

【crifan推荐】轻量级文本编辑器，Notepad最佳替代品：Notepad++

版本：v3.0

Crifan Li

摘要

本文主要介绍了笔者crifan推荐的轻量级文本编辑器，Notepad最佳替代品：Notepad++，内容主要包含notepad，notepad2，notepad++，ultraEdit的比较，使用Notepad++前要了解的知识，Notepad++的插件，Notepad++的安装过程，Notepad++的各种常见和不常见的功能的详解



本文提供多种格式供：

在线阅读	HTML ¹	HTMLs ²	PDF ³	CHM ⁴	TXT ⁵	RTF ⁶	WEBHELP ⁷
下载（7zip压缩包）	HTML ⁸	HTMLs ⁹	PDF ¹⁰	CHM ¹¹	TXT ¹²	RTF ¹³	WEBHELP ¹⁴

HTML版本的在线地址为：

http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/html/rec_soft_npp.html

有任何意见，建议，提交bug等，都欢迎去讨论组发帖讨论：

http://www.crifan.com/bbs/categories/rec_soft_npp/

修订历史

修订 3.0

2013-12-15

crl

1. 添加了3个正则表达式替换的例子：给sect2的title不变前提下添加xml:id
2. 添加了3个正则表达式替换的例子：将antlr的token中的id的引号去掉变成token
3. 添加了3个正则表达式替换的例子：将原先的单引号括起来的ID变成antlr中的该token的定义

4. 如果乱码则切换到所推测的编码直至消除乱码

¹ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/html/rec_soft_npp.html

² http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/htmls/index.html

³ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/pdf/rec_soft_npp.pdf

⁴ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/chm/rec_soft_npp.chm

⁵ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/txt/rec_soft_npp.txt

⁶ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/rtf/rec_soft_npp.rtf

⁷ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/webhelp/index.html

⁸ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/html/rec_soft_npp.html.7z

⁹ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/htmls/index.html.7z

¹⁰ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/pdf/rec_soft_npp.pdf.7z

¹¹ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/chm/rec_soft_npp.chm.7z

¹² http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/txt/rec_soft_npp.txt.7z

¹³ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/rtf/rec_soft_npp.rtf.7z

¹⁴ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/webhelp/rec_soft_npp.webhelp.7z

5. 修复了一些笔误
6. 把正则的查找和替换，单独整理成子xml，同时把图片也移至独立的文件夹。
7. 把列模式单独整理成子xml
8. 增加正则替换的例子：dd宏定义中去除多国语言字符串，C宏定义转java变量定义
9. 增加正则替换的例子：docbook的link变bibliomixed

修订 2.1	2013-09-04	crl
--------	------------	-----

1. 添加了用Notepad++选用合适的编码打开对应的文件
2. 添加了XBracket Lite插件的内容
3. 添加了2个正则表达式替换的例子
4. 添加了1个正则表达式查找的例子
5. 添加了文字右对齐
6. 添加了格式化XML
7. 添加了单词的自动完成
8. 添加了1个正则表达式替换的例子
9. 添加了1个正则表达式替换的例子：给sect2添加xml:id和title
10. 添加了Ctrl+D拷贝单行或多行

修订 1.0	2012-08-03	crl
--------	------------	-----

1. 把npp从crifan_rec_soft中提取称为单独的book

【crifan推荐】轻量级文本编辑器，Notepad最佳替代品：Notepad

++:

Crifan Li

版本：v3.0

出版日期 2013-12-15

版权 © 2013 Crifan, <http://crifan.com>

本文章遵从：[署名-非商业性使用 2.5 中国大陆\(CC BY-NC 2.5\)](#)¹⁵

¹⁵ http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#cc_by_nc

目录

正文之前	x
1. 目的	x
2. 原帖	x
3. 为何推荐Notepad++	x
3.1. Notepad++的一些基本特点	x
3.2. notepad, notepad2, notepad++ , ultraEdit比较	x
4. 使用Notepad++前要了解的知识	xi
4.1. Notepad++的名称和缩写	xi
4.2. Notepad++修改设置后, 立即生效	xi
4.3. Notepad++的版本: ANSI和Unicode	xi
5. 声明	xii
1. Notepad++的插件	1
1.1. Notepad++插件的存在形式与位置	1
1.1.1. Notepad++中被禁用了的插件	1
1.1.1.1. 如何恢复被禁用的插件	2
1.2. Notepad++默认已经安装了的一些插件	2
1.3. Notepad++插件的下载与安装	2
1.3.1. 去哪里下载Notepad++的插件	2
1.3.2. Notepad++插件的安装方法	3
1.3.2.1. 手动安装插件	3
1.3.2.2. 通过Plugin Manager安装插件	4
1.4. Notepad++中常用的插件	4
1.4.1. 插件管理器: Plugin Manager	4
1.4.1.1. 打开Plugin Manager	4
1.4.1.2. 通过Plugin Mangager安装插件	6
1.4.2. 16进制编辑器: HEX-Editor	8
1.4.2.1. 以16进制模式查看当前文件	8
1.4.3. 导出彩色代码为其他(word, HTML)格式的文件: NppExport	10
1.4.3.1. 将当前(已着色/代码高亮/语法高亮的)彩色代码导出到Word文件或HTML网页中	10
1.4.4. 轻量级资源管理器: Light Explorer	16
1.4.4.1. 通过Light Explorer浏览文件, 打开文件	16
1.4.5. 括号自动补全插件: XBracket Lite	17
1.4.5.1. 通过XBracket Lite实现括号的自动补全	18
1.4.6. 支持Notepad++中使用js脚本实现功能扩展: jN	21
1.4.7. 显示函数列表: Function List	21
1.4.8. js代码压缩,js代码格式化,JSON查看器: JSMinNpp	22
2. Notepad++的安装过程	25
2.1. Notepad++安装: 选择语言	25
2.2. Notepad++安装: 安装向导	27
2.3. Notepad++安装: 许可证协议	27
2.4. Notepad++安装: 选择安装位置	28
2.5. Notepad++安装: 选择组件	29
2.6. Notepad++安装: 选择组件(参数)	41
2.7. Notepad++安装: 安装完成	43
3. Notepad++的功能详解	44
3.1. Notepad++支持右击打开所选文件	44
3.2. Notepad++的语法高亮	45
3.2.1. 语法高亮的实现逻辑	45
3.2.2. 实现未知类型(不支持的后缀名)的文件的语法高亮	46
3.2.2.1. 通过手动设置语言, 实现未知类型文件的语法高亮	46
3.2.2.2. 通过添加文件后缀名, 实现未知类型文件的语法高亮	48
3.2.3. 自定义语法高亮的颜色	50
3.3. Notepad++的多种编码支持	50
3.3.1. 用Notepad++查看当前文件编码	50

【crifan推荐】轻量级文
本编辑器，Notepad最
佳替代品：Notepad++

3.3.2. 用Notepad++选用合适的编码打开对应的文件	52
3.3.3. 如果乱码则切换到所推测的正确的编码直至消除乱码	54
3.3.4. 用Notepad++实现不同字符编码之间的转换	56
3.3.5. 设置Notepad++新建文件（默认）的编码类型	60
3.4. Notepad++的正则表达式替换和替换	62
3.5. Notepad++的计数功能	90
3.6. Notepad++的列编辑功能	91
3.6.1. 什么是列编辑模式	91
3.6.2. Notepad++的列编辑模式的基本操作	91
3.6.3. Notepad++的列编辑模式的高级用法	95
3.7. Notepad++的代码折叠	118
3.8. Notepad++的自动完成	120
3.8.1. Notepad++的单词自动完成	123
3.9. Notepad++的默认HTML查看器	125
3.10. Notepad++支持显示回车符，换行符，TAB键，行首，行尾等特殊字符	128
3.11. Notepad++支持插入特殊字符	131
3.12. Notepad++高亮所选关键字	134
3.13. Notepad++中Windows,Unix,Mac三种格式之间的转换	134
3.13.1. 查看当前文件格式（换行符用的是什么字符）	134
3.13.2. 从Windows转换为Unix/Linux格式	135
3.13.3. 从Unix/Linux转换为Mac格式	136
3.13.4. 从Mac格式转换为Windows格式	137
3.14. Notepad++高亮匹配的键对	138
3.15. Notepad++文件自动更新	139
3.16. Notepad++设置用空格取代TAB键	141
3.17. Notepad++给代码单行/批量，添加/取消注释	141
3.18. Notepad++的多主题	143
3.18.1. Notepad++主题: Default	144
3.18.2. Notepad++主题: Bepin	145
3.18.3. Notepad++主题: Black Board	145
3.18.4. Notepad++主题: Choco	146
3.18.5. Notepad++主题: Zenburn	146
3.18.6. Notepad++主题: Deep Black	147
3.18.7. Notepad++主题: Hello Kitty	147
3.18.8. Notepad++主题: Mono Industrial	148
3.18.9. Notepad++主题: Monokai	148
3.18.10. Notepad++主题: Obsidian	149
3.18.11. Notepad++主题: Plastic Code Wrap	149
3.18.12. Notepad++主题: Ruby Blue	150
3.18.13. Notepad++主题: Solarized-Light	150
3.18.14. Notepad++主题: Solarized	151
3.18.15. Notepad++主题: Twilight	151
3.18.16. Notepad++主题: Vibrant Ink	152
3.18.17. Notepad++主题: Vim Dark Blue	152
3.19. Notepad++的一些小功能	153
3.19.1. 支持双击Tab标签关闭文件	153
3.19.2. 显示文件行号	154
3.19.3. 支持是否打印行号	155
3.19.4. 编辑文件后，是否备份	157
3.19.5. 调用搜索引擎搜索选定文字	158
3.19.5.1. 更改google搜索功能的快捷键	159
3.19.6. 暂时隐藏某些行	161
3.19.7. 支持鼠标滚轮控制缩放	163
3.19.8. 全屏显示	163
3.19.9. Ctrl+Tab实现在多个打开的窗口间切换	163
3.19.10. 支持设置是否显示列边界（右边线）	164
3.19.11. 文字显示：右对齐和左对齐	165
3.19.11.1. 如何设置文字右对齐	165

【crifan推荐】轻量级文
本编辑器，Notepad最
佳替代品：Notepad++

3.19.11.2. 文字右对齐的效果	166
3.19.12. 格式化（成漂亮的输出效果）	168
3.19.12.1. 用Notepad++格式化XML	168
3.19.13. Ctrl+D拷贝单行或多行	169
3.19.13.1. 直接复制光标所在的当前行的内容	170
3.19.13.2. 复制选中的部分文字，多行内容等内容	170
参考书目	172

插图清单

2.1. Notepad++的右击打开文件功能	31
2.2. Notepad++默认已安装了一些插件	34
3.1. XML文件中的Notepad++的代码折腾功能	119

表格清单

1. Notepad,Notepad2,Notepad++,UltraEdit之间的对比	x
--	---

范例清单

3.1. 在Notepad++中使用ISO-8859-1编码打开VirtualBox的Manual的HTML源码	52
3.2. 出现乱码, 猜测出是西欧编码, 切换到ISO 8859-1而消除乱码	54
3.3. Notepad++正则表达式替换举例: 一次性替换多个文件的后缀	62
3.4. Notepad++正则表达式替换举例: 一次性替换多个路径	65
3.5. Notepad++正则表达式替换举例: 一次性替换多个listitem为sect4	66
3.6. Notepad++正则表达式替换举例: 给每一行都添加AddIcon的前缀	69
3.7. Notepad++正则表达式替换举例: 给book的标题和地址添加html代码	71
3.8. Notepad++正则表达式替换举例: 查找单个的CR或LF	74
3.9. Notepad++正则表达式替换举例: 去除href链接	76
3.10. Notepad++正则表达式替换举例: 把标题和地址转换为link格式	77
3.11. Notepad++正则表达式替换举例: 给关键字添加双引号, 把逗号变成竖杠	78
3.12. Notepad++正则表达式替换举例: wlw中图片分行	78
3.13. Notepad++正则表达式替换举例: 给sect2添加xml:id和title	80
3.14. Notepad++正则表达式替换举例: 保持sect2和title添加xml:id	81
3.15. Notepad++正则表达式替换举例: 去除单引号变成antlr的token	82
3.16. Notepad++正则表达式替换举例: 将单引号加ID变成antlr的token的定义	83
3.17. Notepad++正则表达式替换举例: dd宏定义中去除多国语言字符串	85
3.18. Notepad++正则表达式替换举例: C宏定义转java变量定义	86
3.19. Notepad++正则表达式替换举例: docbook的link变bibliomixed	88
3.20. 列编辑: 多行输入同样内容	91
3.21. 列编辑: 删除多行内容	91
3.22. 列编辑: 多行中插入相同或不同的内容	92
3.23. 列编辑: 同时复制和粘贴多列	96
3.24. Notepad++可以查看特殊字符的用途举例	130
3.25. Notepad++可以插入特殊字符的用途举例	131

正文之前

1. 目的

本文旨在推荐我个人 (crifan) 觉得好用的，轻量级文本编辑器，Notepad最佳替代品：Notepad++

2. 原帖

此文所对应的，原先的旧帖为：[\[1\]](#)

3. 为何推荐Notepad++

因为Notepad++的确很好用，有很多优点，值得推荐给不熟悉的人。

3.1. Notepad++的一些基本特点

关于Notepad++，先用几句话概述一下其特点：

- Notepad++是一个**文本编辑工具**，而且是**免费**的。
- Notepad++旨在**替代Windows默认的Notepad**而生，比notepad的功能强大很多很多。
- Notepad++比UltraEdit更加**轻量级**，比Notepad2等更强大好用。
- Notepad++本身的功能已经非常强大，且又支持**插件**，可以扩展更多的功能。并且已经有很多很有用和好用的插件可供选择使用。

3.2. notepad，notepad2，notepad++，ultraEdit比较

在详细介绍Notepad++之前，先来解释一下，为何要选择Notepad++，即把常见的一些文本编辑器和Notepad++比较，看看其有啥点好：

常见的文本编辑器有很多，此处，只提及Notepad，Notepad2，Notepad++和UltraEdit。

对于这些文本编辑器，之前或多或少都曾用过，但是随着使用经验的的增长，越加发现，Notepad++是最好用的。

目前，在Windows环境下，偶已完全抛弃除了Notepad++之外的其他文本编辑器了。

下面，就对这几种常见的文本编辑器，做个大概比较：

表 1. Notepad,Notepad2,Notepad++,UltraEdit之间的对比

文本编辑器	是否免费	功能强弱	易用性 ^①	资源消耗	启动和编辑速度 ^②	其他说明
Notepad	免费	非常弱	易用	很少	快	Windows自带，免安装
Notepad2	免费	强	易用	一般	快	
Notepad++	免费	非常强	易用	稍多	快	
UltraEdit	收费	非常强	难用	非常多	慢	

① 此处的易用性，主要指的是，有很多细节的功能上，虽然都支持某功能，但是是否方便用户使用。

比如，列编辑的功能，UltraEdit和Notepad++都支持，但是[第 3.6 节 “Notepad++的列编辑功能”](#)支持直接通过Alt+鼠标，就可以选择多列并拷贝粘贴了，很是方便使用。

- ② 此处主要是，对于大一点的文件，在编辑的时候，是否很快。

因此，根据启动速度，占用资源，功能支持等方面来综合衡量，对于notepad，notepad2，notepad++，ultraEdit来说，无疑notepad++是最好的选择了。

