1. 通用的
2. 清除cookies
3. ADSL重拨换IP
4. 验证码识别（deathbycaptcha.eu）这个网站似乎直接有提供c#的api。没有的话得自己写。
5. 取unix时间戳。这个c#应该有这个函数把？反正易语言似乎木有。可能需要13位
6. 打乱数组。
7. 写log
8. 转拼音
9. 关键词类
10. 读取关键词列表
11. 保存关键词列表
12. 从关键词列表中随机取出N个关键词（N>=1）
13. 按顺序从关键词列表中取出1个关键词（需要记录当前关键词序数）
14. 文章类
15. 文章分英文跟中文文章，其中英文文章没有要求，中文文章必须是做好分词的。这个我可以提供。类似于这样的：  
    “  
    但|不够|熟练|。|布|蕾|妮|毫不|怀疑|男孩|的|勇气|，|只是|不|放心|他|的|训练|水平|。|虽然|他|名义|上|是|个|侍从|，|但|他|侍奉|的|人|对|他|的|武艺|没有|帮助|。|  
    ”
16. 取文章：随机返回N行文本
17. 打乱文章并插入关键词：随机取出N行文本，按分词打乱文章。并插入M个关键词（M>=1）。对于插入的每个关键词个数需要保证关键词密度在2%-8%，不能太高。最后的结果需要做下处理，保证不会连续出现2个标点符号。结尾的标点符号要是“。！？”这种结尾标点。每段的开头不能有标点符号。
18. 不打乱文章并插入关键词。同3，但是不进行打乱。
19. 随机文章插入链接。这个适用于英文文章，一般用来发外链之用。随机取N行英文文本。然后随机插入M个链接代码。链接代码由参数传递。
20. 百度类
21. 登陆百度。这里需要做个判断。可能需要输入验证码。如果要验证码的话进行验证码识别。
22. 换下一个百度ID。（百度ID也同样需要记录，按顺序执行。用完了循环。）
23. 发帖（返回成功的地址）
24. 转贴（返回成功的转贴地址）。这个一般发帖成功之后点一下转贴，这样似乎容易收录点。
25. 其他就是百度的关键词列表保存，序数保存之类的。
26. 似乎现在百度改版了。老的方式似乎还可以用。新的帐号需要激活。具体还没用过，听说的。
27. 新浪类

同百度类。只是不需要转贴了。

1. 关于配置

关键词，两种文章库，各种帐号的导入需要有

关键词序数，各种帐号的序数

对于要操作的网站平台来说，（百度，新浪是用来发外链的，不算）每个帐号发几篇文章，发几个帐号后换ID，也需要做下配置。

基本就是这样。附上百度跟新浪的代码。易语言写的。话说易语言的代码怎么就这么蛋疼呢。

附一、百度

.版本 2

.支持库 eDB

.支持库 spec

.支持库 eAPI

.程序集 百度类, , 公开

.程序集变量 账号数组\_私有, 文本型, , "0"

.程序集变量 帐号行数\_私有, 整数型

.程序集变量 是否循环, 逻辑型

.程序集变量 数据库操作\_私有, 数据库读写

.程序集变量 帐号文件名\_私有, 文本型

.程序集变量 用户名\_私有, 文本型

.程序集变量 密码\_私有, 文本型

.程序集变量 ID\_私有, 文本型, , , 可空

.程序集变量 类识别\_私有, 文本型

.程序集变量 验证码字节集, 字节集

.程序集变量 验证码, 文本型

.程序集变量 编辑框log, 编辑框

.子程序 \_初始化, , , 当基于本类的对象被创建后，此方法会被自动调用

.子程序 \_销毁, , , 当基于本类的对象被销毁前，此方法会被自动调用

.子程序 log

.参数 内容, 文本型

编辑框log.加入文本 (内容 ＋ #换行符)

.子程序 初始化, , 公开

.参数 数据库地址, 文本型

.参数 数据库控件, 数据库连接

.参数 记录集控件, 记录集

.参数 帐号文件名, 文本型

.参数 是否循环取帐号, 逻辑型

.参数 识别码, 文本型, , 如百度，新浪等，用于数据库记录变量[识别符]帐号行数

.参数 log编辑框, 编辑框

数据库操作\_私有.初始化 (数据库控件, 记录集控件, 数据库地址)

帐号文件名\_私有 ＝ 帐号文件名

是否循环 ＝ 是否循环取帐号

类识别\_私有 ＝ 识别码

编辑框log ＝ log编辑框

.子程序 读取, , 公开

.局部变量 临时文本, 文本型

.局部变量 临时数组, 文本型, , "0"

.局部变量 i, 整数型

临时文本 ＝ 到文本 (读入文件 (帐号文件名\_私有))

临时数组 ＝ 分割文本 (临时文本, #换行符, )

重定义数组 (账号数组\_私有, 假, 0)

