编程助手包含模块：

翻译模块:调用百度翻译api，进行英汉互译操作。

文章模块:可批量爬取文章的爬虫。

图集模块:可批量爬取图片的爬虫。

图片提取模块:调用api进行图片文字识别，F3截图，截图区域右键重新选择，截图区域外右键，呼出菜单，可进行保存，复制，提取文字等操作。也可选择图片进行文字识别。

文档引擎模块:对制定文件目录下的txt文档进行索引查询。

自动测试模块:自动化调用其他程序，可动态拼接命令并执行，也可调用命令行插件。

插件机制:插件分为两种:

一种为界面插件，基于JComponent可动态加载插件面板，执行UI操作，会在左侧动态生成插件树。

第二种为命令行插件，可动态加载后，由自动化测试模块，进行调用。

使用说明:

环境配置:

1.D盘新建tmp文件夹。

2.tmp文件夹内，新建jars文件夹，index文件夹，libs文件夹。config.xml文件，xml文件内容如下:

<config>

<srcdir>D:\mydocs</srcdir>

<indexdir>D:\tmp\index</indexdir>

<editor>E:\soft\UltraEdit\uedit64.exe</editor>

</config>

Srcdir 配置的是搜索txt文档根目录，里面可以有多个文件夹嵌套包含txt文件。

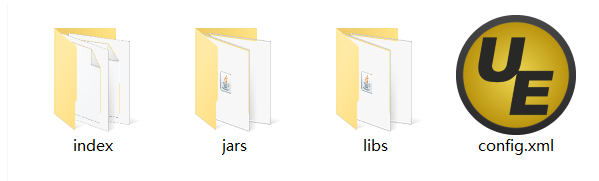