4. 使用Notepad++前要了解的知识

安装和使用Notepad++之前，需要了解一些相关的基本知识：

4.1. Notepad++的名称和缩写

Notepad++，对应的英文全称为Notepad Plus Plus，一般写为NotepadPlusPlus，常缩写为npp

4.2. Notepad++修改设置后，立即生效

Notepad++中，改变某配置的值之后，（不像其他软件，需要点击“启用”或“确定”，才能看到效果，而是），立即生效，即可可以看到效果。

最明显的一个例子就是，当改变了Notepad++的主题后，即可可以看到当前主题的效果。详参：[第 3.18 节 “Notepad++的多主题”](#)

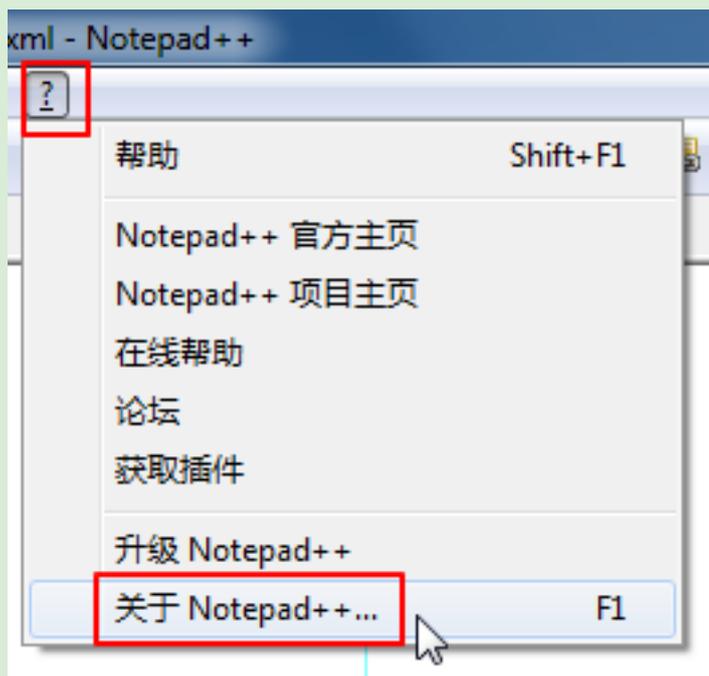
4.3. Notepad++的版本：ANSI和Unicode

Notepad++有两个版本，一个是ANSI版本，一个是UNICODE版本。

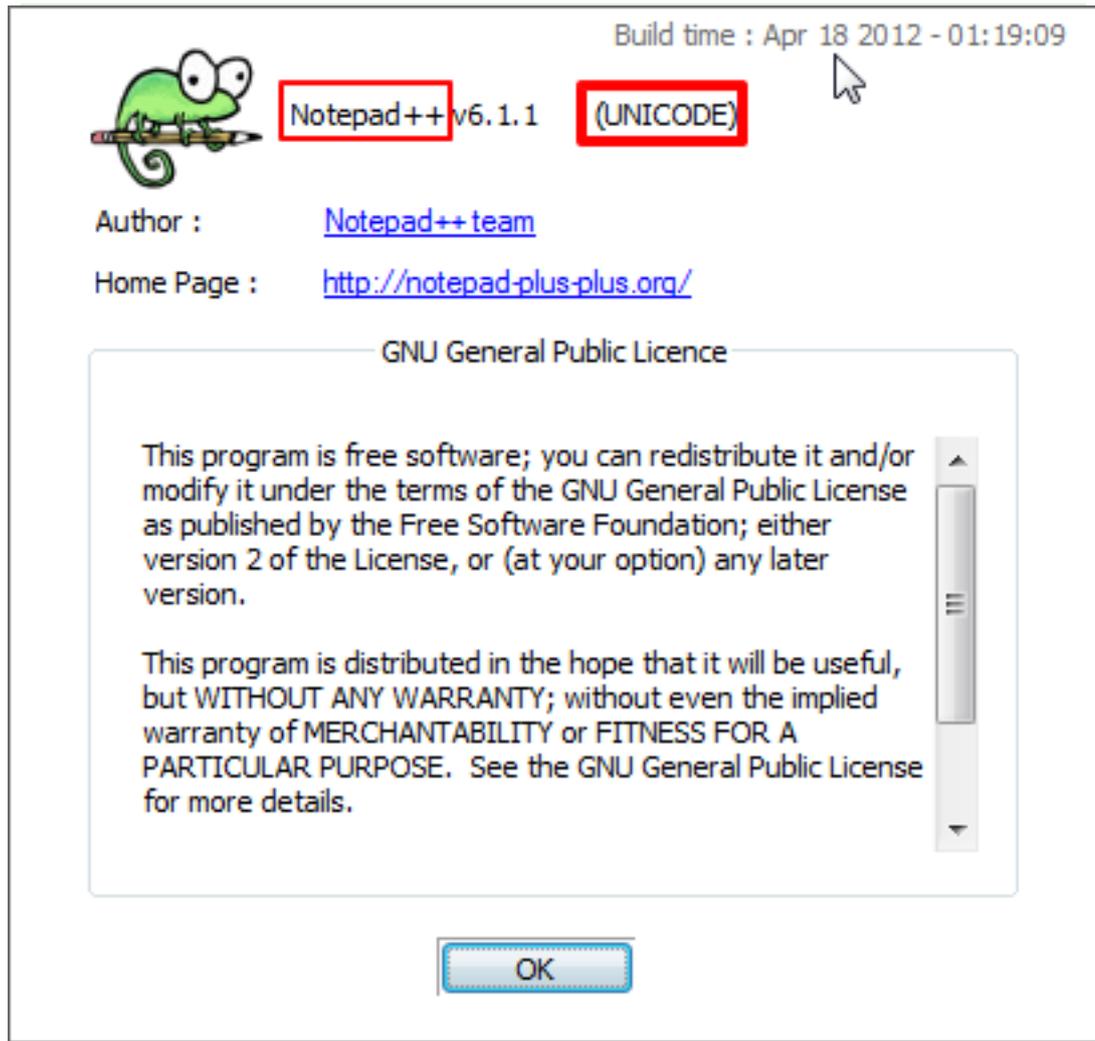


如何查看Notepad++的版本

Notepad++中，点击：? → 关于Notepad++...F1



即可看到对应的标有Unicode，或者ANSI的字样，我这里的是Unicode的：



对于大多数人把Notepad++作为notepad的增强版，而需要的普通应用的话，ANSI版本和UNICODE版本，其实没啥差别。

不过，就像[2]中所说的，如果不是非得用ANSI版本才支持的极个别插件的话，那么还是推荐大家使用Unicode版本的，其支持几乎所有的字符。



Unicode几乎包括了世界上的所有字符

如题，Unicode就是为了统一世界上如此多的字符而出现的，所以推荐使用Unicode版本的Notepad++。

关于Unicode更多的解释，请参看：[3]

结论：不用管太多，记住用Unicode版本的Notepad++就行了。

5. 声明

任何疑问，意见，建议等，都欢迎一起探讨：admin (at) crifan.com。

第 1 章 Notepad++ 的插件

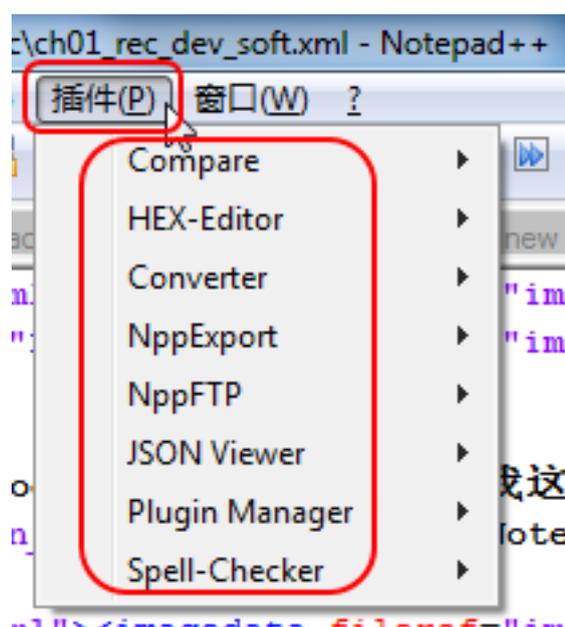
插件，意味着功能的扩展。

所以当你需要某些功能，而Notepad++本身没有提供此功能，此时就可以考虑插件了。

Notepad++不仅支持通过插件扩展已有功能，更主要的是目前已经有非常多的插件可供选择了。

而且很多功能都是很实用和很方便使用的。

你可以通过点击工具栏中的 插件(P)，而找到目前已经安装了哪些可用的插件：



下面对Notepad++的插件的各个方面，进行详细介绍：

1.1. Notepad++ 插件的存在形式与位置

Notepad++的插件是dll文件的形式存在的

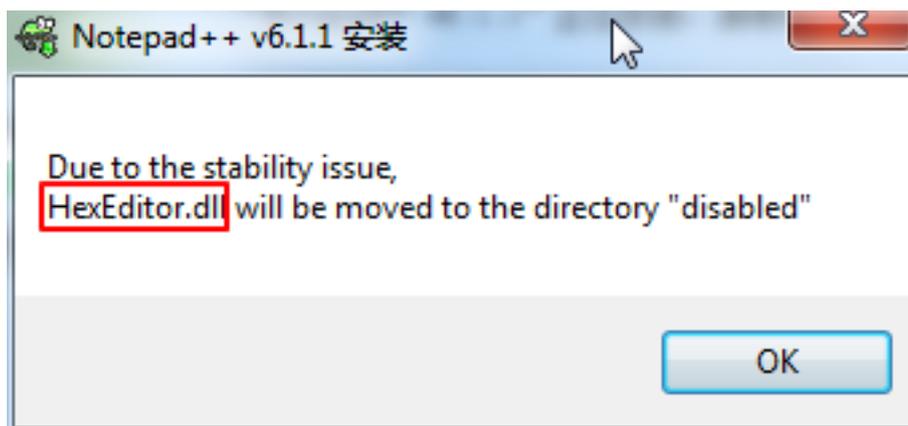
比如16进制编辑器，HEX-Editor，对应的dll文件是：HexEditor.dll

插件是存放在安装目录下的plugins目录中，比如D:\Program Files (x86)\Notepad++\plugins

1.1.1. Notepad++ 中被禁用了的插件

某些插件，由于某些原因，比如兼容性不够好的话，会在重新安装Notepad++的过程中，被Notepad++直接禁用掉。

比如，我最喜欢的其中一个插件，叫做HEX-Editor，由于兼容性不好而在安装过程中被禁用：



被禁用的插件，其dll文件会被移至到disabled目录中，比如D:\Program Files (x86)\Notepad++\plugins\disabled

1.1.1.1. 如何恢复被禁用的插件

有时候，我们需要恢复被禁用的插件。

比如上述HEX-Editor，对于我来说，用了几年了，其实都没遇到啥兼容性问题，所以（在安装完毕后）需要恢复（继续使用）该插件。

而恢复该插件的办法，其实很简单，就是到disabled中，找到对应的dll文件，然后放回plugins目录中即可。

比如此处的例子，就是把HexEditor.dll从D:\Program Files (x86)\Notepad++\plugins\disabled移动到D:\Program Files (x86)\Notepad++\plugins即可。

1.2. Notepad++ 默认已经安装了的一些插件

如图 2.2 “Notepad++默认已安装了一些插件” 中所介绍，在安装Notepad++的过程中，默认已经安装了一些常用插件，这些插件的目前包括：

- Spell Checker
- NPP FTP
- NppExport: [第 1.4.3 节 “导出彩色代码为其他 \(word , HTML \) 格式的文件: NppExport”](#)
- Plugin Manager: [第 1.4.1 节 “插件管理器: Plugin Manager”](#)
- Converter

关于各插件的功能的详细介绍，请参考[第 1.4 节 “Notepad++中常用的插件”](#)

1.3. Notepad++ 插件的下载与安装

1.3.1. 去哪里下载Notepad++的插件

目前已知这些地方可以找到Notepad++的插件：

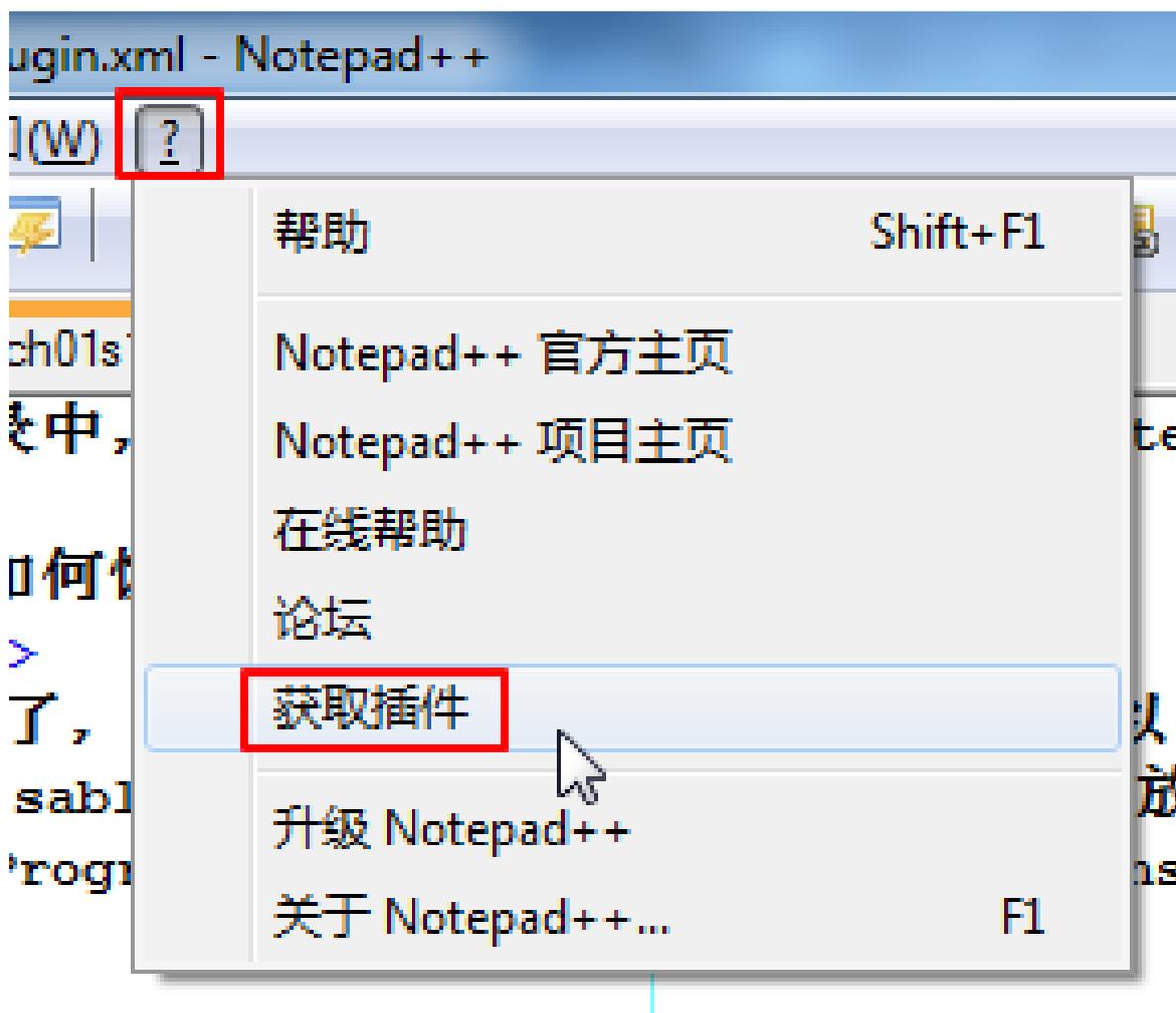
- [Notepad++的SourceForge的Plugins主页](#)¹

里面可以直接下载对应的插件。不过需要注意的是，该页面中的插件数量，相对不多。

- ? ⇒ 获取插件

¹ <http://sourceforge.net/projects/npp-plugins/files/>

在Notepad++软件中，点击菜单栏左右边的那个问号，然后点击获取插件：



就可以自动帮你调用浏览器打开页面:[\[4\]](#)