.计次循环首 (取数组下标 (临时数组, ), i)

.判断开始 (临时数组 [i] ＝ “”)

.默认

加入成员 (账号数组\_私有, 临时数组 [i])

.判断结束

.计次循环尾 ()

数据库操作\_私有.连接 ()

帐号行数\_私有 ＝ 到整数 (数据库操作\_私有.读变量 (类识别\_私有 ＋ “帐号行数”))

数据库操作\_私有.关闭 ()

调试输出 (取数组下标 (账号数组\_私有, ))

.子程序 保存, , 公开

.参数 帐号文本, 文本型, 可空

.参数 账号行数, 整数型, 可空

.判断开始 (是否为空 (帐号文本))

.默认

写到文件 (帐号文件名\_私有, 到字节集 (帐号文本))

.判断结束

.判断开始 (是否为空 (账号行数))

.默认

数据库操作\_私有.连接 ()

数据库操作\_私有.写变量 (类识别\_私有 ＋ “帐号行数”, 到文本 (账号行数))

数据库操作\_私有.关闭 ()

.判断结束

读取 ()

log (“读取完毕”)

.子程序 换ID, 逻辑型, 公开

.局部变量 临时文本, 文本型

.局部变量 临时数组, 文本型, , "0"

退出 ()

.判断开始 (帐号行数\_私有 ＞ 取数组下标 (账号数组\_私有, ))

.判断开始 (是否循环)

帐号行数\_私有 ＝ 1

.默认

返回 (假)

.判断结束

.判断 (帐号行数\_私有 ≤ 0)

帐号行数\_私有 ＝ 1

.默认

.判断结束

临时文本 ＝ 账号数组\_私有 [帐号行数\_私有]

临时数组 ＝ 分割文本 (临时文本, “|”, )

.判断开始 (取数组下标 (临时数组, ) ＜ 2)

帐号行数\_私有 ＝ 帐号行数\_私有 ＋ 1

返回 (假)

.默认

.判断结束

用户名\_私有 ＝ 临时数组 [1]

密码\_私有 ＝ 临时数组 [2]

.判断开始 (取数组下标 (临时数组, ) ＞ 2)

ID\_私有 ＝ 临时数组 [3]

.默认

.判断结束

帐号行数\_私有 ＝ 帐号行数\_私有 ＋ 1

数据库操作\_私有.连接 ()

数据库操作\_私有.写变量 (类识别\_私有 ＋ “帐号行数”, 到文本 (帐号行数\_私有))

数据库操作\_私有.关闭 ()

log (“百度换ID--用户名：” ＋ 用户名\_私有 ＋ “ 密码：” ＋ 密码\_私有)

返回 (真)

.子程序 用户名, 文本型, 公开, 返回用户名

返回 (用户名\_私有)

.子程序 密码, 文本型, 公开, 返回密码

返回 (密码\_私有)

.子程序 ID, 文本型, 公开, 返回ID

返回 (ID\_私有)

.子程序 取验证码, 逻辑型, , 判断是否需要验证码

.参数 返回验证码字节集, 字节集, 参考 可空

.参数 返回内容, 文本型, 参考 可空

.局部变量 编码后用户名, 文本型

编码后用户名 ＝ 用户名\_私有

编码后用户名 ＝ 编码\_URL编码GBK (编码后用户名)

返回内容 ＝ 到文本 (网页\_访问 (“https://passport.baidu.com/?logcheck&username=” ＋ 编码后用户名))

' loginCheck("0");不需要验证码 loginCheck("1");需要验证码

.判断开始 (寻找文本 (返回内容, “1”, , 假) ＞ 0)

' 需要验证码

log (“需要验证码”)

验证码字节集 ＝ 网页\_验证码读取 (“https://passport.baidu.com/?verifypic”)

返回验证码字节集 ＝ 验证码字节集

返回 (真)

.默认

' 不需要验证码

log (“不需要验证码”)

验证码字节集 ＝ { }

验证码 ＝ “”

返回验证码字节集 ＝ 验证码字节集

返回 (假)

.判断结束

.子程序 识别, 文本型, , 识别验证码

调试输出 (验证码字节集)

.判断开始 (验证码字节集 ≠ { })

wc\_验证码识别\_deathbycaptcha (验证码字节集, “wbxfire”, “wbxuo235”, 验证码)

log (“验证码识别结果：” ＋ 验证码)

.默认

验证码 ＝ “”

.判断结束

返回 (验证码)

.子程序 账号列表, 文本型, 公开, 返回帐号列表

返回 (到文本 (读入文件 (帐号文件名\_私有)))

.子程序 当前帐号行数, 整数型, 公开, 返回当前帐号行数

返回 (帐号行数\_私有)

.子程序 置帐号, , 公开

.参数 用户名, 文本型

.参数 密码, 文本型

用户名\_私有 ＝ 用户名

密码\_私有 ＝ 密码

.子程序 登录, 逻辑型, 公开

.参数 用户名, 文本型, 参考 可空, 用户名

.参数 密码, 文本型, 参考 可空, 密码

.参数 状态文本, 文本型, 参考 可空

.局部变量 strPost, 文本型

.局部变量 xmlhttp, 类\_XMLHTTP对象

.局部变量 strUrl, 文本型

.局部变量 tmpStr, 文本型

.局部变量 vbsRegExp, 类\_正则表达式

.局部变量 iCount, 整数型

.局部变量 bRet, 逻辑型

.局部变量 编码后用户名, 文本型

取验证码 ()

识别 ()

' 退出 ()

编码后用户名 ＝ 用户名\_私有

编码后用户名 ＝ 彗星URL编码 (编码后用户名)

bRet ＝ 假

.判断开始 (验证码 ＝ “”)

strPost ＝ “tpl\_ok=&next\_target=&tpl=mn&skip\_ok=&aid=&need\_pay=&need\_coin=&pay\_method=&u=http%3A%2F%2Fwww.baidu.com%2F2F&return\_method=get&more\_param=&return\_type=&psp\_tt=0&password=” ＋ 密码\_私有 ＋ “&safeflg=0&isphone=tpl&username=” ＋ 编码后用户名 ＋ “&verifycode=&mem\_pass=o”

.默认

strPost ＝ “tpl\_ok=&next\_target=&tpl=mn&skip\_ok=&aid=&need\_pay=&need\_coin=&pay\_method=&u=http%3A%2F%2Fwww.baidu.com%2F2F&return\_method=get&more\_param=&return\_type=&psp\_tt=0&password=” ＋ 密码\_私有 ＋ “&safeflg=0&isphone=tpl&username=” ＋ 编码后用户名 ＋ “&verifycode=” ＋ 验证码 ＋ “&mem\_pass=on”

.判断结束

strUrl ＝ “https://passport.baidu.com/?login”

xmlhttp.Open (strUrl, “POST”, 假)

xmlhttp.setRequestHeader (“Referer”, “https://passport.baidu.com/?login”)

xmlhttp.setRequestHeader (“Host”, “passport.baidu.com”)

xmlhttp.setRequestHeader (“Content-Type”, “application/x-www-form-urlencoded”)

xmlhttp.setRequestHeader (“Accept”, “\*/\*”)

xmlhttp.Send\_Str (strPost)

tmpStr ＝ 到文本 (xmlhttp.responseBody ())

调试输出 (tmpStr)

调试输出 (strPost)

状态文本 ＝ tmpStr

vbsRegExp.创建 (“location.href=”, 假, 真)

用户名 ＝ 用户名\_私有

密码 ＝ 密码\_私有

iCount ＝ vbsRegExp.搜索全部 (tmpStr)

状态文本 ＝ tmpStr

.如果真 (iCount ＞ 0)

log (“登录成功”)

返回 (真)

.如果真结束

log (“登录失败”)

返回 (假)

.子程序 退出, , 公开

.局部变量 源码, 文本型

网页\_访问 (“http://passport.baidu.com/?logout&aid=7&u=http%3A//wenku.baidu.com/”, 0, , , , )

删除临时文件 (1)

wc\_清除cookie (“baidu.com”)

wc\_清除cookie (“passport.baidu.com”)

wc\_清除cookie (“www.baidu.com”)

wc\_清除cookie (“hi.baidu.com”)

wc\_清除cookie (“apps.baidu.com”)

wc\_清除cookie (“.baidu.com”)

.子程序 获取Token, 文本型, , \_tb\_token\_

.参数 获取目标网址, 文本型

.局部变量 strUrl, 文本型

.局部变量 strTmp, 文本型

.局部变量 vbsRegExp, 类\_正则表达式

.局部变量 iCount, 整数型

.局部变量 xmlhttp, 类\_XMLHTTP对象

.局部变量 strRet, 文本型

strUrl ＝ 获取目标网址

xmlhttp.Open (strUrl, “GET”, 假)

xmlhttp.Send ()

strTmp ＝ 到文本 (xmlhttp.responseBody ())

' 编辑框4.内容 ＝ strTmp

vbsRegExp.创建 (“<input[^h]\*?hidden[^b]\*?bdstoken[^v]\*?value=[” ＋ #引号 ＋ “]\*?([^>]+?)[” ＋ #引号 ＋ “]\*?>”, 假, 真)

iCount ＝ vbsRegExp.搜索全部 (strTmp)

.判断开始 (iCount ＞ 0)

strRet ＝ vbsRegExp.取子匹配文本 (1, 1)

.默认

.判断结束

strRet ＝ 子文本替换 (strRet, #引号, “”, , , 真)

strRet ＝ 子文本替换 (strRet, “ ”, “”, , , 真)

返回 (strRet)

.子程序 发贴, 逻辑型, 公开, post发帖。

.参数 b标题, 文本型, , 标题

.参数 b分类, 文本型, 可空, 分类。可空。默认为“默认分类”

.参数 b内容, 文本型, , 内容。HTML格式

.参数 返回内容, 文本型, 参考 可空

.参数 文章地址, 文本型, 参考 可空, 文章地址

.局部变量 strPost, 文本型

.局部变量 xmlhttp, 类\_XMLHTTP对象

.局部变量 strUrl, 文本型

.局部变量 tmpStr, 文本型

.局部变量 vbsRegExp, 类\_正则表达式

.局部变量 iCount, 整数型

.局部变量 bRet, 逻辑型

.局部变量 博客标题, 文本型

.局部变量 博客分类, 文本型

.局部变量 博客内容, 文本型

.局部变量 编码后用户名, 文本型

.局部变量 发帖成功地址, 文本型

.局部变量 提交地址, 文本型

.局部变量 提交内容, 文本型

.局部变量 Token, 文本型

.局部变量 shid, 文本型

.局部变量 延时时间, 整数型

.局部变量 用户名, 文本型, , , 要发帖的用户名

用户名 ＝ 用户名\_私有

bRet ＝ 假

博客标题 ＝ b标题

博客标题 ＝ 彗星URL编码 (博客标题)

.判断开始 (是否为空 (b分类) ＝ 真)

博客分类 ＝ 彗星URL编码 (“默认分类”)

.默认

博客分类 ＝ b分类

博客分类 ＝ 彗星URL编码 (博客分类)

.判断结束

博客内容 ＝ b内容

博客内容 ＝ 彗星URL编码 (博客内容)

编码后用户名 ＝ 用户名

编码后用户名 ＝ 彗星URL编码 (编码后用户名)

Token ＝ 获取Token (“http://hi.baidu.com/” ＋ 编码后用户名 ＋ “/blog/create”)

log (Token)

strUrl ＝ “http://hi.baidu.com/” ＋ 用户名 ＋ “/blog/submit/createblog”

strPost ＝ “bdstoken=” ＋ Token ＋ “&ct=1&cm=1&spBlogID=&spBlogCatName\_o=&edithid=&previewImg=&spBlogTitle=” ＋ 博客标题 ＋ “&spBlogText=%3CP%3E” ＋ 博客内容 ＋ “%3C%2FP%3E&spBlogCatName=%C4%AC%C8%CF%B7%D6%C0%E0&spBlogPower=0&spIsCmtAllow=1&spShareNotAllow=0&spVcode=&spVerifyKey=”

tmpStr ＝ 彗星HTTP读文本 (strUrl, “POST”, , , strPost, )

返回内容 ＝ tmpStr ＋ #换行符 ＋ “发送内容：” ＋ strPost ＋ #换行符 ＋ “发送地址：” ＋ strUrl

.如果真 (寻找文本 (tmpStr, “发表成功”, , 假) ＝ -1)

log (“发表失败”)

返回 (假)

.如果真结束

发帖成功地址 ＝ 文本\_取出中间文本 (tmpStr, “/blog/item/”, “.html”)

文章地址 ＝ “http://hi.baidu.com/” ＋ 编码后用户名 ＋ “/blog/item/” ＋ 发帖成功地址 ＋ “.html”

log (“发表成功：” ＋ 文章地址)

返回 (真)

.子程序 发贴并转贴, 逻辑型, 公开, post发帖。

.参数 b标题, 文本型, , 标题

.参数 b分类, 文本型, 可空, 分类。可空。默认为“默认分类”

.参数 b内容, 文本型, , 内容。HTML格式

.参数 延时1, 整数型, 可空, 发帖后延时多少ms转贴，默认1秒

.参数 返回内容, 文本型, 参考 可空

.参数 文章地址, 文本型, 参考 可空, 文章地址

.局部变量 strPost, 文本型

.局部变量 xmlhttp, 类\_XMLHTTP对象

.局部变量 strUrl, 文本型

.局部变量 tmpStr, 文本型

.局部变量 vbsRegExp, 类\_正则表达式

.局部变量 iCount, 整数型

.局部变量 bRet, 逻辑型

.局部变量 博客标题, 文本型

.局部变量 博客分类, 文本型

.局部变量 博客内容, 文本型

.局部变量 编码后用户名, 文本型

.局部变量 发帖成功地址, 文本型

.局部变量 提交地址, 文本型

.局部变量 提交内容, 文本型

.局部变量 Token, 文本型

.局部变量 shid, 文本型

.局部变量 延时时间, 整数型

.局部变量 用户名, 文本型, , , 要发帖的用户名

用户名 ＝ 用户名\_私有

.判断开始 (是否为空 (延时1))

延时时间 ＝ 1000 × 1000

.默认

延时时间 ＝ 延时1 × 1000

.判断结束

bRet ＝ 假

博客标题 ＝ b标题

博客标题 ＝ 编码\_URL编码GBK (博客标题)

.判断开始 (是否为空 (b分类) ＝ 真)

博客分类 ＝ 编码\_URL编码GBK (“默认分类”)

.默认

博客分类 ＝ b分类

博客分类 ＝ 编码\_URL编码GBK (博客分类)

.判断结束

博客内容 ＝ b内容

博客内容 ＝ 编码\_URL编码GBK (博客内容)

编码后用户名 ＝ 用户名

编码后用户名 ＝ 彗星URL编码 (编码后用户名)

Token ＝ 获取Token (“http://hi.baidu.com/” ＋ 编码后用户名 ＋ “/blog/create”)

log (Token)

strUrl ＝ “http://hi.baidu.com/” ＋ 用户名 ＋ “/blog/submit/createblog”

strPost ＝ “bdstoken=” ＋ Token ＋ “&ct=1&cm=1&spBlogID=&spBlogCatName\_o=&edithid=&previewImg=&spBlogTitle=” ＋ 博客标题 ＋ “&spBlogText=%3CP%3E” ＋ 博客内容 ＋ “%3C%2FP%3E&spBlogCatName=%C4%AC%C8%CF%B7%D6%C0%E0&spBlogPower=0&spIsCmtAllow=1&spShareNotAllow=0&spVcode=&spVerifyKey=”

tmpStr ＝ 彗星HTTP读文本 (strUrl, “POST”, , , strPost, )

返回内容 ＝ tmpStr ＋ #换行符 ＋ “发送内容：” ＋ strPost ＋ #换行符 ＋ “发送地址：” ＋ strUrl

.如果真 (寻找文本 (tmpStr, “发表成功”, , 假) ＝ -1)

log (“发表失败”)

返回 (假)

.如果真结束

发帖成功地址 ＝ 文本\_取出中间文本 (tmpStr, “/blog/item/”, “.html”)

文章地址 ＝ “http://hi.baidu.com/” ＋ 编码后用户名 ＋ “/blog/item/” ＋ 发帖成功地址 ＋ “.html”

log (“发表成功：” ＋ 文章地址)

' ------------------------------------------------

' 转贴

高精度等待 (延时时间)

博客标题 ＝ b标题

博客标题 ＝ 编码\_URL编码UTF8 (博客标题)

.判断开始 (是否为空 (b分类) ＝ 真)

博客分类 ＝ 编码\_URL编码UTF8 (“默认分类”)

.默认

博客分类 ＝ b分类

博客分类 ＝ 编码\_URL编码UTF8 (博客分类)

.判断结束

博客内容 ＝ b内容

博客内容 ＝ 编码\_URL编码UTF8 (博客内容)

编码后用户名 ＝ 用户名

编码后用户名 ＝ 彗星URL编码 (编码后用户名)

log (文章地址)

' 第一个包

提交地址 ＝ “http://apps.hi.baidu.com/share/cm/”

提交内容 ＝ “\_method=create\_share&dtype=json&ie=utf-8&url=http%3A%2F%2Fhi.baidu.com%2F” ＋ 编码后用户名 ＋ “%2Fblog%2Fitem%2F” ＋ 发帖成功地址 ＋ “.html&auth=0&zzToBlog=1&title=” ＋ 博客标题 ＋ “&comment\_content=&option=on&shid=&vote\_opt=%E5%88%86%E4%BA%AB%E6%98%AF%E7%BE%8E%E5%BE%B7~&bdstoken=” ＋ Token

xmlhttp.Open (提交地址, “post”, 假)

xmlhttp.setRequestHeader (“Referer”, “http%3A%2F%2Fhi.baidu.com%2F” ＋ 编码后用户名 ＋ “%2Fblog%2Fitem%2F” ＋ 发帖成功地址 ＋ “.html”)

xmlhttp.setRequestHeader (“Origin”, “http://apps.hi.baidu.com”)

xmlhttp.setRequestHeader (“Content-Type”, “application/x-www-form-urlencoded”)

xmlhttp.setRequestHeader (“X-Request-By”, “baidu.ajax”)

xmlhttp.setRequestHeader (“Accept”, “\*/\*”)

xmlhttp.setRequestHeader (“Accept-Encoding”, “gzip,deflate”)

xmlhttp.setRequestHeader (“Accept-Language”, “zh-CN”)

xmlhttp.setRequestHeader (“Accept-Charset”, “iso-8859-1,\*,utf-8”)

xmlhttp.Send\_Str (提交内容)

tmpStr ＝ xmlhttp.responseText ()

shid ＝ 文本\_取出中间文本 (tmpStr, “shid” ＋ #引号 ＋ “:”, “,”)

调试输出 (tmpStr)

调试输出 (shid)

' 第二个包

提交内容 ＝ “\_method=create\_comment&shid=” ＋ shid ＋ “&content=&async=false&dtype=json&ie=utf-8&bdstoken=” ＋ Token

xmlhttp.Open (提交地址, “post”, 假)

xmlhttp.setRequestHeader (“Referer”, “http%3A%2F%2Fhi.baidu.com%2F” ＋ 编码后用户名 ＋ “%2Fblog%2Fitem%2F” ＋ 发帖成功地址 ＋ “.html”)

xmlhttp.setRequestHeader (“Origin”, “http://apps.hi.baidu.com”)

xmlhttp.setRequestHeader (“Content-Type”, “application/x-www-form-urlencoded”)

xmlhttp.setRequestHeader (“X-Request-By”, “baidu.ajax”)

xmlhttp.setRequestHeader (“Accept”, “\*/\*”)

xmlhttp.setRequestHeader (“Accept-Encoding”, “gzip,deflate”)

xmlhttp.setRequestHeader (“Accept-Language”, “zh-CN”)

xmlhttp.setRequestHeader (“Accept-Charset”, “iso-8859-1,\*,utf-8”)

xmlhttp.Send\_Str (提交内容)

tmpStr ＝ xmlhttp.responseText ()

' 第三个包

提交内容 ＝ “\_method=create\_tag&shid=” ＋ shid ＋ “&tag\_text=a&async=false&dtype=json&ie=utf-8&bdstoken=” ＋ Token

xmlhttp.Open (提交地址, “post”, 假)

xmlhttp.setRequestHeader (“Referer”, “http%3A%2F%2Fhi.baidu.com%2F” ＋ 编码后用户名 ＋ “%2Fblog%2Fitem%2F” ＋ 发帖成功地址 ＋ “.html”)

xmlhttp.setRequestHeader (“Origin”, “http://apps.hi.baidu.com”)

xmlhttp.setRequestHeader (“Content-Type”, “application/x-www-form-urlencoded”)

xmlhttp.setRequestHeader (“X-Request-By”, “baidu.ajax”)

xmlhttp.setRequestHeader (“Accept”, “\*/\*”)

xmlhttp.setRequestHeader (“Accept-Encoding”, “gzip,deflate”)

xmlhttp.setRequestHeader (“Accept-Language”, “zh-CN”)

xmlhttp.setRequestHeader (“Accept-Charset”, “iso-8859-1,\*,utf-8”)

xmlhttp.Send\_Str (提交内容)

tmpStr ＝ xmlhttp.responseText ()

返回 (真)

附录二、新浪

.版本 2

.支持库 eDB

.程序集 新浪类, , 公开

.程序集变量 账号数组\_私有, 文本型, , "0"

.程序集变量 帐号行数\_私有, 整数型

.程序集变量 是否循环, 逻辑型

.程序集变量 数据库操作\_私有, 数据库读写

.程序集变量 帐号文件名\_私有, 文本型

.程序集变量 用户名\_私有, 文本型

.程序集变量 密码\_私有, 文本型

.程序集变量 ID\_私有, 文本型, , , 可空

.程序集变量 类识别\_私有, 文本型

.程序集变量 编辑框log, 编辑框

.子程序 \_初始化, , , 当基于本类的对象被创建后，此方法会被自动调用

.子程序 \_销毁, , , 当基于本类的对象被销毁前，此方法会被自动调用

.子程序 初始化, , 公开

.参数 数据库地址, 文本型

.参数 数据库控件, 数据库连接

.参数 记录集控件, 记录集

.参数 帐号文件名, 文本型

.参数 是否循环取帐号, 逻辑型

.参数 识别码, 文本型, , 如百度，新浪等，用于数据库记录变量[识别符]帐号行数

.参数 log编辑框, 编辑框

数据库操作\_私有.初始化 (数据库控件, 记录集控件, 数据库地址)

帐号文件名\_私有 ＝ 帐号文件名

是否循环 ＝ 是否循环取帐号

类识别\_私有 ＝ 识别码

编辑框log ＝ log编辑框

.子程序 log

.参数 内容, 文本型

编辑框log.加入文本 (内容 ＋ #换行符)

.子程序 读取, , 公开

.局部变量 临时文本, 文本型

.局部变量 临时数组, 文本型, , "0"

.局部变量 i, 整数型

临时文本 ＝ 到文本 (读入文件 (帐号文件名\_私有))

临时数组 ＝ 分割文本 (临时文本, #换行符, )

重定义数组 (账号数组\_私有, 假, 0)

.计次循环首 (取数组下标 (临时数组, ), i)

.判断开始 (临时数组 [i] ＝ “”)

.默认

加入成员 (账号数组\_私有, 临时数组 [i])

.判断结束

.计次循环尾 ()

数据库操作\_私有.连接 ()

帐号行数\_私有 ＝ 到整数 (数据库操作\_私有.读变量 (类识别\_私有 ＋ “帐号行数”))

数据库操作\_私有.关闭 ()

log (“新浪读取完毕”)

.子程序 保存, , 公开

.参数 帐号文本, 文本型, 可空

.参数 账号行数, 整数型, 可空

.判断开始 (是否为空 (帐号文本))

.默认

写到文件 (帐号文件名\_私有, 到字节集 (帐号文本))

.判断结束

.判断开始 (是否为空 (账号行数))

.默认

数据库操作\_私有.连接 ()

数据库操作\_私有.写变量 (类识别\_私有 ＋ “帐号行数”, 到文本 (账号行数))

数据库操作\_私有.关闭 ()

.判断结束

log (“新浪保存完毕”)

读取 ()

.子程序 换ID, 逻辑型, 公开

.局部变量 临时文本, 文本型

.局部变量 临时数组, 文本型, , "0"

退出 ()

.判断开始 (帐号行数\_私有 ＞ 取数组下标 (账号数组\_私有, ))

.判断开始 (是否循环)

帐号行数\_私有 ＝ 1

.默认

返回 (假)

.判断结束

.判断 (帐号行数\_私有 ≤ 0)

帐号行数\_私有 ＝ 1

.默认

.判断结束

临时文本 ＝ 账号数组\_私有 [帐号行数\_私有]

临时数组 ＝ 分割文本 (临时文本, “|”, )

.判断开始 (取数组下标 (临时数组, ) ＜ 2)

帐号行数\_私有 ＝ 帐号行数\_私有 ＋ 1

返回 (假)

.默认

.判断结束

用户名\_私有 ＝ 临时数组 [1]

密码\_私有 ＝ 临时数组 [2]

.判断开始 (取数组下标 (临时数组, ) ＞ 2)

ID\_私有 ＝ 临时数组 [3]

.默认

.判断结束

帐号行数\_私有 ＝ 帐号行数\_私有 ＋ 1

数据库操作\_私有.连接 ()

数据库操作\_私有.写变量 (类识别\_私有 ＋ “帐号行数”, 到文本 (帐号行数\_私有))

数据库操作\_私有.关闭 ()

log (“新浪换ID--用户名：” ＋ 用户名\_私有 ＋ “ 密码：” ＋ 密码\_私有)

返回 (真)

.子程序 用户名, 文本型, 公开

返回 (用户名\_私有)

.子程序 密码, 文本型, 公开

返回 (密码\_私有)

.子程序 ID, 文本型, 公开

返回 (ID\_私有)

.子程序 账号列表, 文本型, 公开

返回 (到文本 (读入文件 (帐号文件名\_私有)))

.子程序 当前帐号行数, 整数型, 公开

返回 (帐号行数\_私有)

.子程序 退出, , 公开

log (“清除sina-cookies”)

wc\_清除cookie (“.sina.com.cn”)

wc\_清除cookie (“blog.sina.com.cn”)

wc\_清除cookie (“www.sina.com.cn”)

.子程序 登录, 逻辑型, 公开

.参数 新浪用户名, 文本型, 参考 可空

.参数 新浪密码, 文本型, 参考 可空

.参数 返回内容, 文本型, 参考 可空

.局部变量 post, 文本型

.局部变量 返回值, 文本型

.局部变量 http, http\_https

退出 ()

新浪用户名 ＝ 用户名\_私有

新浪密码 ＝ 密码\_私有

post ＝ “entry=sso&gateway=1&from=&savestate=0&useticket=0&username=” ＋ 新浪用户名 ＋ “&service=sso&password=” ＋ 新浪密码 ＋ “&encoding=GB2312&callback=parent.sinaSSOController.loginCallBack&returntype=IFRAME&setdomain=1”

到文本 (http.提交数据\_SSL (“https://login.sina.com.cn/sso/login.php?client=ssologin.js(v1.3.14)”, post))

返回值 ＝ 到文本 (http.读取数据 (“http://login.sina.com.cn/member/my.php?entry=sso”))

返回内容 ＝ 返回值

.如果 (寻找文本 (返回值, “您好”, , 假) ≥ 1)

log (“登录成功”)

返回 (真)

.否则

log (“登录失败”)

返回 (假)

.如果结束

.子程序 取发帖参数, 文本型

.局部变量 源码, 文本型

.局部变量 临时文本数组, 文本型, , "0"

.局部变量 name, 文本型, , "0"

.局部变量 value, 文本型, , "0"

.局部变量 个数, 整数型

.局部变量 i, 整数型

.局部变量 返回文本, 文本型

.局部变量 time, 文本型

源码 ＝ 编码\_utf8到gb2312 (网页\_取网页源码 (“http://control.blog.sina.com.cn/admin/article/article\_add.php”))

' <meta http-equiv="Expires" CONTENT="-1" >

' <meta http-equiv="Cache-Control" CONTENT="no-cache" >

' <meta http-equiv="Cache-Control" CONTENT="no-store" >

' <meta http-equiv="Pragma" CONTENT="no-cache" ><script>window.location.href='http://control.blog.sina.com.cn/myblog/htmlsource/blog\_notopen.php?uid=2326263480&version=7';</script>

' <input id="articleTime" name="time" type="text" size="8" value="18:55:58" class="Fm\_input1" />

time ＝ 文本\_取出中间文本 (源码, “articleTime”, “/>”)

time ＝ 文本\_取出中间文本 (time, “value=” ＋ #引号, #引号)

time ＝ 文本\_替换 (time, , , , “:”, “%3A”)

源码 ＝ 文本\_取出中间文本 (源码, “editorForm”, “<!-- 内容开始 -->”)

重定义数组 (临时文本数组, 假, 0)

' <input type="hidden" name="date\_pub" id="date\_pub" value="2011-08-20"/>

文本\_取中间\_批量 (源码, “hidden”, “/>”, 临时文本数组)

个数 ＝ 取数组下标 (临时文本数组, )

返回文本 ＝ “”

重定义数组 (name, 假, 个数)

重定义数组 (value, 假, 个数)

.计次循环首 (个数, i)

.判断开始 (寻找文本 (临时文本数组 [i], “disabled”, , 真) ＞ 0)

到循环尾 ()

.默认

.判断结束

name [i] ＝ 文本\_取出中间文本 (临时文本数组 [i], “name=” ＋ #引号, #引号)

value [i] ＝ 文本\_取出中间文本 (临时文本数组 [i], “value=” ＋ #引号, #引号)

返回文本 ＝ 返回文本 ＋ name [i] ＋ “=” ＋ value [i] ＋ “&”

.计次循环尾 ()

返回文本 ＝ 返回文本 ＋ “time=” ＋ time

.判断开始 (个数 ＜ 3)

返回 (“”)

.默认

返回 (返回文本)

.判断结束

.子程序 发帖, 逻辑型, 公开

.参数 传递标题, 文本型

.参数 传递内容, 文本型, , html

.参数 传递标签, 文本型

.参数 返回内容, 文本型, 参考 可空

.参数 返回文章ID, 文本型, 参考 可空, http://blog.sina.com.cn/s/blog\_[ID].html

.局部变量 提交地址, 文本型

.局部变量 提交内容, 文本型

.局部变量 协议头, 文本型

.局部变量 参数, 文本型

.局部变量 超时, 整数型

.局部变量 conlen, 整数型

' blog\_title=[title]&blog\_body=[content]&blog\_class=00&tag=[tag]&x\_cms\_flag=0&sina\_sort\_id=117&join\_circle=1

提交地址 ＝ “http://control.blog.sina.com.cn/admin/article/article\_post.php”

协议头 ＝ “Accept: \*/\*” ＋ #换行符 ＋ “Referer: http://control.blog.sina.com.cn/admin/article/article\_add.php?index” ＋ #换行符 ＋ “Accept-Language: zh-CN” ＋ #换行符 ＋ “Content-Type: application/x-www-form-urlencoded” ＋ #换行符

参数 ＝ “”

超时 ＝ 0

.判断循环首 (参数 ＝ “”)

.判断开始 (超时 ＞ 10)

返回 (假)

.默认

.判断结束

参数 ＝ 取发帖参数 ()

高精度等待 (1000 × 500)

超时 ＝ 超时 ＋ 1

.判断循环尾 ()

conlen ＝ 文本\_取长度 (传递内容)

提交内容 ＝ 参数 ＋ “&blog\_title=” ＋ 彗星URL编码\_UTF8 (传递标题) ＋ “&blog\_body=” ＋ 彗星URL编码\_UTF8 (传递内容) ＋ “&blog\_class=00&tag=” ＋ 彗星URL编码\_UTF8 (传递标签) ＋ “&x\_cms\_flag=0&sina\_sort\_id=117&join\_circle=1”

' conlen=&

提交内容 ＝ 文本\_替换 (提交内容, , , , “conlen=&”, “conlen=” ＋ 到文本 (conlen) ＋ “&”)

返回内容 ＝ 到文本 (网页\_访问 (提交地址, 1, , 协议头, 到字节集 (提交内容), , , , 1))

log (返回内容)

' {"code":"B06001","data":"4fc53be70100u7el","ycon\_length":38,"jcon\_length":"38"}

返回文章ID ＝ 文本\_取出中间文本 (返回内容, “data” ＋ #引号 ＋ “:” ＋ #引号, #引号)

.判断开始 (返回文章ID ＝ “”)

log (“发表失败”)

返回 (假)

log (“发表成功:” ＋ “http://blog.sina.com.cn/s/blog\_” ＋ 返回文章ID ＋ “.html”)

.默认

返回 (真)

.判断结束