Indexdir 配置索引文件放置的目录，更新txt文件后，需要点击更新索引。

Editor 配置，搜索的结果双击打开调用的程序，如该配置双击调用UltraEdit打开搜索的结果。如删除Editor标签，则默认调用记事本打开。

jars 文件夹存放编写的插件。插件命名以包名点类名的命名方式。例如测试插件起名为com.lit.plug.MergePlug.jar。

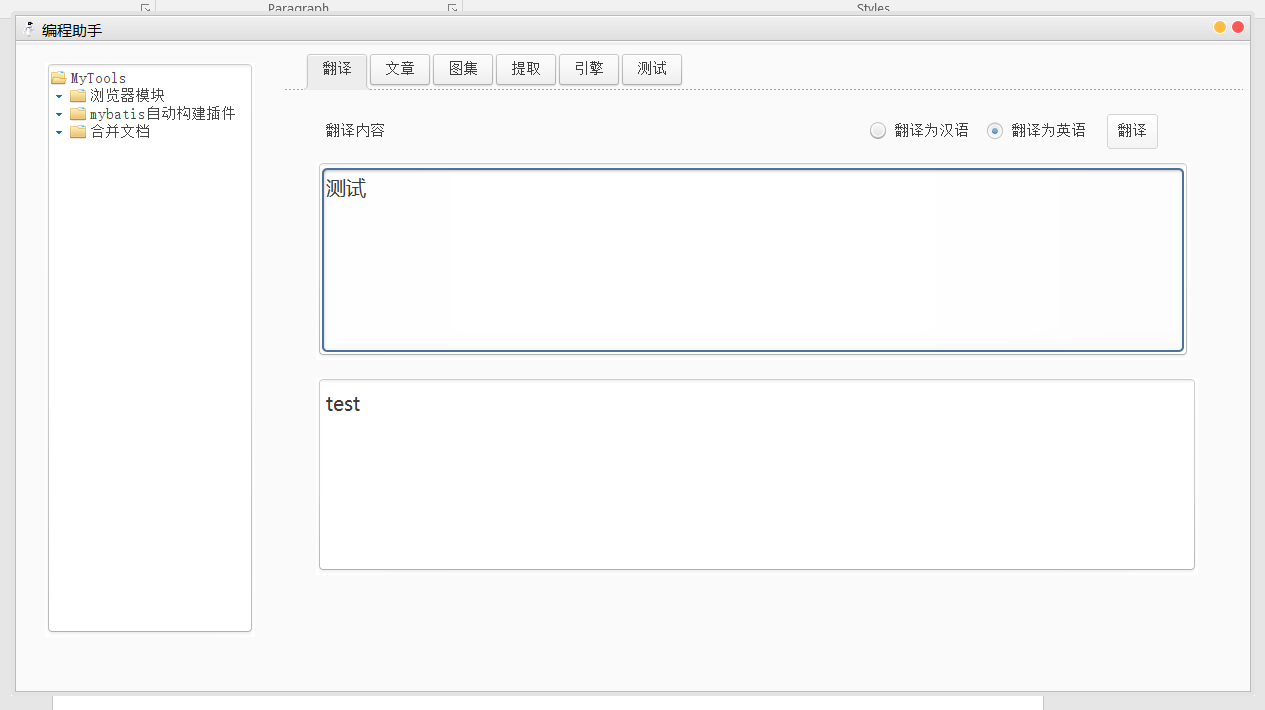
libs 文件夹存放插件初始化时所依赖的jar包。

示例截图:tmp文件夹下



模块使用说明:

翻译模块:提供英汉互译功能。



文章模块:提供批量下载文章的功能。

使用示例:

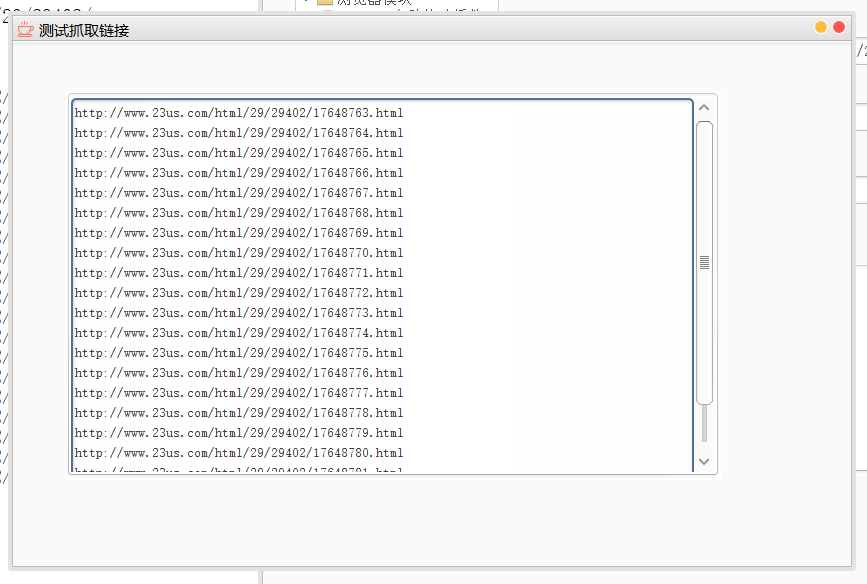
选择种子URL如 <http://www.23us.com/html/29/29402/>



类似的导航目录作为种子。

输入链接选择器 table[id=at] td>a选择文章正文的链接。

输入种子URL和链接选择器后可点击测试链接，进行验证。等待一会，爬虫进行抓取，若成功获取，则返回所有文章的链接。如图所示



打开其中的文章链接输入标题选择器和文本选择器，例如

标题选择器 div.bdsub>dl>dd

文本选择器 #contents

以及保存文件夹，点击开始抓取即可。如图所示



图集模块:提供图片批量下载的功能。

使用示例:

与文章下载类似

输入种子URL以及链接选择器

种子URL http://desk.zol.com.cn/bizhi/7052\_87534\_2.html

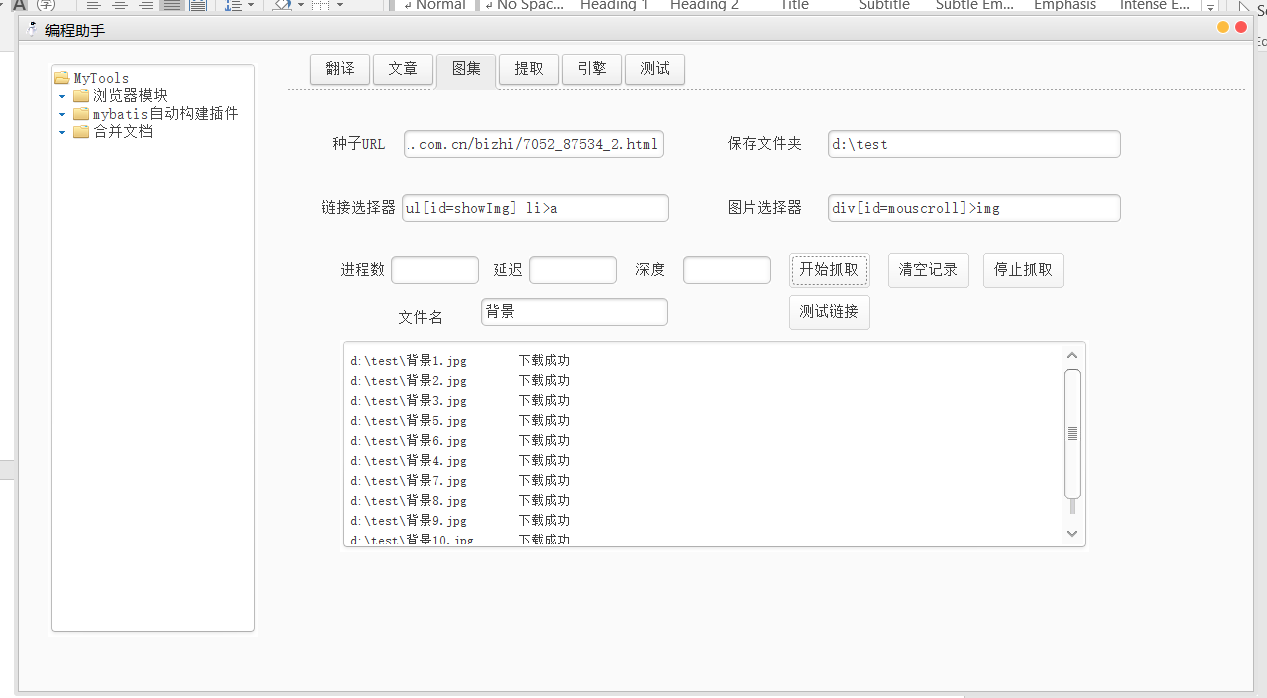
链接选择器 ul[id=showImg] li>a

点击测试链接即可进行抓取测试。

示例结果如图所示

输入图片选择器 div[id=mouscroll]>img

以及保存文件夹，点击开始抓取即可。



提取模块:提供文字提取功能。

使用方法，

快捷键F3可调出截图窗口。在选中区域内双击可复制到粘贴板，右键可重新选择，在选中区域外右键则呼出菜单，可进行图片保存，文字提取等功能。

快捷键F4，隐藏或者显示本窗体。

也可选择图片，进行文字内容的提取如图所示



文档引擎模块:提供文档内容和标题的检索功能。

需要搜索的资源目录，以及索引文件存放目录，会自动读取配置文件。D盘tmp文件夹下config.xml文件。第一次使用需先生成索引，点击生成索引按钮。待提示生成索引成功后，即可搜索。若文档内容有更新，则需重新生成索引。

可按文件标题，或者文件内容进行搜索，双击搜索结果即可调用编辑器打开。

如图所示：



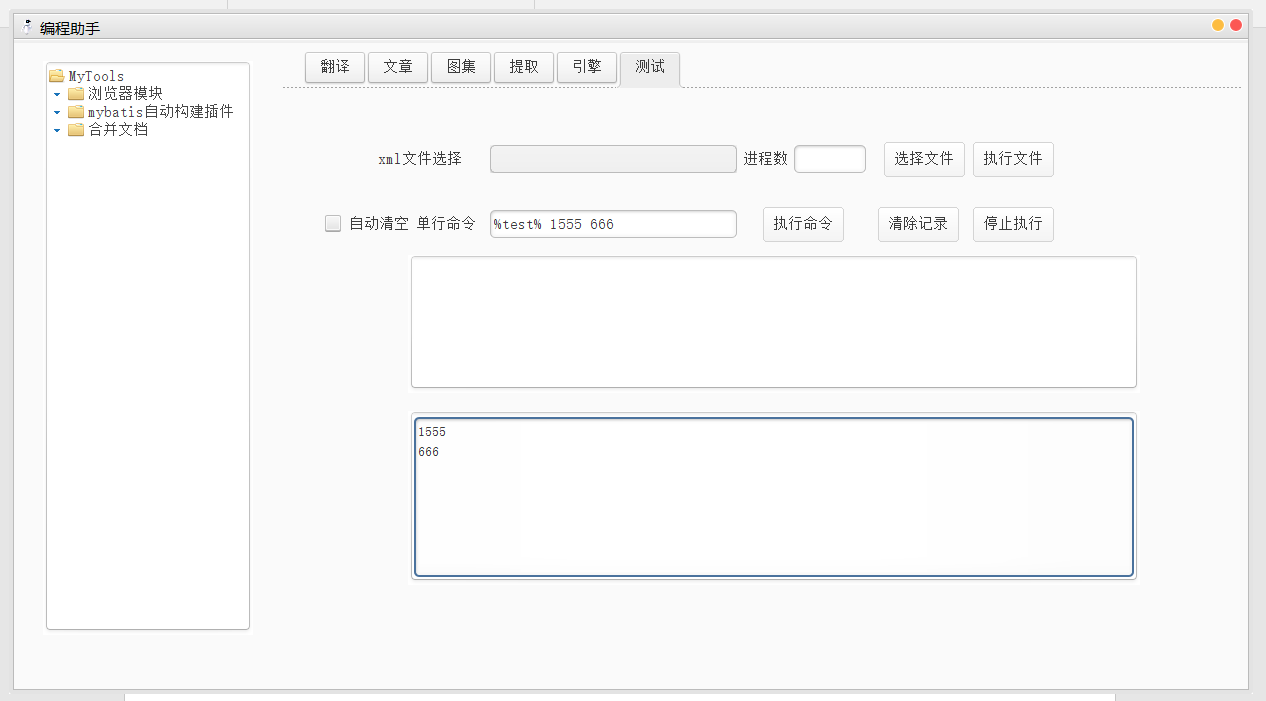
自动化测试模块:提供多线程多次调用其他程序的功能。

即自动拼接命令行，亦可自动拼接调用命令行插件。

1.调用命令行插件范例

例如，调用test插件模块，并且传递参数（test插件可自己编写，放在d:\tmp\jars文件夹下，即可自动加载。插件编写方法，后文详细说明。）

该实例已加载插件test。该插件为示范插件，将传入的参数直接传递给输出作为结果。



2.调用cmd命令范例

可直接执行cmd命令，例如执行java命令



3.可解析xml文件，拼接执行cmd命令或插件命令。

Xml文件书写说明：

xml文件包含若干个cmd标签 每个cmd标签对应一条cmd命令，cmd命令可执行多次。

cmd标签属性 context 该命令的上下文（类似于命令行下的cd到该目录） 可选

times 该命令执行的次数（必须为整数，缺省则执行一次） 可选

delay 该命令执行的延迟（必须为整数，单位毫秒） 可选

cmd命令可有子标签 可选

res 标签包裹命令段，包含部分命令的内容，多个res与其他内容拼接为一条命令。

list 标签与times配合使用，每次调用list中的一条内容，与res配合占位

有filePath属性，每次读取一行。逐行调用。每个cmd标签只能有一个list

int 标签占位一个整数，可选add属性 多次调用时自增步长，步长可为负数。

float 标签占位一个浮点数，用法同上。

均有可选参数 是否有空格nbsp=true 默认为flase

子标签需与res标签配合使用，res标签包裹静态命令，其他标签包裹动态参数。

Xml文件使用示例:

<cmd times="5">

<res>ping</res>

<list nbsp="true">

<item>127.0.0.1</item>

<item>127.0.0.2</item>

<item>127.0.0.3</item>

<item>127.0.0.4</item>

<item>127.0.0.5</item>

</list>

</cmd>

nbsp="true"属性为真，则res内容与item内容之间有空格隔开。若<res>ping </res>直接在ping后面输入空格，不用该标签则空格不起作用。

以上xml文件命令执行结果为第一次调用ping 127.0.0.1第二次调用ping 127.0.0.2以此类推。

<cmd times="5">

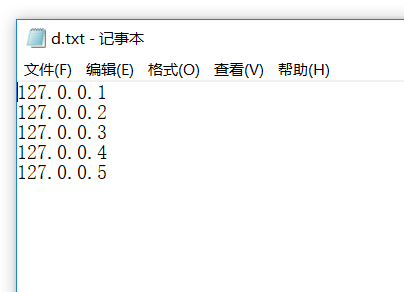
<res>ping</res>

<list nbsp="true" filePath="D:\d.txt">

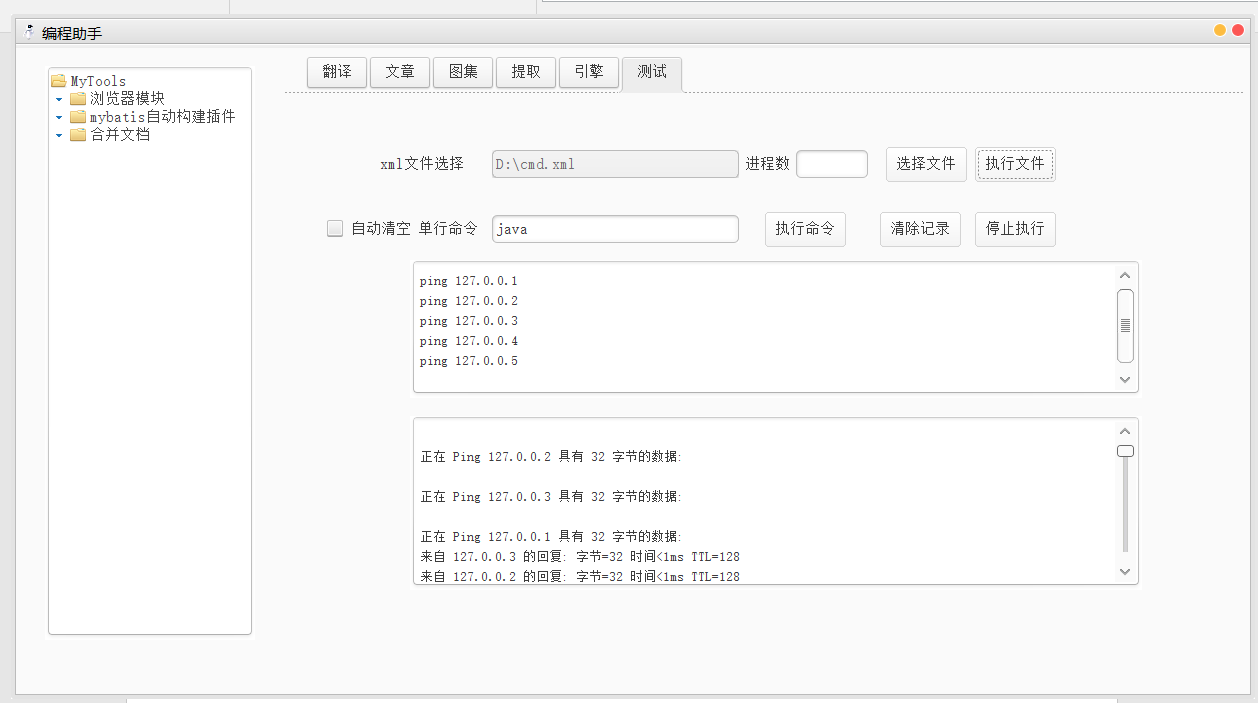
</list>

</cmd>

将ping 与D:\d.txt文件内容进行拼接成命令并执行，每次调用一行。

d.txt内容

执行结果：



Xml文件结果亦可调用模块命令。只需命令以形如%test%开头即可，中间即为需调用的模块名，若模块名不存在则会报警告提示。

例如：

该命令未写nbsp="true"属性，则直接将888与1拼接为8881

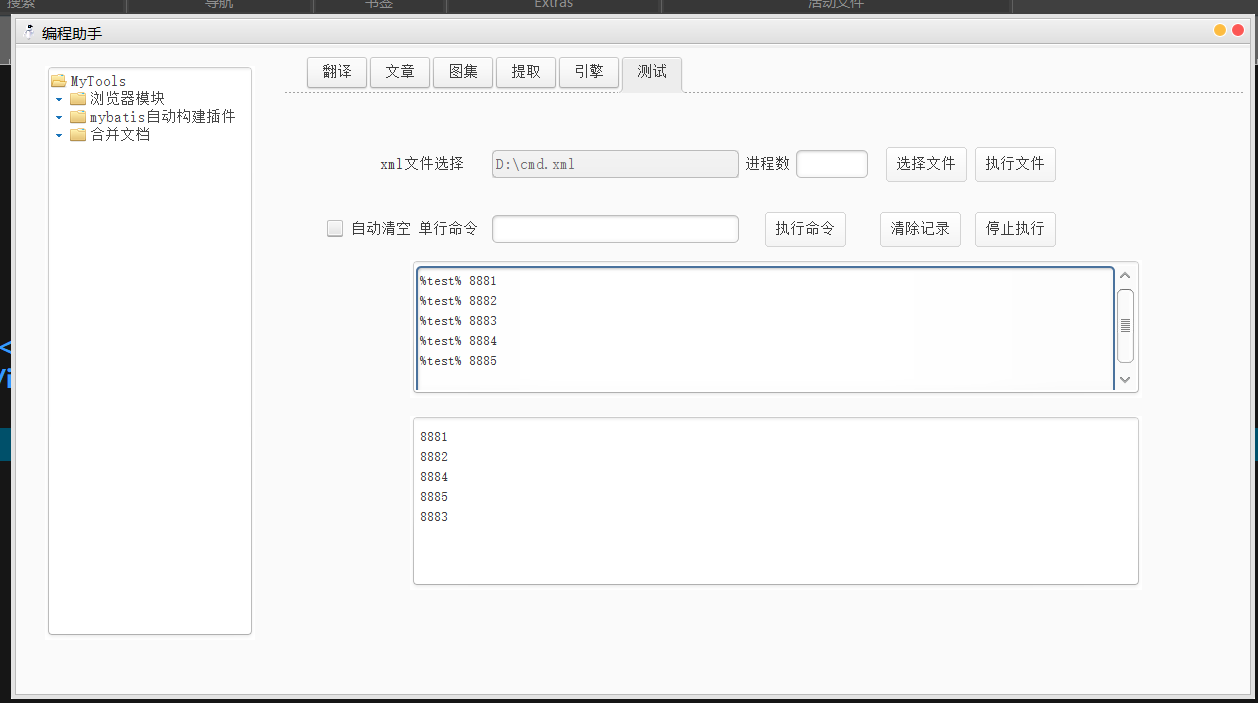
<cmd times="5">

<res>%test% 888</res>

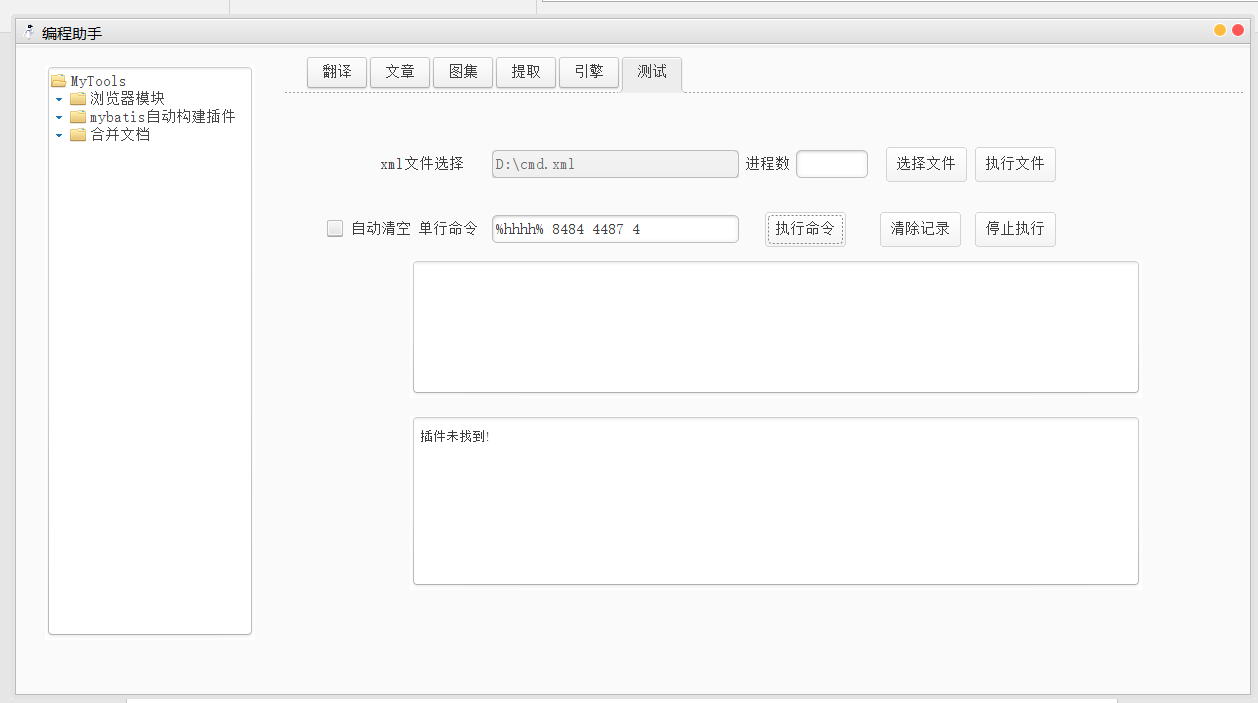
<int add="1">0</int>

</cmd>

执行结果如图所示



4.若模块不存在，则会报警告提示。



5.实际应用

与phantomjs联合应用，动态访问网页，执行js脚本，结合参数拼接功能，动态设置传递参数。

示例代码如下：

<cmd times="5" context="D:\docs\phantomjs\bin">

<res>cmd /c phantomjs.exe 127.0.0.1 test.js</res>

<int nbsp="true" add="1">0</int>

</cmd>

context="D:\docs\phantomjs\bin" 代表定位到该文件夹下

cmd /c phantomjs.exe调用phantomjs

访问127.0.0.1

执行test.js脚本，给传递参数1，执行5次，每次参数自增1。

作为自动化测试使用。

插件机制：提供两种插件机制

1. 面板插件，加载后会在左侧的插件树中进行显示。

基于JPanel，可动态加载jar包中的JPanel到工具中来使用。

使用效果如图



使用方法:

1.将mytools-sdk.jar加入build path。

2.创建插件类，继承自PlugModule类，并实现该类中的方法。

3.导出jar包文件，命名为继承自该类的包名点类名的方式，放在jars文件夹下即可。

使用实例:

package com.generator.plug;

import java.net.URLClassLoader;

import java.util.ArrayList;

import java.util.List;

import com.generator.helper.MybatisGUI;

import com.helper.inter.IHotKeyManager;

import com.helper.plugs.ModuleNode;

import com.helper.plugs.PlugModule;

//com.generator.plug.MybatisPlug

public class MybatisPlug extends PlugModule{

@Override

public String getModuleName() {

// TODO Auto-generated method stub

return "mybatis自动构建插件";

}

@Override

public List<ModuleNode> getModuleNodes(URLClassLoader arg0, IHotKeyManager arg1) {

// TODO Auto-generated method stub

List<ModuleNode> list = new ArrayList<ModuleNode>();

list.add(new ModuleNode("mysql通用构建", new MybatisGUI(arg0,arg1).getPanel()));

return list;

}

@Override

public void onWindowClose() {

// TODO Auto-generated method stub

}

@Override

public void moduleInit() {

// TODO Auto-generated method stub

}

}

导入保存的jar包名为com.generator.plug.MybatisPlug.jar

方法解释:

getModuleName设置模块名。

getModuleNodes设置插件包含的面板，一个或多个。

new ModuleNode("mysql通用构建", new MybatisGUI(arg0,arg1).getPanel())

第一个参数为面板的名称，第二个参数为JPanel类型的面板。

onWindowClose函数会在窗口将要关闭时调用。

moduleInit函数会在模块初始化时候调用一次。

getModuleNodes函数有两个参数:

URLClassLoader arg0，用于读取jar包中的资源，例如插件中根目录下有xml文件夹，文件夹下有test.xml

获取方式为arg0.getResourceAsStream("xml/test.xml");

IHotKeyManager arg1可注册全局快捷键。

1. 命令行插件，该插件加载后不会显示，但是可通过自动化测试模块进行调用。

使用方法:

1. 将mytools-sdk.jar加入build path。
2. 创建插件类，继承自CmdListener。
3. 实现其中的方法并到处jar包，以包名点类名的方式命名，放入jars文件夹。

使用实例：

package com.plug.hotkey;

import com.helper.cmd.CmdListener;

//com.plug.hotkey.Plug

public class Plug extends CmdListener{

@Override

public String getListenerName() {

// TODO Auto-generated method stub

return "test";

}

@Override

public void initListener() {

// TODO Auto-generated method stub

}

@Override

public void runCmd(String[] arg0) {

// TODO Auto-generated method stub

for(String str:arg0)

{

getCmdbacker().cmdBacker(str);

}

}

@Override

public void windowClose() {

// TODO Auto-generated method stub

}

}

将jar包到处并且保存为文件名com.plug.hotkey.Plug即可。

函数说明:

getListenerName 设置ListenerName，即调用时%test%，两个百分号中的内容。

initListener 初始化时被调用一次。

windowClose 窗口关闭时会被调用。

runCmd 类似于main函数的参数。可传递多个。当调用命令为%test% 888 999时，runCmd的参数为888和999两个字符串。ListenerName不会被传递。

在该例子中，直接调用getCmdbacker().cmdBacker(str);

将传递的参数进行结果显示，在实际使用中可执行逻辑代码，调用以上代码对执行的结果进行反馈。