该页面中，包含了对于每个插件的简单介绍，以及其所适用于的版本，是ANSI还是Unicode的。

对于想要找一些功能，但是连插件名都不知道的情况下，很适合去那里找找。

如同第 4.3 节 “Notepad++的版本：ANSI和Unicode” 中的介绍，Notepad++插件也是分ANSI和Unicode版本的，所以，你下载插件的时候，记得要下载和你当前Notepad++版本所匹配的插件。

比如，[\[5\]](#)版本的[\[6\]](#)中就有：HexEditor_0_9_5_UNI_dll.zip和HexEditor_0_9_5_ANSI_dll.zip，所以，如果你用的是前面所推荐的Unicode版本的话，那么所需要下载的就是HexEditor_0_9_5_UNI_dll.zip。

1.3.2. Notepad++插件的安装方法

Notepad++的插件有多种安装方法：

1.3.2.1. 手动安装插件

参考第 1.3.1 节 “去哪里下载Notepad++的插件” 中的描述，去[最全的Notepad++插件的官方主页](#)²中，找到自己想要的插件。该页面中同时也提供了下载地址。

² http://sourceforge.net/apps/mediawiki/notepad-plus/index.php?title=Plugin_Central

把下载到的插件的dll文件，放到对应的安装目录下的plugins目录中即可。

个别的插件，还需要安装其他一些配置文件，比如一些xml配置文件等。具体如何操作，下载下来的插件压缩包中，都会有对应的提示的，按照提示安装即可。

1.3.2.2. 通过Plugin Manager安装插件

关于如何通过使用Plugin Manager去安装插件，请参看[第 1.4.1 节 “插件管理器: Plugin Manager”](#)



插件安装完毕后，最好重启Notepad++

Notepad++插件安装完毕后，需要重启Notepad++，这样才能正确识别并可用。

1.4. Notepad++中常用的插件

1.4.1. 插件管理器: Plugin Manager

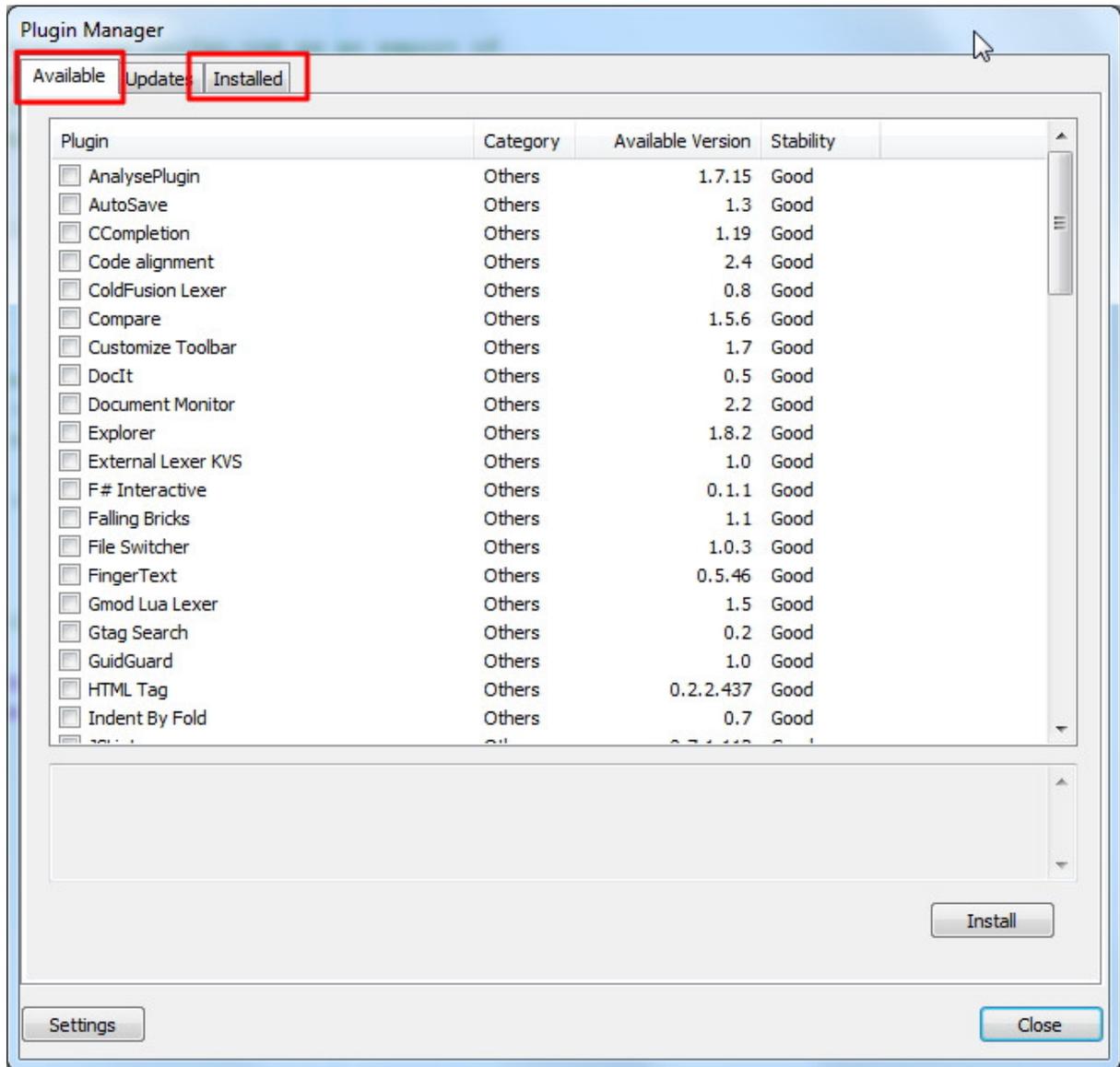
插件功能：此插件可以帮你管理插件，包括查看当前已经安装的插件有哪些，以及自动帮你下载相应的插件。

插件用途：主要用于管理（安装和卸载）插件

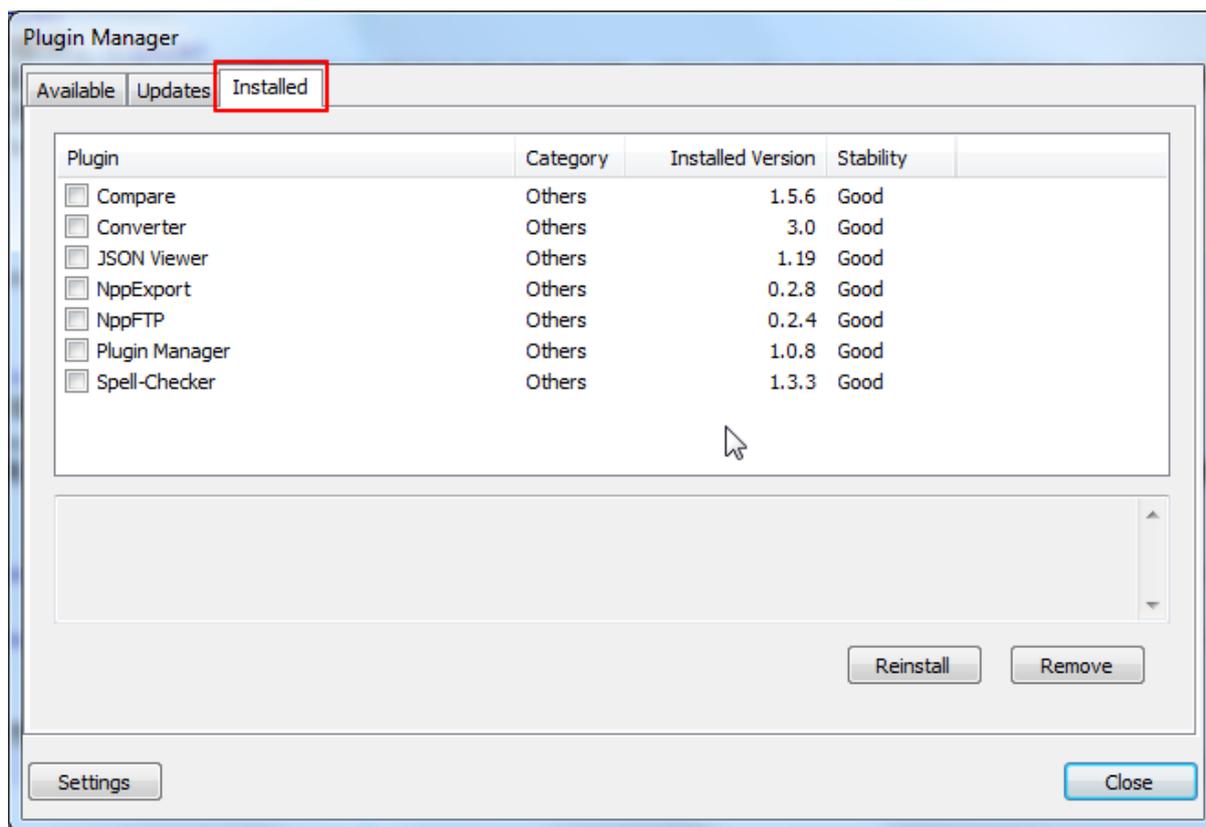
插件安装：在安装过程中，默认已选择安装此插件：[图 2.2 “Notepad++默认已安装了一些插件”](#)

1.4.1.1. 打开Plugin Manager

插件(P) → Plugin Manager → Show Plugin Manager → Available，一栏显示当前可用的插件列表，选中你要的插件，然后点击下面的Install即可自动下载和安装对应插件，很是方便。



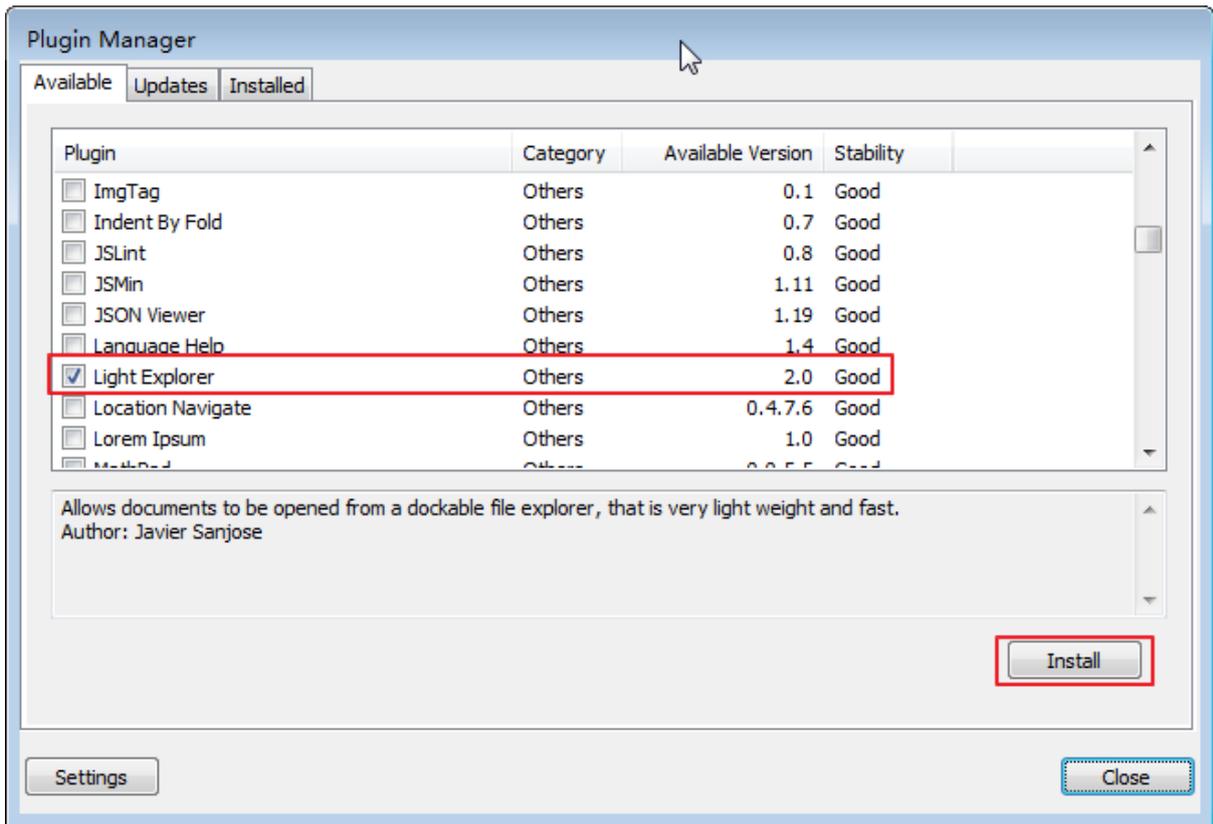
其中，在Intalled一栏可以看到当前已经安装了的插件有哪些：



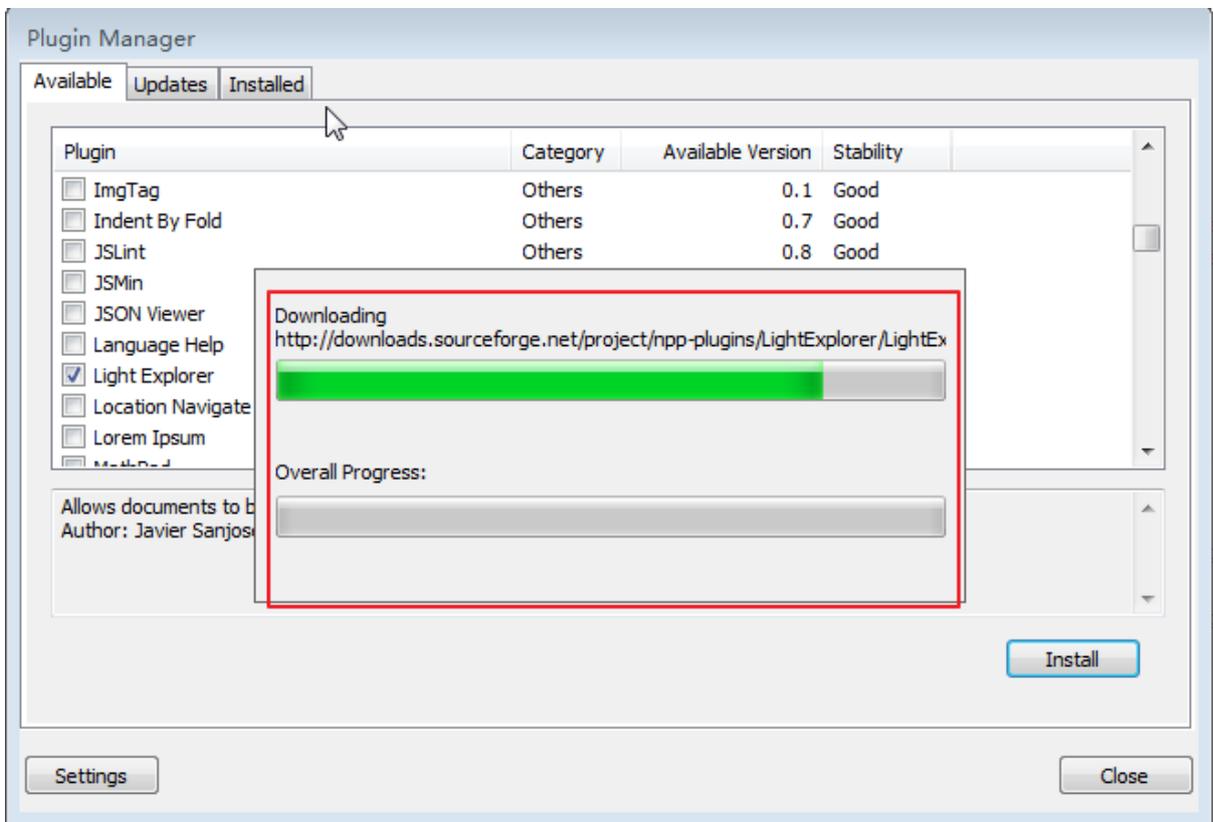
1.4.1.2. 通过Plugin Manager安装插件

下面以Light Explorer为例，说明如何通过Plugin Manager来安装插件：

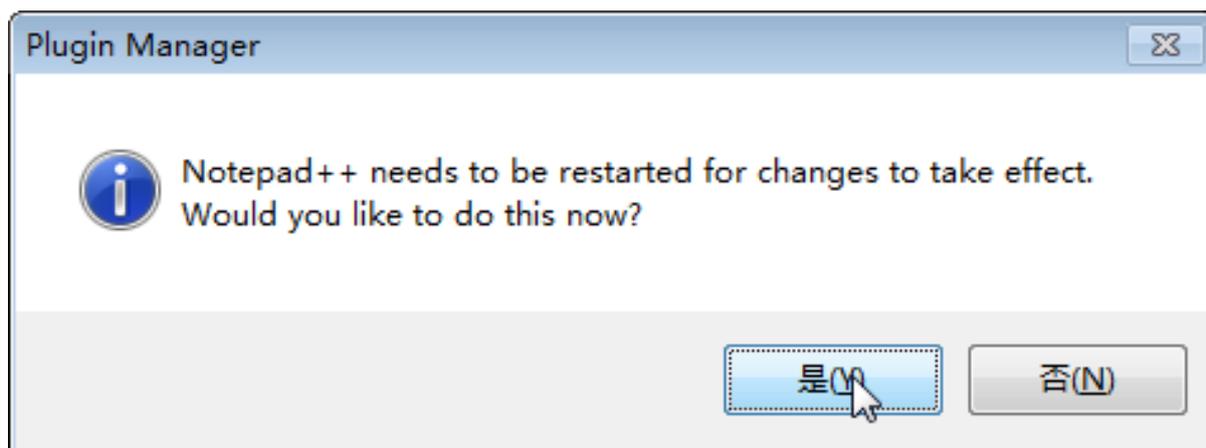
通过[第 1.4.1.1 节 “打开Plugin Manager”](#) 打开Plugin Manager后，找到并选中Light Explorer，然后点击Install：



Plugin Manager就会帮你自动下载对应的插件并安装：

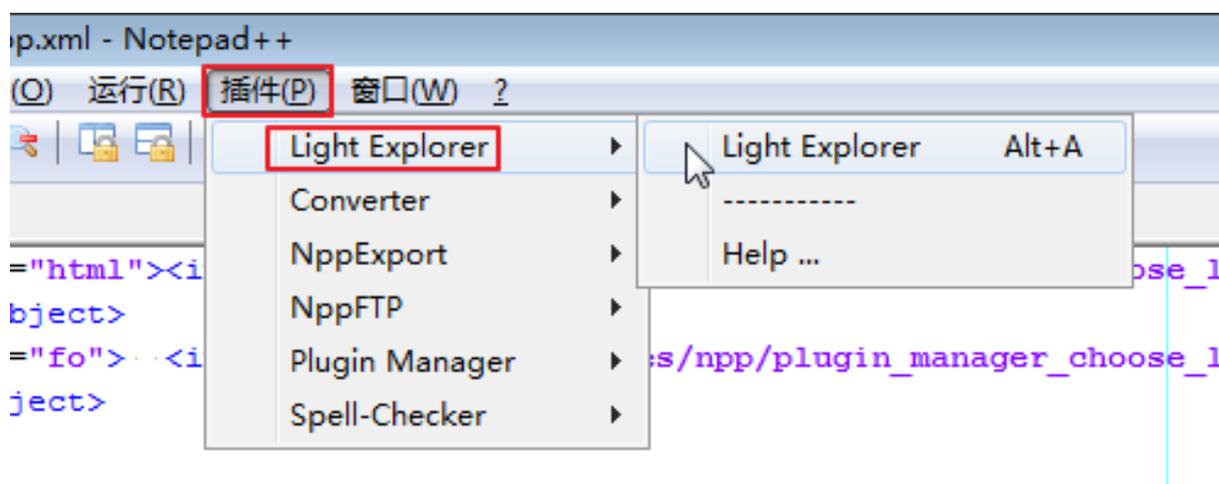


安装完毕后，会提示你是否重启：



选择“是”后，Notepad++会自动重启，并且会自动打开关闭前已打开的这些文件。

此时，就可以看到已经安装的插件了：



1.4.2. 16进制编辑器: HEX-Editor

插件功能：此插件主要提供了16进制查看与编辑的功能。

插件用途：以16进制模式查看和编辑文件。

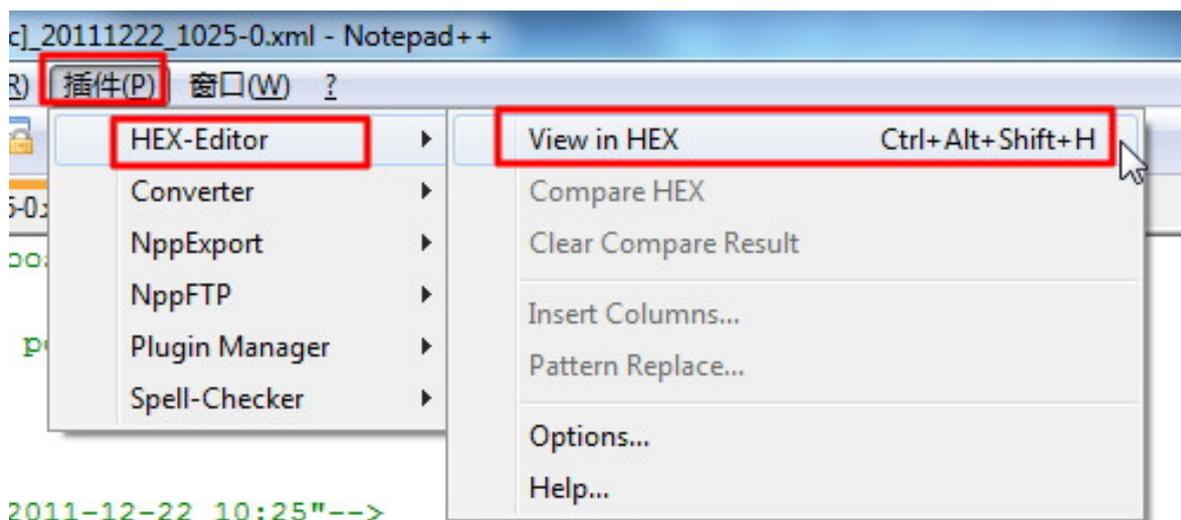
插件安装：参考[第 1.3 节 “Notepad++插件的下载与安装”](#) 去安装Hex Editor插件

HEX-Editor功能详解：

1.4.2.1. 以16进制模式查看当前文件

有几种方法实现以16进制模式查看当前文件：

- 插件(P) ⇒ HEX-Editor ⇒ View in HEX Ctrl+Alt+Shift+H



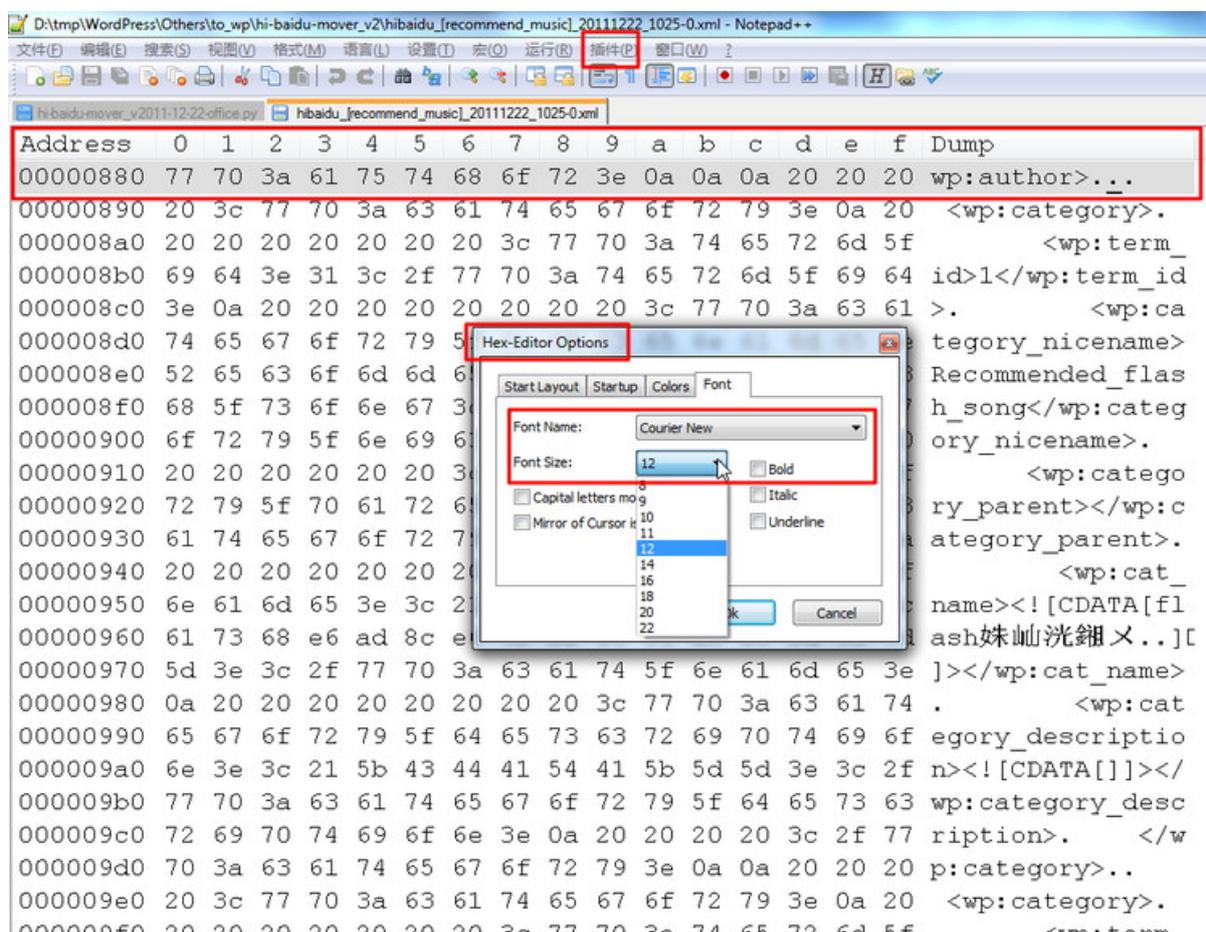
- 点击工具栏上HEX Editor的图标
- 直接使用快捷键：Ctrl+Alt+Shift+H



提示

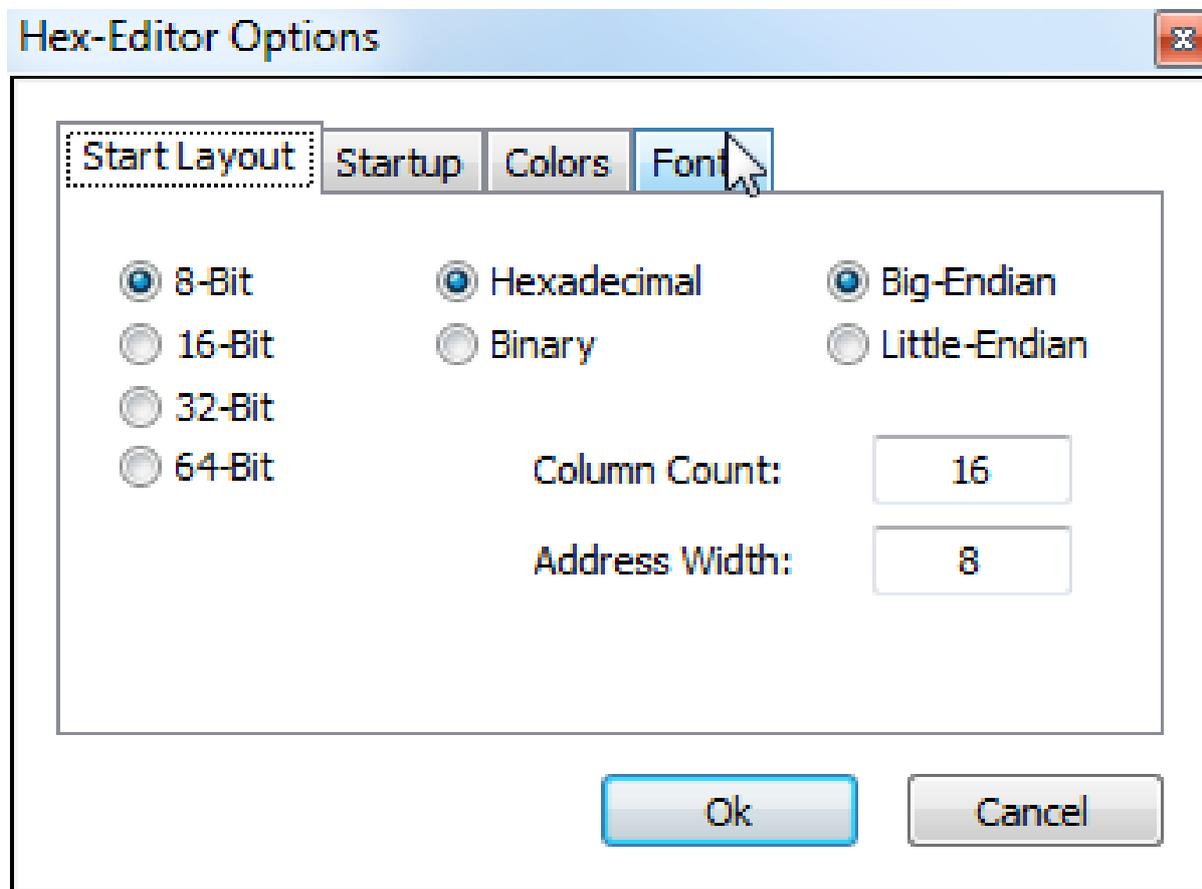
退出16进制模式，和进入16进制模式的方法相同。

然后就可以以16进制模式，查看当前文件了：



接着，就可以去查看和编辑对应的16进制的值。

关于更多的选项，比如想要设置其他字体类型和字体大小，可以去上述的HEX-Editor下面的Options中，根据自己的需要设置：



1.4.3. 导出彩色代码为其他（word，HTML）格式的文件： NppExport

插件功能：导出已着色代码为其他格式的文件

插件用途：将彩色代码，导出为word文档（RTF）或网页（HTML）文件，或者将彩色代码（RTF格式或HTML格式）拷贝到剪贴板，粘贴到别的（word文档，HTML网页）中去。

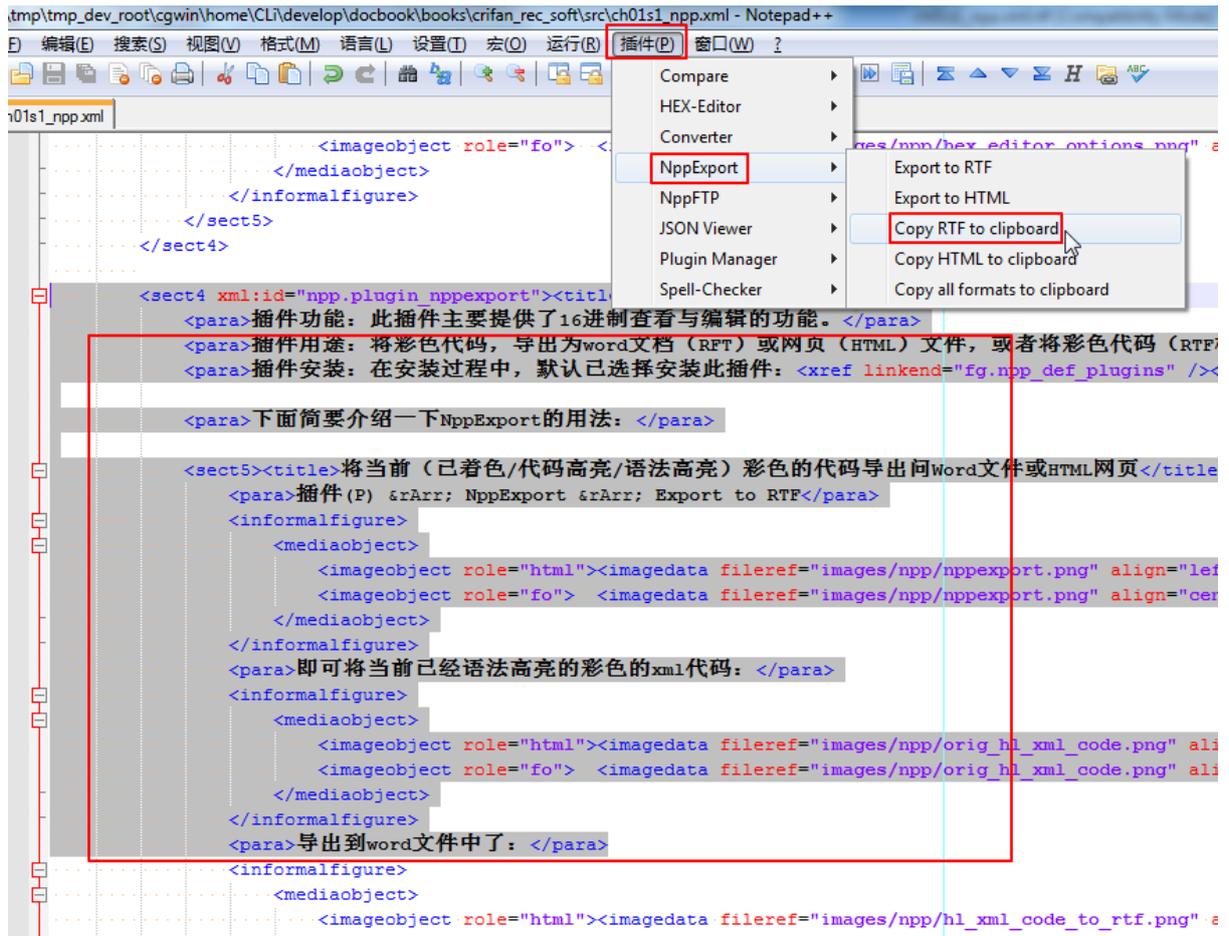
插件安装：在安装过程中，默认已选择安装此插件：[图 2.2 “Notepad++默认已安装了一些插件”](#)

NppExport功能详解：

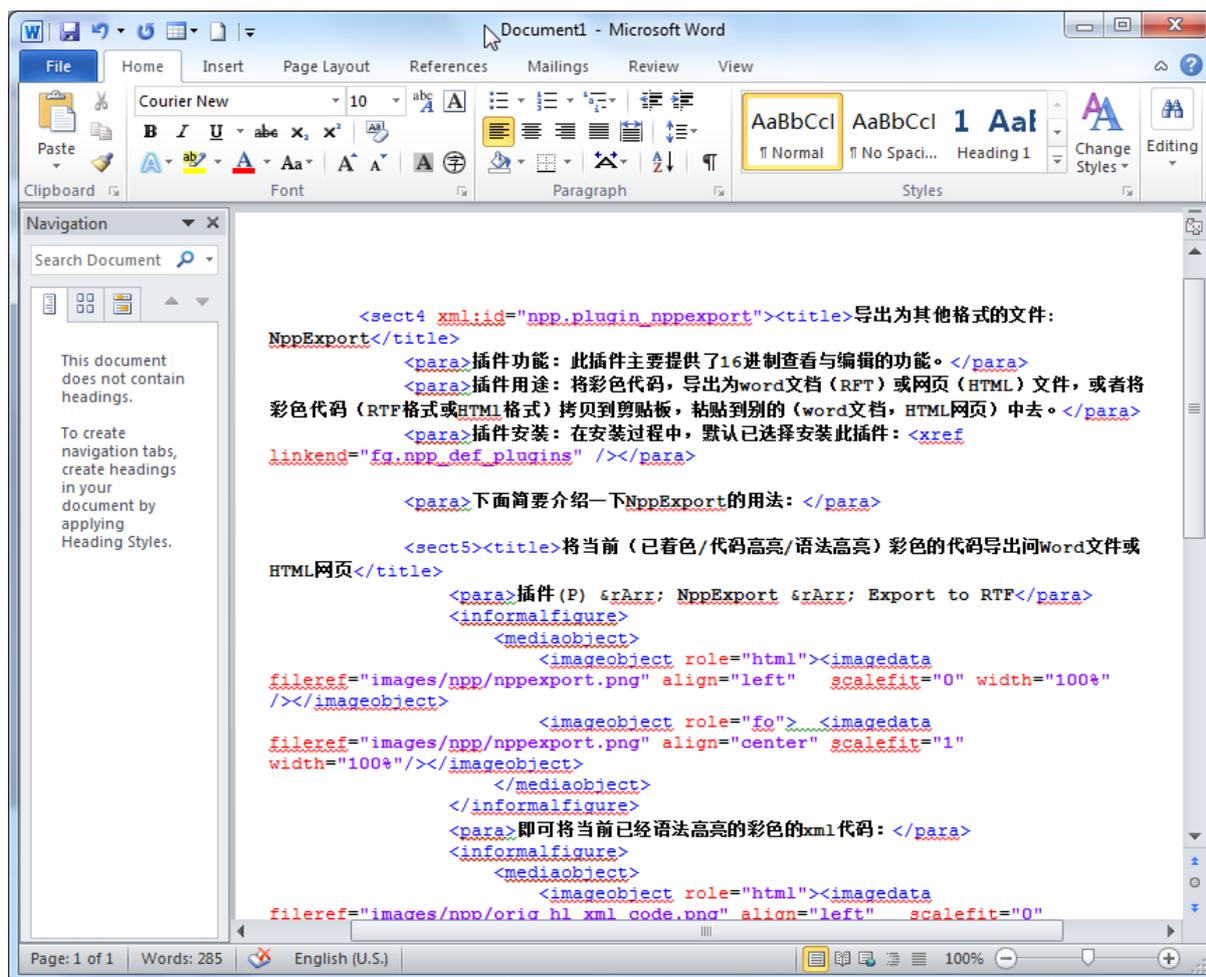
1.4.3.1. 将当前（已着色/代码高亮/语法高亮的）彩色代码导出到 Word文件或HTML网页中

对于选中的代码，将其对应的RTF格式的内容，复制到剪贴板：

Notepad++的插件



然后粘贴到新建的word文档中：



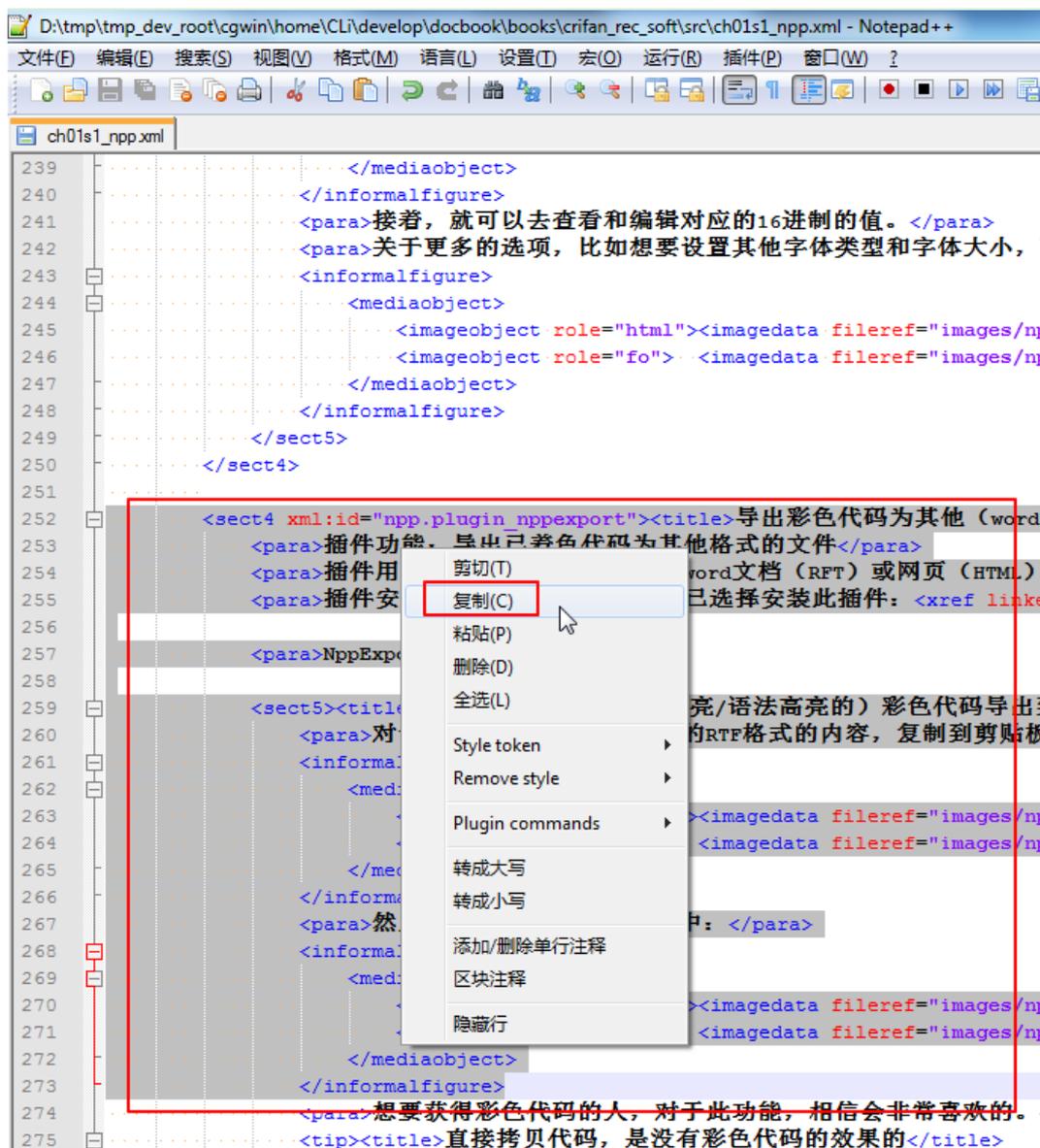
想要获得彩色代码的人, 对于此功能, 相信会非常喜欢的。



直接拷贝代码, 是没有彩色代码的效果的

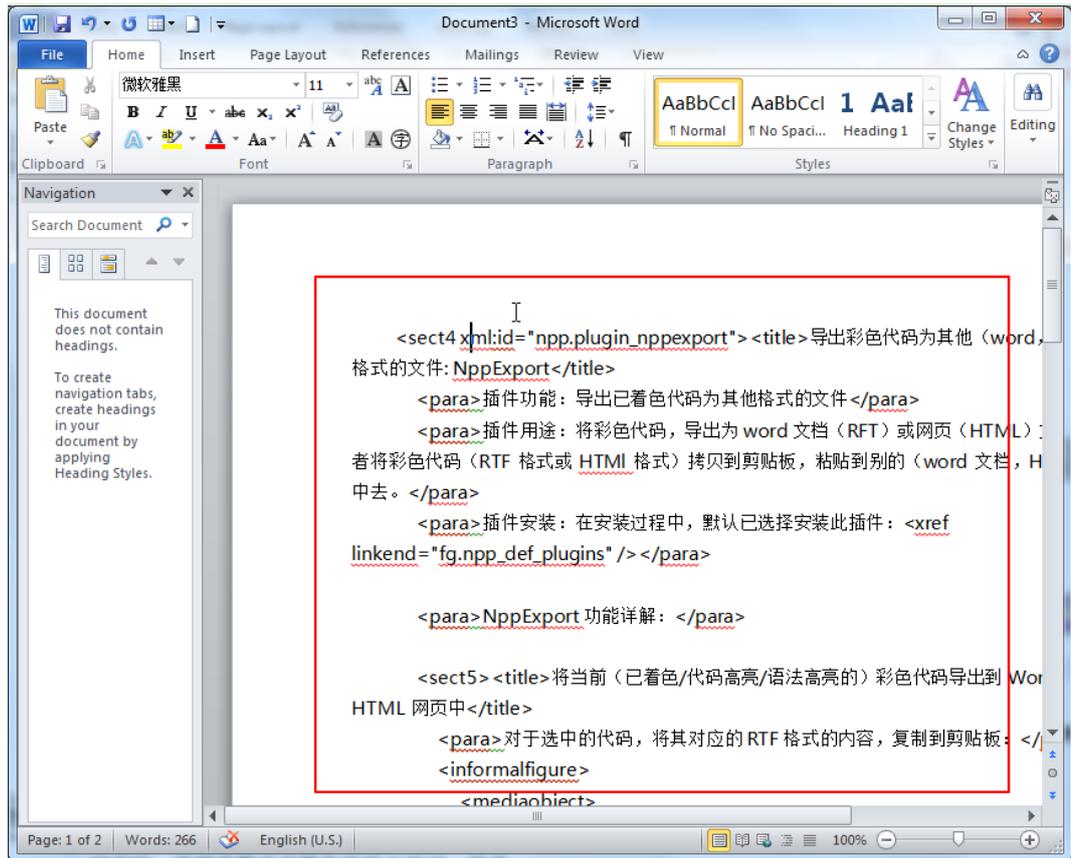
差点忘了说了, 与此处的彩色代码拷贝功能相对应的是:

对于在普通文本编辑器中, 包括当前的Notepad++编辑器, 安装普通赋值操作去的拷贝的一段代码:



然后粘贴到别的地方，是没有彩色代码的效果的：

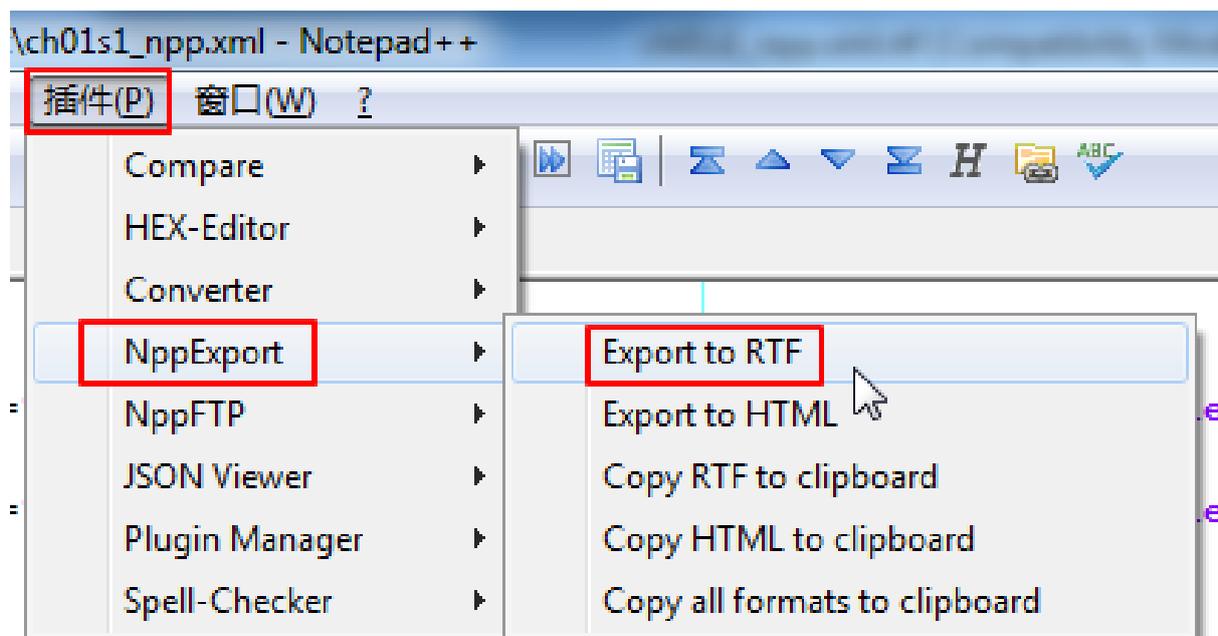
Notepad++的插件



所以，有了对比，才知道此处彩色代码拷贝的好处。

对应的，想要将整个文件全部导出的话，就是：

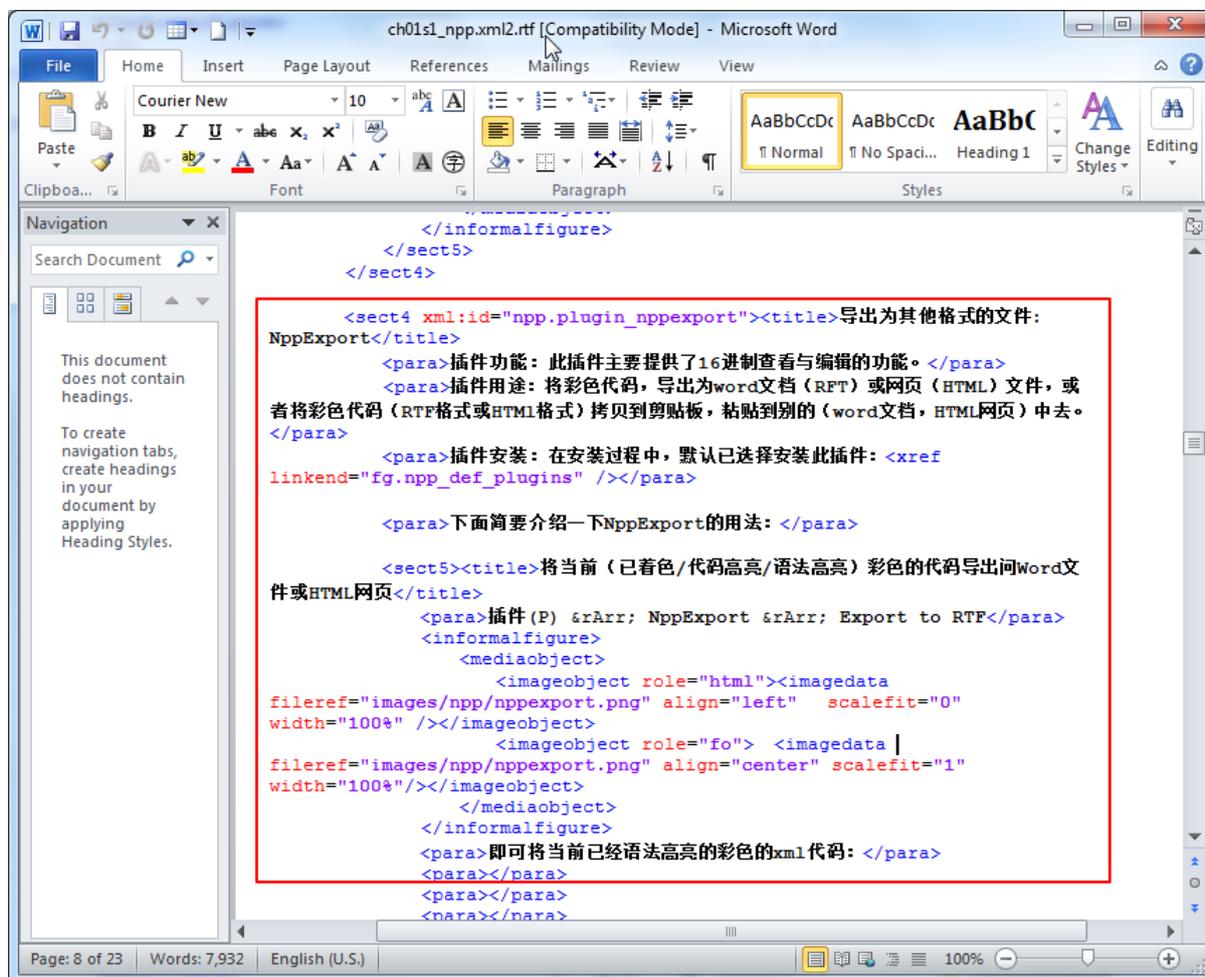
插件(P) ⇒ NppExport ⇒ Export to RTF



即可将当前已经语法高亮的彩色的xml代码：

```
251 <sect4 xml:id="npp.plugin_nppexport"><title>导出为其他格式的文件: NppExport</title>
252 <para>插件功能: 此插件主要提供了16进制查看与编辑的功能。</para>
253 <para>插件用途: 将彩色代码, 导出为word文档 (RTF) 或网页 (HTML) 文件, 或者将彩色
254 <para>插件安装: 在安装过程中, 默认已选择安装此插件: <xref linkend="fg.npp_def_p
255
256 <para>下面简要介绍一下NppExport的用法: </para>
257
258 <sect5><title>将当前 (已着色/代码高亮/语法高亮) 彩色的代码导出问Word文件或HTML网
259 <para>插件 (P) &rArr; NppExport &rArr; Export to RTF</para>
260 <informalfigure>
261 <mediaobject>
262 <imageobject role="html"><imagedata fileref="images/npp/nppexport.png"
263 <imageobject role="fo"><imagedata fileref="images/npp/nppexport.png"
264 </mediaobject>
265 </informalfigure>
266 <para>即可将当前已经语法高亮的彩色的xml代码: </para>
267 <para></para>
```

导出到word文件中了:



很明显, 如果你需要在word中粘贴此彩色代码, 此功能, 再好用不过了。

对于导出为HTML网页, 做法是类似的:



```

<sect4 xml:id="npp.plugin_hex_editor"><title>16进制编辑器: HEX-Editor</title>
<para>插件功能: 此插件主要提供了16进制查看与编辑的功能。</para>
<para>插件用途: 以16进制模式查看和编辑文件。</para>
<para>插件安装: 参考<xref linkend="plugin_download_and_install" />
<para>下面就来介绍一下HEX-Editor的功能用法: </para>

<sect5><title>以16进制模式查看当前文件</title>
<para>有几种方法实现以16进制模式查看当前文件: </para>
<itemizedlist>
<listitem>
<para>插件(P) &rArr; HEX-Editor &rArr; View in HEX C
<informalfigure>
<mediaobject>
<imageobject role="html"><imagedata fileref
<imageobject role="fo"> <imagedata fileref
</mediaobject>
</informalfigure>
</listitem>
<listitem>
<para>点击工具栏上HEX Editor的图标<inlinemediaobject><
</listitem>
<listitem>
<para>直接使用快捷键: Ctrl+Alt+Shift+H</para>
</listitem>
</itemizedlist>
<tip>
<para>退出16进制模式, 和进入16进制模式的方法相同。</para>
</tip>
<para>然后就可以以16进制模式, 查看当前文件了: </para>

```

1.4.4. 轻量级资源管理器: Light Explorer

插件功能: 轻量级的Explorer

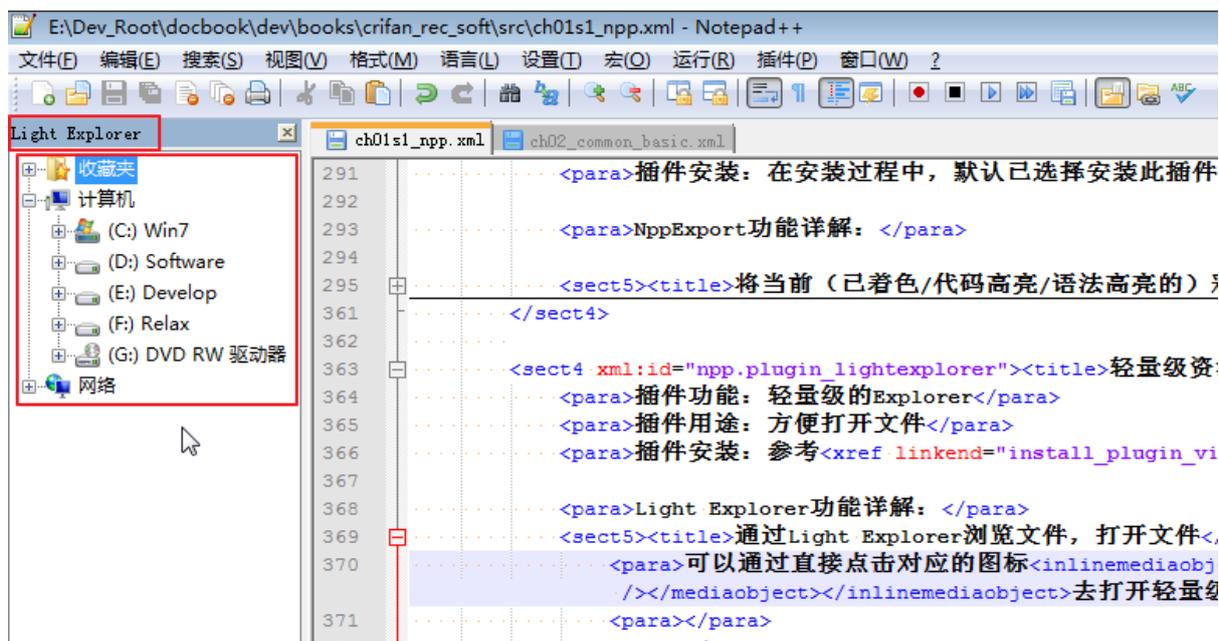
插件用途: 方便打开文件

插件安装: 参考第 1.4.1.2 节 “通过Plugin Mangager安装插件” 去安装Light Explorer插件。

Light Explorer功能详解:

1.4.4.1. 通过Light Explorer浏览文件, 打开文件

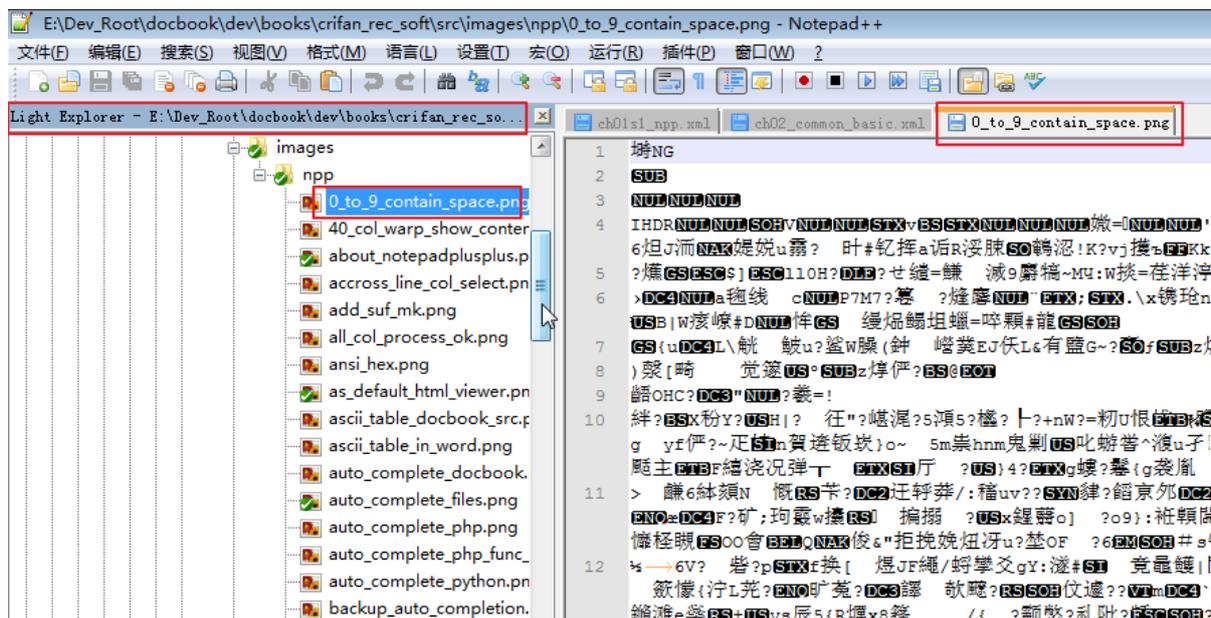
可以通过直接点击对应的图标去打开轻量级的资源管理器:



接下来，就很简单了，就是浏览文件夹，找到想要打开的文件，双击，即可用此Notepad++打开该文件了。

刚试了试，发现，此处，不论是何种文件，比如图片，都可以用Notepad++打开的。

当然，打开的图片等类型的文件的话，也都是以文本模式打开的：



1.4.5. 括号自动补全插件: XBracket Lite

之前就一直希望Notepad++拥有此功能，结果没找到选项，现在终于通过插件得以实现了。

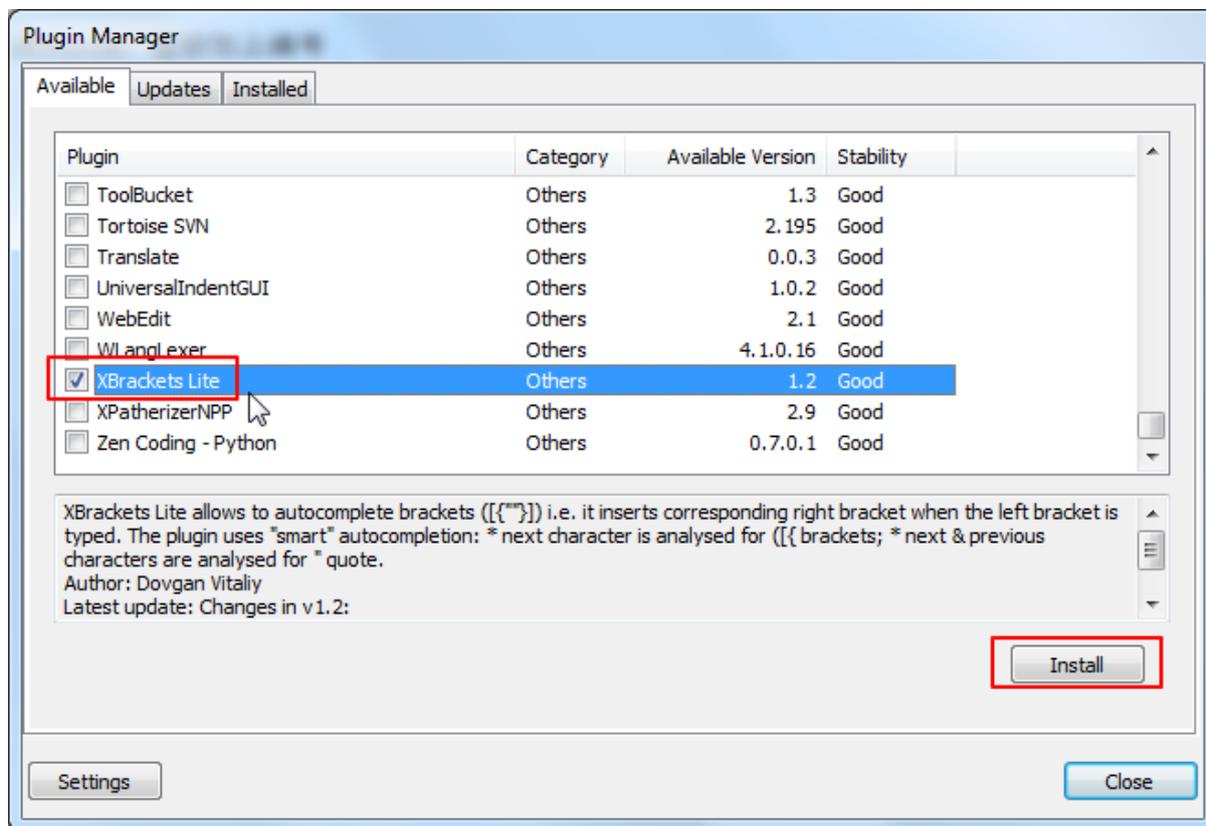
是从[Bracket autocompletion](http://sourceforge.net/apps/mediawiki/notepad-plus/index.php?title=Auto_Completion)³中得知的此插件XBracket Lite的。

插件功能：对于一些括号类的字符，比如单引号',双引号",圆括号(),大括号{,反括号[等，自动帮你实现自动补全

³ http://sourceforge.net/apps/mediawiki/notepad-plus/index.php?title=Auto_Completion

插件用途：减轻了写代码和编译xml类的文件时候的工作量

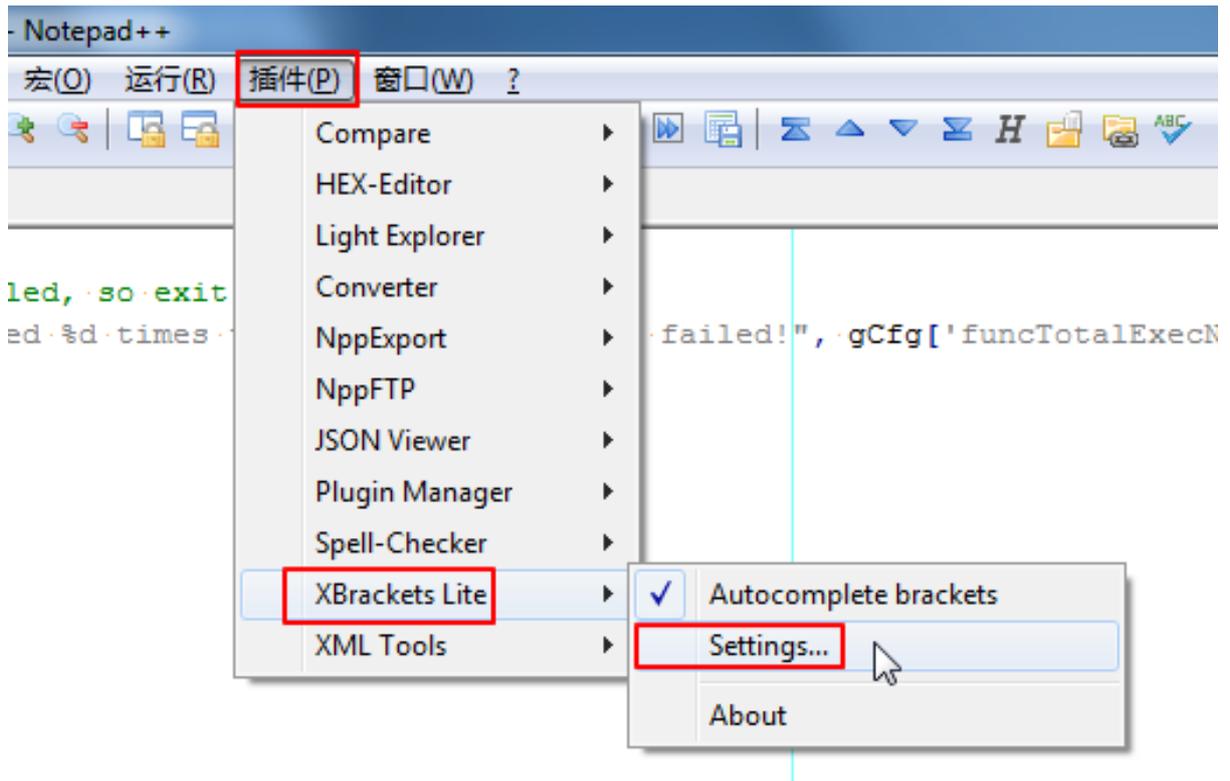
插件安装：参考第 1.4.1.2 节 “通过Plugin Manager安装插件” 去安装XBracket Lite插件，截图如下：



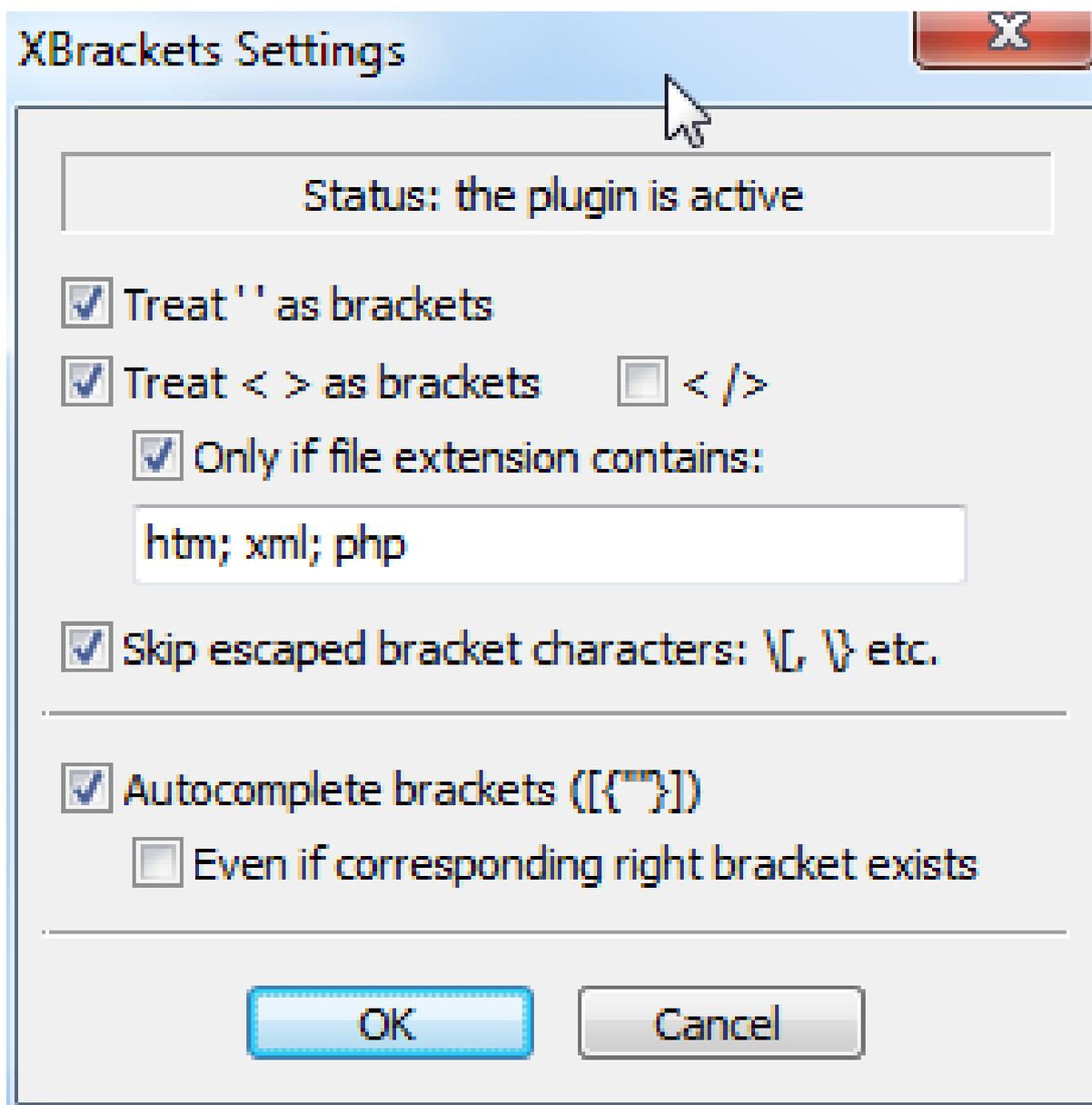
XBracket Lite功能详解：

1.4.5.1. 通过XBracket Lite实现括号的自动补全

先去打开相应的设置：



再根据自己的需要去设置：



其中解释一下相应的选项的含义：

- Treat'' as brackets
把单引号'，也看成是括号，这样以后输入单个单引号，也可以实现自动补全另一个单引号了。
- Treat< > as brackets
把尖括号也看成是括号的一类，这样就方便了在xml，html等文件中写tag的时候，自动帮你补全。
如果选择了: </>，那么输入了<后，就补全出来/>了。
如果选择了：Only if file extension contains，那么就只有当前是这些htm,xml等文件的时候，才对于尖括号使用自动补全。别的类型文件中，不去对尖括号实现补全操作。
- Skip escaped bracket characters:\[, \} etc.
当输入\[或者是\}的时候，就不自动补全。
因为在写代码时，很多时候反斜杠加上这些括号，本身就是想要表示单个字符而已，所以不需要补全。
- Autocomplete brackets({{\"}})

即 是否启用此插件。

设置好之后，启用该插件，随便打开个文件，输入这些括号类字符的时候，就会发现，可以自动帮你实现自动补全了。

1.4.6. 支持Notepad++中使用js脚本实现功能扩展: jN

之前叫做NppScripting，后来改名为jN

插件功能：使得Notepad++中可以通过运行对应的javascript脚本，其会扩展一些额外功能，添加相应的菜单，提供各种功能

插件用途：使得你可以实现，自己写javascript脚本，几乎可以实现任何你想要的功能。而且本身其已经实现了很多常用的功能，供你使用。

插件安装：

参考[第 1.3.2.1 节 “手动安装插件”](#) 下载jn.zip⁴

解压后，把jN文件夹和jN.dll拷贝到Notepad++的plugin目录下即可。

更多的细节，参考[【记录】利用Notepad++的jN插件中的URL编码解码插件，实现从错误的google地址中提取原始url的功能](#)⁵

1.4.7. 显示函数列表：Function List

插件功能：用于显示出函数的列表。支持很多种语言，包括C, C++, Resource File, Java, Assembler, MS INI File, HTML, Javascript, PHP, ASP, Pascal, Python, Perl, Objective C, LUA, Fortran, NSIS, VHDL, SQL, VB and BATCH

