name"这一句不会被打包进APK。只是ADT的Layout Editor在你当前的Layout文件里面设置对应的渲染上下文，说明你当前的Layout所在的渲染上下文是activity name对应的那个activity，如果这个activity在manifest文件中设置了Theme，那么ADT的Layout Editor会根据这个Theme来渲染你当前的Layout。就是说如果你设置的MainActivity设置了一个Theme.Light（其他的也可以），那么你在可视化布局管理器里面看到的背景阿控件阿什么的就应该是Theme.Light的样子。仅用于给你看所见即所得的效果而已。 |

## android:label标签在application和activity中的设置问题

|  |
| --- |
| AndroidMainfest.xml中在application和activity中都可以设置android:lable。  android:label用来设置应用名和标题名。  应用名：  当主activity和application中都设置此值后，应用名会优先使用主activity中的值。  标题名：  当application和activity中都设置android:label时，标题名会优先使用各个activity中的值。当存在activity中没有设置值时，会使用application中的值。 |

## 搬家、备份后启动Android PANIC :Could not open D:\java2\android\android-sdk-windows\.android\avdtest.ini问题的解决

|  |
| --- |
| PANIC： Could notopen Djava2androidandroid-sdk-windows.androidavdtest.ini的错误字样如图：http://img.my.csdn.net/uploads/201209/13/1347541533_9896.png  、http://img.my.csdn.net/uploads/201209/13/1347541624_5586.pnghttp://img.my.csdn.net/uploads/201209/13/1347541656_3532.pnghttp://img.my.csdn.net/uploads/201209/13/1347541677_2339.png  http://img.my.csdn.net/uploads/201209/13/1347541700_4438.png  看到这种状况后，我很快便意识到，这是我以前遇到过的一个问题，以前在解决的时候就费了好大的劲，我首先是上网查找是否有和我出现过同样问题的朋友，寻找解决途径，终于，我知道了出现这个问题的原因：以前我使用过备份工具将C盘下的系统文件备份到了“D:\  个人重要资料(勿删)\朋友一生”的文件夹下，这样做以后，原来安装软件时默认写到C盘的“C:\Documents andSettings\Administrator”的文件就写到了“D:\个人重要资料(勿删)\朋友一生”的文件夹下，对于Android是不能打开有着中文路径的的这种文件的，即在运行Android虚拟机时，由于不能打开“D:\个人重要资料(勿删)\朋友一生\.android\avd ”文件，所以就出现了上述问题。  http://img.my.csdn.net/uploads/201209/13/1347541758_5370.png  当时，我也是通过上网查才知道要去配置环境变量，即：       对着“我的电脑”图标单击鼠标右键，在弹出的菜单中选择“属性”，切换到“高级”选项卡，单击“环境变量”按钮，在弹出的对话框新建一个环境变量，比如名为ANDROID\_SDK\_HOME (不可修改为其他) ，值为D:\AndroidE\android\android-sdk-windows   (假设sdk我们在d盘的目录名为android-sdk-windows)下。  http://img.my.csdn.net/uploads/201209/13/1347541791_4207.png  http://img.my.csdn.net/uploads/201209/13/1347541813_7852.png  接下来重新启动Eclipse，我们通过单击Windows菜单，依次进入 preferneces => Android => Bulid 可以看到 Default debug keystore: 其中.android\为ADT自动创建的，  以前这项可能为 D:\个人重要资料(勿删)\朋友一生\.android\debug.keystore)      http://img.my.csdn.net/uploads/201209/13/1347541883_9249.png  http://img.my.csdn.net/uploads/201209/13/1347541921_6841.png  另外，也可以通过设置下面的custom debug keystore来设置可以免除添加环境变量的步骤。  然后重建avd。运行程序，应该就没有问题了。    当然，如果这样做还不行的话，你可以这样做试试：  在配置好后即：已经设置了环境变量：ANDROID\_[SDK](http://www.daozhao.com/daozhao/tag/sdk)\_HOME值为你的SDK目录。如我的是：变量名 ANDROID\_[SDK](http://www.daozhao.com/daozhao/tag/sdk)\_HOME 值为D:\AndroidE\android\android-sdk-windows  2、在eclipse安装目录修改配置文件：找到以下文件:eclipseconfiguration.settingsorg.eclipse.ui.ide.prefs打开后，在后面补充改刚刚配置的环境变量。 如我补充的是：ANDROID\_SDK\_HOME=D\:\\AndroidE\\android\android-sdk-windows（注意斜杠格式）  这样  http://img.my.csdn.net/uploads/201209/13/1347542000_1613.pnghttp://img.my.csdn.net/uploads/201209/13/1347542029_3819.png  这样就应该行了，网上还有这样说的  查看你的AVD目前所在的目录    　　1. 把整个目录复制到C:\Documents and Settings\Administrator\ 目录下        　　2. 修改 C:\Documents and Settings\Administrator\.android/avd/zte345.ini    　　原来的内容如下：    　　target=android-8path=D:\我的文档\.android\avd\zte345.avd        　　将里面包含的路径修改为    　　target=android-8path=C:\Documents and Settings\Administrator\.android\avd\zteluck.avd  或是说拷贝.android文件夹来解决的，这些方法我也都试了一下，不过在我这里是没有起作用。    http://img.my.csdn.net/uploads/201209/13/1347542167_4675.png |

## Android 设置环境出现Java not found in your path的解决办法

|  |
| --- |
| 在安装android SDK时，设置好JAVA环境和android环境变量时  WARNING: Java not found in your path.  Checking if Java is installed in C:\Program Files\Java.  ERROR: No suitable Java found. In order to properly use the Android Developer  Tools, you need a suitable version of Java JDK installed on your system.  We recommend that you install the JDK version of JavaSE , available here:  http://www.oracle.com/technetwork/java/javase/downloads  You can find the complete Android SDK requirements here:  http://developer.android.com/sdk/requirements.html  在CMD中运行java 和javac命令时都没问题  出现这个问题的原因是因为你安装的JDK版本不对，可能你安装的是J2EE 的JDK  在上面推荐的下载网站中下载JavaSE的JDK，安装好后重新设置一下JAVA环境变量就可以正常使用的 |

## Android不能正常生成R.java文件：

|  |
| --- |
| 由于gen目录下的R.java是由eclipse工具生成的  所以看不到R.java我就头疼了开始了我的折腾之路：  由于我之前还是能用的手贱的点了 更新adt 但是由于没能正常更新adt我就想把更新的删除掉于是我按照删除插件的方法进行删除：  无奈失败了————于是我只好把原来的eclipse还原回来就好了 （这里是adt的问题：  网上对于这种情况的说明：（原理都是一样：ADT工具自动生成的）   通常情况下R.java文件是项目建立时自动生成的。但是，大家在eclipse下做开发的时候偶尔会遇到R.java文件不能自动生成。现在就来解决这个问题。     1.eclipse ADT插件的版本与android SDK版本不对应     一般问题就出现在这儿，这样的情况只要把ADT插件和SDK都更新到最新，重启eclipse一切就OK了     红框那儿要全部install到最新     2.eclipse自己出问题（和你闹别扭，哈哈）     这样的你就clean一下你的项目，或者使用android工具fix下。R.java文件就又回来了 |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# HTML5

创建带边框的表单：

|  |
| --- |
| <form action="">  <fieldset>  <legend>Personal information:</legend>  Name: <input type="text" size="30"><br>  E-mail: <input type="text" size="30"><br>  Date of birth: <input type="text" size="10">  </fieldset>  </form> |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |