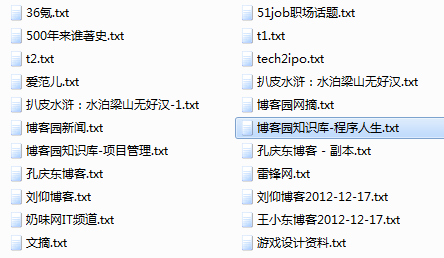
爱学习，爱使用移动设备阅读电子书的朋友不能不拥有一款属于自己的采集利器。

而使用此程序即可以简单轻松的实现采集任务。



采集效果图



程序介绍

在程序运行子目录 **Config** 是程序的配置的保存目录。

**path.txt**

[config]

Config\HtmlFormatConfig.xml

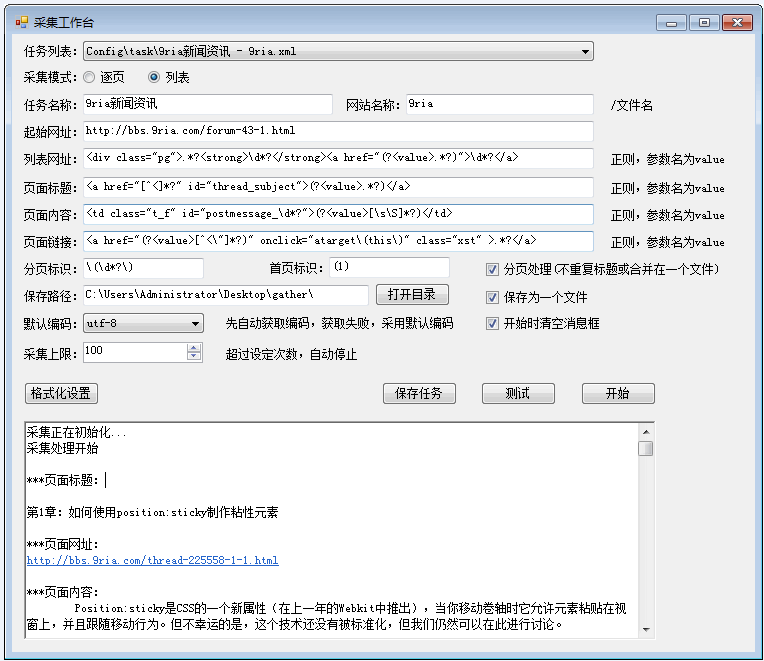
[task]

Config\task\

**config** 设置采集内容格式化的配置保存路径

**task** 设置任务工作规则保存和加载路径

采集工作窗体



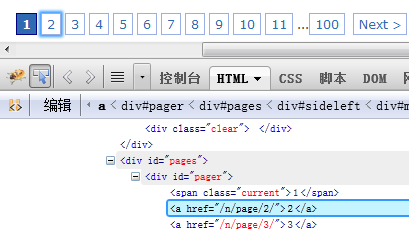
**起始网址**

采集开始的网址，如果是逐页的模式，则是第一页的地址；

如果是列表的模式，则是第一个列表页面的地址。

**列表网址**

匹配下一个列表页面地址的正则表达式。



在逐页模式下，不用填写。

**页面标题**

匹配页面标题的正则表达式

**页面内容**

匹配页面内容的正则表达式

**页面链接**

匹配内容页面链接的正则表达式

在逐页模式，采集到一个内容页面之后，可以匹配到下一个页面链接。

在列表模式，采集到一个列表页面之后，就可以匹配到若干个页面链接。

**分页标识**

识别是否为分页标题的匹配正则表达式

**首页标识**

第一页的标识，比如标题【科技业的员工到底有多年轻 （1）】，那么标识可以是(1)

**保存路径**

采集内容的保存目录

**分页处理**

是指采集的文章资料是进行了分页的，那么程序会根据设定的规则，判断是否是分页章节，如果是，则不重复添加标题。

比如

科技业的员工到底有多年轻 （1）

科技业的员工到底有多年轻 （2）

科技业的员工到底有多年轻 （3）

那么采集过程中，只会写入一个标题【科技业的员工到底有多年轻】

**保存为一个文件**

如果勾选，则采集到的所有内容都写入到一个文件中

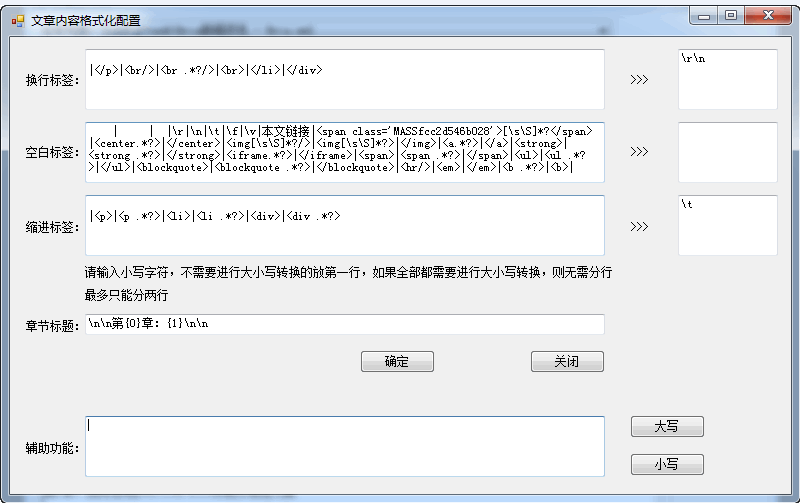
**开始**

开始采集并将内容保存

**测试**

在消息框显示采集的效果

格式化设置窗体



左边是匹配到的字符，后边是表示要替换成的字符。

程序运行时，会将第二行（如果有两行）的字符拷贝一份转换为大写组合在一起，进行格式化。

**换行标签、空白标签、缩进标签**

可以输入包含正则在内的字符进行匹配

**章节标题**

{0}表示采集的序号（采集一个地址则加1），{1}表示采集到的标题。

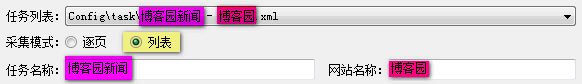
**辅助功能**

可以将输入的字符进行大小写转换

编写新规则

编写采集规则需要有一定的正则表达式的知识，如果不了解阅读这个页面：<http://www.jb51.net/tools/zhengze.html>

任务是以xml文件的形式保存，文件名命名格式是：任务名称 - 网站名称.xml



在任何一个任务状态下，只需要修改任务名称，或者网站名称，再点击保存任务，即可新建一个任务。

如果名称一样会提示是否覆盖。

这里以博客园新闻为例

博客园新闻是一个列表式的采集任务——在一个页面可以匹配得到若干个页面地址

<http://news.cnblogs.com/>

使用firebug或者其它前端调试工具，可以轻松得到采集特征

比如下图



点击红框【**点击查看页面中的元素**】然后在页面的【创业公司如何评估 – 度量公司潜力的方法】位置点下。

就可以定位到html代码

这样就可以获取到内容页面的链接特征

<h2 class="news\_entry">

<a target="\_blank" href="/n/182026/">创业公司如何估值 — 度量公司潜力的方法</a>

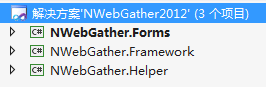
</h2>

然后需要观察这个标识是不是唯一特征的，也就是这个特征匹配到的都是自己期望中的内容。否则就需要增加更多的限制特征。

将特征编写为匹配的正则表达式



源码说明



解决方案有3个项目组成

Forms是视窗程序

Framework是采集程序

Helper是辅助程序

|  |  |
| --- | --- |
|  | 由于考虑到以后会增加不同的采集任务，因此采用MDI窗体。  Config目录是默认配置  FrmFormatConfig是内容格式化配置窗体  FrmGatherWorker是采集工作窗体  MDIParentMain是窗体容器 |
|  | Config是内容格式化配置实体类  Task是采集任务规则实体类  Worker是采集工作类 |
|  |  |

Worker采集工作类说明

先看看3个主要事件

/// <summary>

/// 错误触发事件，传入参数 引发的异常对象、错误的类型、当前工作的网址

/// </summary>

public event Action<Exception, ErrorType, string> OnError;

/// <summary>

/// 工作结束触发事件

/// </summary>

public event Action OnWorkEnd;

/// <summary>

/// 一次/地址采集完成触发事件，传入参数 采集内容的标题、内容、网址

/// </summary>

public event Action<string, string, string> OnWorkItemEnd;

创建对象

Worker work = new Worker(\_httpRequest, \_config, \_task);

work.OnError += w\_OnError;

work.OnWorkItemEnd += work\_OnWorkItemEnd;

work.OnWorkEnd += work\_OnWorkEnd;

/// <summary>

/// 一次(个网址)采集完成后，执行内容写入文件操作

/// </summary>

private void work\_OnWorkItemEnd(string curWebTitle, string curWebContent, string curUrl)

{

//将采集到的内容写入到文件流中

byte[] byteWebContent = Encoding.UTF8.GetBytes(curWebContent);

if (\_task.IsSaveOnlyFile)

{

//如果当前内容标题为空，则可能分页

if (!string.IsNullOrEmpty(curWebTitle))

{

byte[] byteWebTitle = Encoding.UTF8.GetBytes(curWebTitle);

\_curSavaFile.Write(byteWebTitle, 0, byteWebTitle.Length);

}

\_curSavaFile.Write(byteWebContent, 0, byteWebContent.Length);

}

else

{

using (FileStream curSavaFile2 = new FileStream("{0}{1}.txt".FormatWith(\_task.SavePath, curWebTitle), FileMode.OpenOrCreate, FileAccess.ReadWrite))

{

curSavaFile2.Write(byteWebContent, 0, byteWebContent.Length);

}

}

UpdateWorkMessage("\n已采集：{0}，网址：{1}".FormatWith(curWebTitle, curUrl));

Application.DoEvents();

}

其它更多，请下载源码查看

其它

运行程序下载：<http://files.cnblogs.com/yelaiju/NWebGather.rar>

源码下载请到开源地址下载

开源地址：<https://github.com/alifellod/NWebGather>

希望各位朋友对采集有兴趣的一起维护和贡献代码，如此大家都可以轻松的共享同一个采集框架。

QQ群：9524888

欢迎大家入群交流共享采集任务规则，讨论技术，讨论人生……