## 竭诚为您提供优质文档/双击可除

## 

## jacob,操作word表格

　　篇一：jacob读取word表格  
　　packagetest;  
　　importjava.io.bufferedReader;  
　　importjava.io.FileReader;  
　　importjava.io.ioexception;  
　　importjava.util.arraylist;  
　　importjava.util.list;  
　　importbean.dx;  
　　importcom.jacob.activex.activexcomponent;  
　　importcom.jacob.com.comthread;  
　　importcom.jacob.com.dispatch;  
　　importcom.jacob.com.Variant;  
　　publicclassReadwordbystream{  
　　publicstaticvoidmain(string[]args)throwsioexception{  
　　dispatchworddoc=null;  
　　activexcomponentword=null;  
　　try{  
　　word=newactivexcomponent("word.application");  
　　word.setproperty("Visible",newVariant(false));  
　　dispatchdocuments=word.getproperty("documents").todispatch();  
　　worddoc=dispatch.call(documents,"open","d:\\word\\1.doc").todispatch();  
　　dispatchtables=dispatch.get(worddoc,"tables").todispatch();  
　　dispatchtable=dispatch.call(tables,"item",newVariant(1)).todispatch();dispatchrows=dispatch.get(table,"Rows").todispatch();  
　　dispatchcolumns=dispatch.get(table,"columns").todispatch();dispatchcell;  
　　dispatchrange;  
　　stringdata;  
　　//dxd=newdx();  
　　//listlist=newarraylist();  
　　system.out.println("一共打多少收按行="+dispatch.get(rows,"count").getint());  
　　system.out.println("一共多少列="+dispatch.get(columns,"count").getint());for(inti=1;i　　for(intj=1;j　　range=dispatch.get(cell,"Range").todispatch();  
　　data=dispatch.get(range,"text").getstring();  
　　system.out.print(data.trim()+"|");}system.out.println();}}catch(exceptione){e.printstacktrace();}finally{dispatch.call(worddoc,"close",newVariant(true));word.invoke("quit",newVariant[0]);}}}  
　　篇二：jacob操作word详细教程  
　　jacob操作word详细教程  
　　博客分类：  
　　  
　　javatomcatqq应用服务器F#  
　　首先，大家先要了解一下jacob，官方的解释是javacombridge，即java和com组件间的桥梁，这里说说为什么我们用jacob操纵word。而不直接使用java去做？  
　　这要原因：在java开源世界没有很好工具来操作word文档,poi对word操作还是很不完善,所以我们无法使用它很方便操作word文档来满足我们需求。相比之下使用jacob操作word文档非常方便。也比较容易。  
　　jacob下载地址:http://danadler.com/jacob/这个网址还可以下载到源码和例子程序  
　　jacob使用方法：将jacob1.7里面jacob.jar添加到我们应用程序环境中，并将jacob.dl（l就是我前面说的com组件）把放到c:/windows/system32下。如果是web环境中，需要将jacod.jar放到tomcat的lib目录下.（如果用tomcat服务器）  
　　值得注意的是，不同的版本的系统使用不同的dll文件  
　　所以如果你编译成功，但运行失败一般是dll文件问题  
　　遇到这种情况，可以到  
　　http://downloads.sourceforge.net/jacob-project/jacob\_1.9.zipmodtime=1109437002  
　　2.  
　　3.importjava.io.bufferedinputstream;  
　　4.importjava.io.bufferedoutputstream;  
　　5.importjava.io.File;  
　　6.importjava.io.Fileinputstream;  
　　7.importjava.io.Fileoutputstream;  
　　8.importjava.io.inputstream;  
　　9.importjava.io.outputstream;  
　　10.importjava.text.simpledateFormat;  
　　11.importjava.util.arraylist;  
　　12.importjava.util.date;  
　　13.importjava.util.hashmap;  
　　14.importjava.util.iterator;  
　　15.importjava.util.list;  
　　16.importjava.util.map;  
　　17.importjava.util.set;  
　　18.  
　　19.publicclasswordwriter{  
　　20.  
　　21.privatewordoperatorword;  
　　22.  
　　23.publicwordwriter(stringfilepath){  
　　24.word=newwordoperator();  
　　25.word.opendocument(filepath);  
　　26.}  
　　27.  
　　28.publicwordwriter(inputstreaminput,stringfilepath,stringfilename)throwsexception{  
　　29.stringpath=saveasdocFile(input,filepath,filename);  
　　30.word=newwordoperator();  
　　31.word.opendocument(path);  
　　32.}  
　　33./\*\*  
　　34.\*将word文档输入流保存为本地得到word文件  
　　35.\*@paraminput  
　　36.\*@paramfilepath  
　　37.\*@paramfilename  
　　38.\*@throwsexception  
　　39.\*/  
　　40.@suppresswarnings("unused")  
　　41.privatestringsaveasdocFile(inputstreaminput,stringfilepath,stringfilename)throwsexception{  
　　42.if(!stringutils.isValidatestring(filepath)||!stringutils.isValidatestring(filename)){  
　　43.thrownewexception("thefilepathorfilenameiserror");  
　　44.}  
　　45.if(input==null){  
　　46.thrownewexception("inputstreamisnull");  
　　47.}  
　　48.Filefile=newFile(filepath);  
　　49.  
　　50.if(!file.exists()){  
　　51.thrownewexception("theFilepathisnull");  
　　52.}  
　　53.filepath=validateFilepath(filepath);  
　　54.filename=getRandomFilename(filename);  
　　55.inputstreamin=null;  
　　56.outputstreamout=null;  
　　57.try{  
　　58.in=newbufferedinputstream(input);  
　　59.out=newbufferedoutputstream(newFileoutputstream(filepath+filename));  
　　60.byte[]b=newbyte[1024];  
　　61.for(intp=0;(p=in.read(b))!=-1;){  
　　62.out.write(b);  
　　63.out.flush();  
　　64.}  
　　65.}finally{  
　　66.if(out!=null){  
　　67.out.close();  
　　68.}  
　　69.if(in!=null){  
　　70.in.close();  
　　71.}  
　　72.}  
　　73.returnfilepath+filename;  
　　74.}  
　　75./\*\*  
　　76.\*验证word文件路径  
　　77.\*@paramfilepath  
　　78.\*@return  
　　79.\*/  
　　80.privatestringvalidateFilepath(stringfilepath){  
　　81.if((filepath.lastindexof("\\\\")==-1)  
　　83.}  
　　84.returnfilepath;  
　　85.}  
　　86./\*\*  
　　87.\*生成一个新的文件名（保证文件名不相同）  
　　88.\*@paramfilename  
　　89.\*@return  
　　90.\*/  
　　91.privatestringgetRandomFilename(stringfilename){  
　　92.filename=filename+"\_"+newsimpledateFormat("yyyymmddhhmmssz").format(newdate())+".doc";  
　　93.returnfilename;  
　　94.}  
　　95./\*\*  
　　96.\*replacetext  
　　97.\*@parammap  
　　98.\*/  
　　99.publicvoidreplacealltext(mapmap){100.if(map==null){  
　　101.return;  
　　102.}  
　　103.setkeys=map.keyset();  
　　104.iteratorit=keys.iterator();  
　　105.while(it.hasnext()){  
　　106.stringkey=it.next();  
　　107.word.replacealltext(key,map.get(key));108.}  
　　109.}  
　　110./\*\*  
　　111.\*adddetails  
　　112.\*@paramvalues  
　　113.\*/  
　　114.publicvoidinsertcontextinRow(list>values,inttableindex){  
　　115.if(tableindex　　116.tableindex=1;  
　　117.}  
　　118.if(values==null||values.size()　　119.return;  
　　120.}  
　　121.int[]p=null;  
　　122.mapm=values.get(0);  
　　123.setkeys=m.keyset();  
　　124.iteratorit=keys.iterator();  
　　125.while(it.hasnext()){  
　　126.stringstr=it.next();  
　　127.  
　　128.int[]a=word.gettablecellpostion(str,tableindex);  
　　129.if(a!=null  
　　131.}  
　　132.  
　　133.}  
　　134.if(p!=nulli　　136.word.addtableRow(tableindex,p[0]);//在表格插入行数  
　　137.}  
　　138.}  
　　139.  
　　140.iteratorit2=keys.iterator();141.while(it2.hasnext()){  
　　142.introw=p[0];  
　　143.intcol=0;  
　　144.stringstr=it2.next();  
　　145.  
　　146.int[]a=word.gettablecellpostion(str,tableindex);  
　　147.if(a!=null){  
　　148.col=a[1];  
　　149.}  
　　150.for(mapmap:values){  
　　151.word.puttxttocell(tableindex,row,col,map.get(str));  
　　152.row++;  
　　153.}  
　　154.}  
　　155.  
　　156.}  
　　157.  
　　158./\*\*  
　　159.\*closedocument  
　　160.\*/  
　　161.publicvoidclose(){  
　　162.word.closedocument();  
　　163.word.close();  
　　164.}  
　　165./\*\*  
　　166.\*依据word文件完整路径删除文件  
　　167.\*@parampath  
　　168.\*@throwsexception  
　　169.\*/  
　　篇三：java通过jacob操作word  
　　packagecn.netinnet.bysj.word;  
　　importjava.io.File;  
　　importjava.sql.connection;  
　　importjava.sql.drivermanager;  
　　importjava.sql.Resultset;  
　　importjava.sql.statement;  
　　importjava.util.arraylist;  
　　importjava.util.list;  
　　importcom.jacob.activex.activexcomponent;  
　　importcom.jacob.com.dispatch;  
　　importcom.jacob.com.Variant;  
　　publicclassmswordmanager{  
　　//word文档  
　　privatedispatchdoc;  
　　//word运行程序对象  
　　privateactivexcomponentword;  
　　//所有word文档集合  
　　privatedispatchdocuments;  
　　//选定的范围或插入点  
　　privatedispatchselection;  
　　privatebooleansaveonexit=true;  
　　privatestaticfinalintwoRd\_html=8;  
　　privatestaticfinalintwoRd\_txt=(jacob,操作word表格)7;  
　　privatestaticfinalintexcel\_html=44;  
　　/\*\*\*//\*\*  
　　\*  
　　\*@paramvisible为true表示word应用程序可见  
　　\*/  
　　publicmswordmanager(booleanvisible){  
　　if(word==null){  
　　word=newactivexcomponent("word.application");word.setproperty("Visible",newVariant(visible));}  
　　if(documents==null)  
　　documents=word.getproperty("documents").todispatch();  
　　}  
　　/\*\*\*//\*\*  
　　\*设置退出时参数  
　　\*  
　　\*@paramsaveonexitbooleantrue-退出时保存文件，false-退出时不保存文件\*/  
　　publicvoidsetsaveonexit(booleansaveonexit){  
　　this.saveonexit=saveonexit;  
　　}  
　　/\*\*\*//\*\*  
　　\*创建一个新的word文档  
　　\*  
　　\*/  
　　publicvoidcreatenewdocument(){  
　　doc=dispatch.call(documents,"add").todispatch();  
　　selection=dispatch.get(word,"selection").todispatch();  
　　}  
　　/\*\*\*//\*\*  
　　\*打开一个已存在的文档  
　　\*  
　　\*@paramdocpath  
　　\*/  
　　publicvoidopendocument(stringdocpath){  
　　closedocument();  
　　doc=dispatch.call(documents,"open",docpath).todispatch();selection=dispatch.get(word,"selection").todispatch();  
　　}  
　　/\*\*\*//\*\*  
　　\*把选定的内容或插入点向上移动  
　　\*  
　　\*@parampos移动的距离  
　　\*/  
　　publicvoidmoveup(intpos){  
　　if(selection==null)  
　　selection=dispatch.get(word,"selection").todispatch();for(inti=0;i　　　　dispatch.call(selection,"moveup");  
　　}  
　　/\*\*\*//\*\*  
　　\*把选定的内容或者插入点向下移动  
　　\*  
　　\*@parampos移动的距离  
　　\*/  
　　publicvoidmovedown(intpos){  
　　if(selection==null)  
　　selection=dispatch.get(word,"selection").todispatch();for(inti=0;i　　　　dispatch.call(selection,"movedown");  
　　}  
　　/\*\*\*//\*\*  
　　\*把选定的内容或者插入点向左移动  
　　\*  
　　\*@parampos移动的距离  
　　\*/  
　　publicvoidmoveleft(intpos){  
　　if(selection==null)  
　　selection=dispatch.get(word,"selection").todispatch();for(inti=0;i　　　　dispatch.call(selection,"moveleft");  
　　}  
　　}  
　　/\*\*\*//\*\*  
　　\*把选定的内容或者插入点向右移动  
　　\*  
　　\*@parampos移动的距离  
　　\*/  
　　publicvoidmoveRight(intpos){  
　　if(selection==null)  
　　selection=dispatch.get(word,"selection").todispatch();for(inti=0;i　　　　dispatch.call(selection,"moveRight");  
　　}  
　　/\*\*\*//\*\*  
　　\*把插入点移动到文件首位置  
　　\*  
　　\*/  
　　publicvoidmovestart(){  
　　if(selection==null)  
　　selection=dispatch.get(word,"selection").todispatch();  
　　dispatch.call(selection,"homekey",newVariant(6));  
　　}  
　　publicvoidmoveend(){  
　　if(selection==null)  
　　selection=dispatch.get(word,"selection").todispatch();dispatch.call(selection,"endkey",newVariant(6));  
　　}  
　　/\*\*\*//\*\*  
　　\*从选定内容或插入点开始查找文本  
　　\*  
　　\*@paramtoFindtext要查找的文本  
　　\*@returnbooleantrue-查找到并选中该文本，false-未查找到文本\*/  
　　publicbooleanfind(stringtoFindtext){  
　　if(toFindtext==null||toFindtext.equals(""))  
　　returnfalse;  
　　//从selection所在位置开始查询  
　　dispatchfind=word.call(selection,"Find").todispatch();//设置要查找的内容  
　　dispatch.put(find,"text",toFindtext);  
　　//向前查找  
　　dispatch.put(find,"Forward","true");  
　　//设置格式  
　　dispatch.put(find,"Format","true");  
　　//大小写匹配  
　　dispatch.put(find,"matchcase","true");  
　　//全字匹配  
　　dispatch.put(find,"matchwholeword","true");  
　　//查找并选中  
　　returndispatch.call(find,"execute").getboolean();  
　　}  
　　/\*\*\*//\*\*  
　　\*把选定选定内容设定为替换文本  
　　\*  
　　\*@paramtoFindtext查找字符串  
　　\*@paramnewtext要替换的内容  
　　\*@return  
　　\*/  
　　publicbooleanreplacetext(stringtoFindtext,stringnewtext){  
　　if(!find(toFindtext))  
　　returnfalse;  
　　dispatch.put(selection,"text",newtext);  
　　returntrue;  
　　}  
　　/\*\*\*//\*\*  
　　\*全局替换文本  
　　\*  
　　\*@paramtoFindtext查找字符串  
　　\*@paramnewtext要替换的内容  
　　\*/  
　　publicvoidreplacealltext(stringtoFindtext,stringnewtext){  
　　while(find(toFindtext)){  
　　dispatch.put(selection,"text",newtext);  
　　dispatch.call(selection,"moveRight");  
　　}  
　　}  
　　/\*\*\*//\*\*  
　　\*在当前插入点插入字符串  
　　\*  
　　\*@paramnewtext要插入的新字符串  
　　\*/  
　　publicvoidinserttext(stringnewtext){  
　　dispatch.put(selection,"text",newtext);  
　　}  
　　/\*\*\*//\*\*  
　　\*  
　　\*@paramtoFindtext要查找的字符串  
　　\*@paramimagepath图片路径  
　　\*@return  
　　\*/  
　　publicbooleanreplaceimage(stringtoFindtext,stringimagepath){if(!find(toFindtext))  
　　returnfalse;  
　　dispatch.call(dispatch.get(selection,"inlineshapes").todispatch(),"addpicture",imagepath);  
　　returntrue;  
　　}  
　　/\*\*\*//\*\*  
　　\*全局替换图片  
　　\*  
　　\*@paramtoFindtext查找字符串