插件用途：这样以后打开Python等文件，就可以通过双击显示出来的函数列表中的某函数，实现快速定位到相应的位置了。

插件安装：

参考[第 1.3.2.1 节 “手动安装插件”](#) 下载FunctionList_2_1_UNI_dll.zip

解压后，按照install.txt中的提示去：

把FunctionList.dll拷贝到 安装目录\Notepad++\plugin下

把Gmod Lua.bmp和C++.flb拷贝到 安装目录\Notepad++\plugin\config下

把FunctionListRules.xml拷贝到 %APPDATA%\Notepad++\plugins\config下

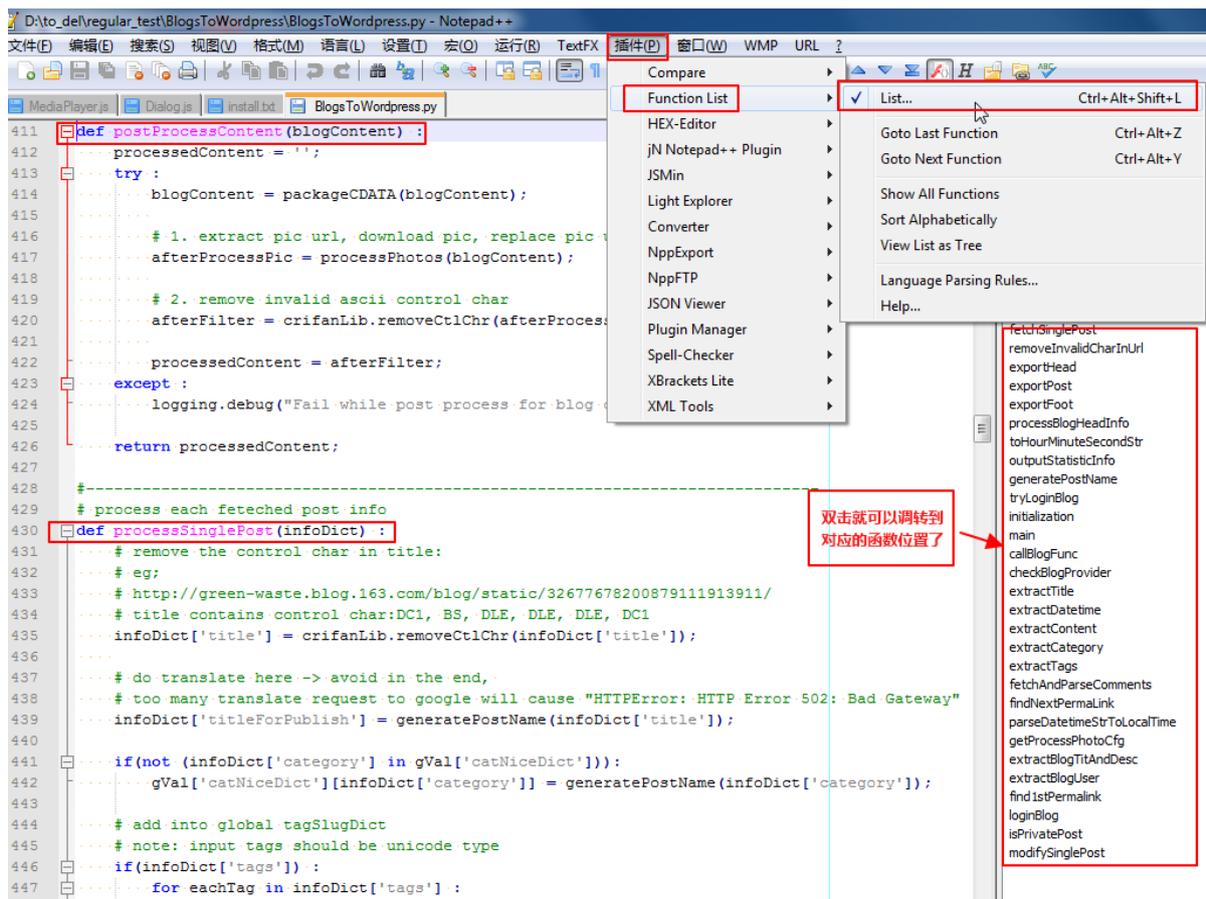
其中需要注意的是，此处的FunctionListRules.xml就是定义了各种语言的搜索函数的规则。必须把该文件拷贝到上述位置才可以。像我最开始的时候，拷贝到安装目录\Notepad++\plugin\config下是不可以的。

而%APPDATA%的位置，在我这里的是C:\Users\CLI\AppData\Roaming\Notepad++\plugins\config

之后通过 插件 ⇒ Function List ⇒ List...或者快捷键Ctrl+Alt+Shift+L，就可以打开当前文件的函数列表了：

⁴ <http://www.softwarecanoe.de/jn/jn.zip>

⁵ http://www.crifan.com/npp_plugin_jn_nppscripting_url_encode_decode/



1.4.8. js代码压缩,js代码格式化,JSON查看器: JSMinNpp

插件功能：包含三个功能：

1. js代码压缩
2. js代码格式化功能
3. JSON代码查看器，以树状列表显示

插件用途：三合一功能的插件，很是方便使用。

插件安装：

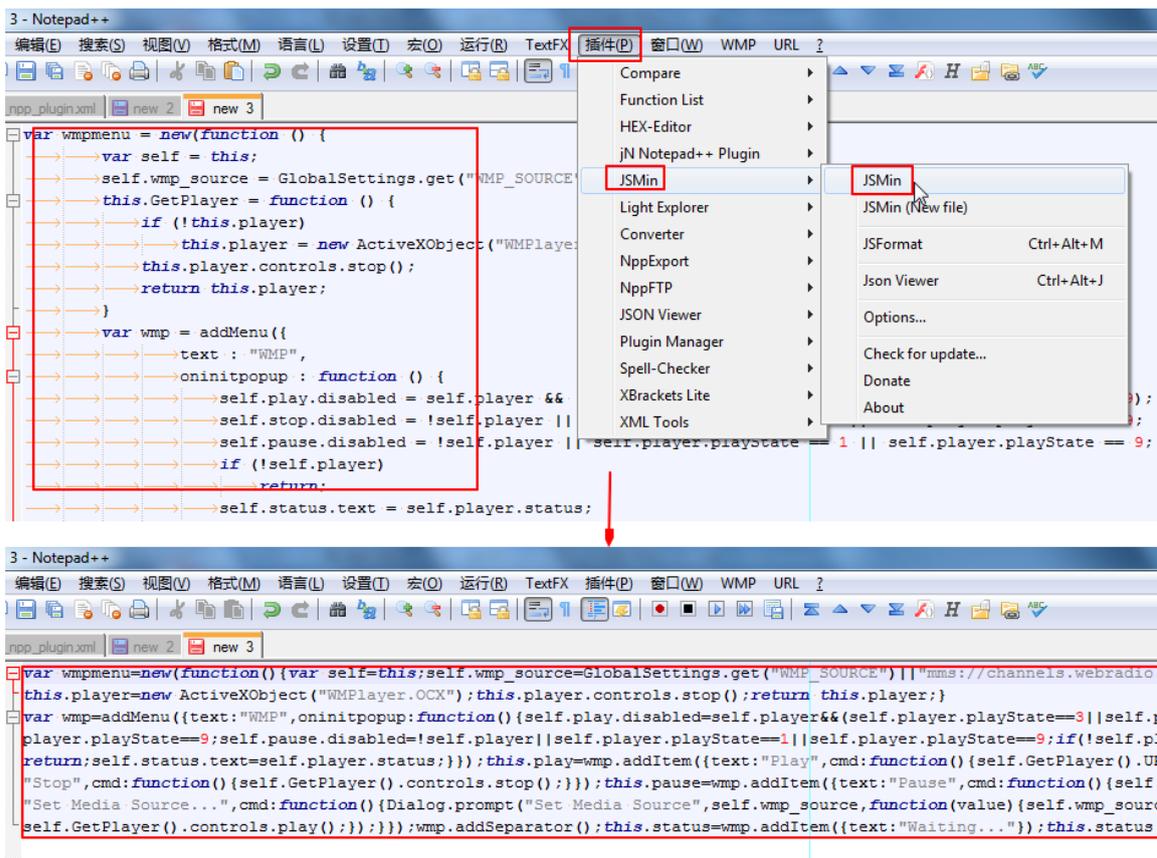
参考第 1.3.2.1 节 “手动安装插件”，去JSMinNpp⁶下载JSMinNPP.1.11.2.uni.zip,解压后把JSMinNPP.dll拷贝到plugin目录即可。

去试了试，真的很好用：

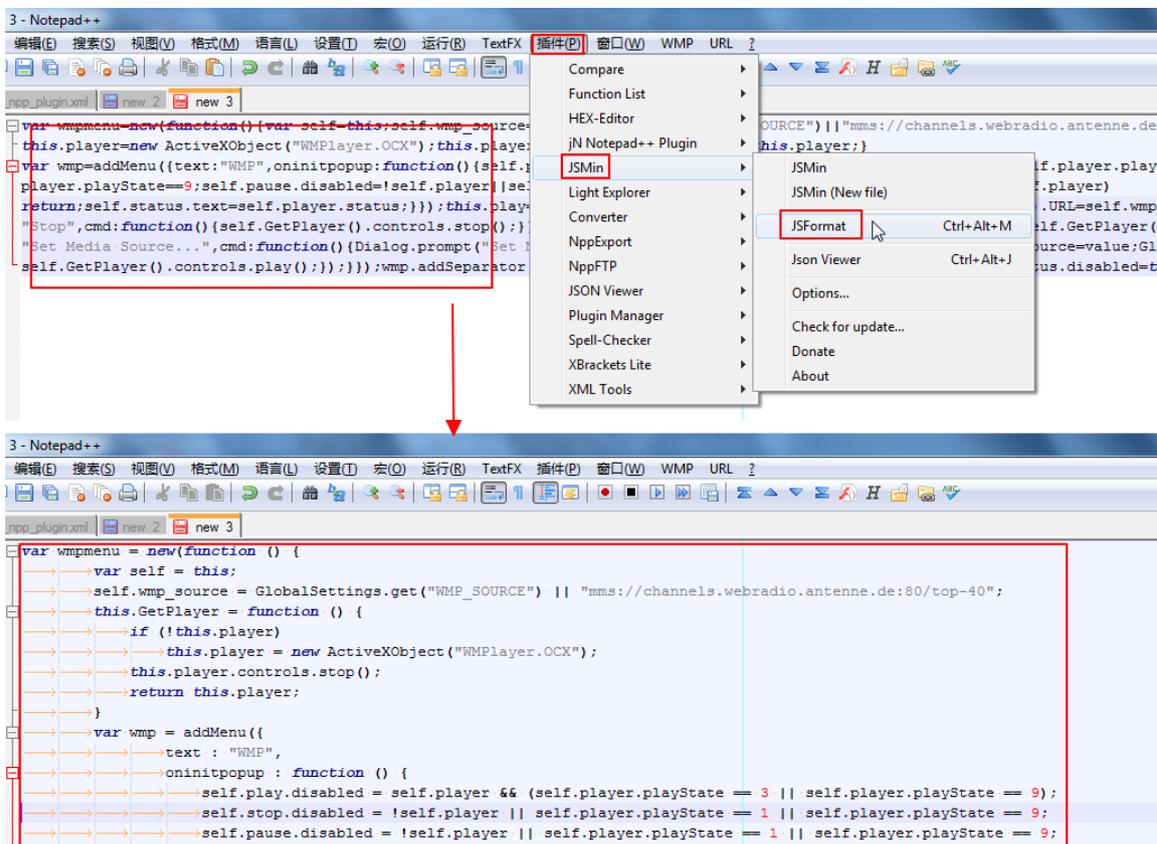
1. js代码压缩

⁶ <http://jsminnpp.sourceforge.net/>

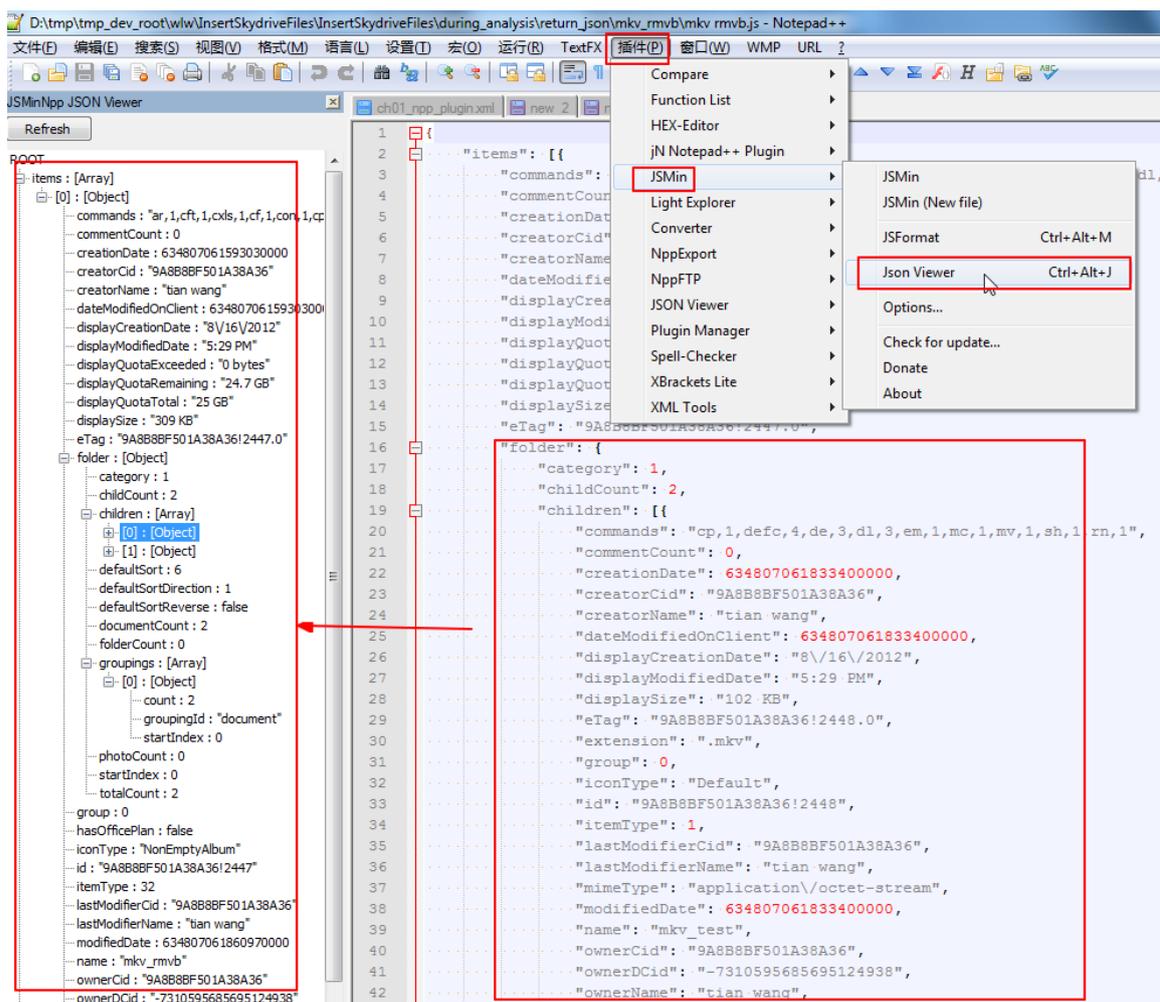
Notepad++的插件



2. js代码格式化功能



3. JSON代码查看器，以树状列表显示



话说，装了这个插件之后，以前一直使用的，在线版本的HTML/JS代码格式化网站：[Beautify JavaScript or HTML](http://jsbeautifier.org)⁷，就可以减少使用次数了。

因为javascript，json等代码，都可以使用此插件格式化并以树状显示了。

只不过，对于html代码的格式化，还是需要用到那个jsbeautifier的。

⁷ <http://jsbeautifier.org>

第 2 章 Notepad++ 的安装过程

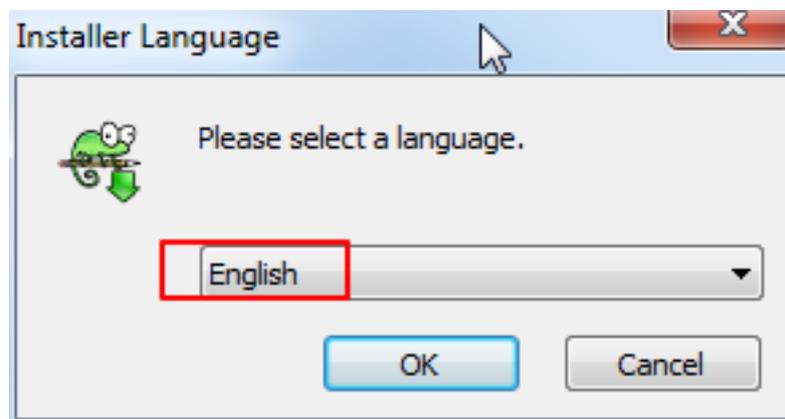
其实关于Notepad++的安装，本身操作步骤简单，没有太多解释必要，但是对于安装过程中很多的参数，有必要解释一下：

下面就详细解释一下Notepad++的安装过程：

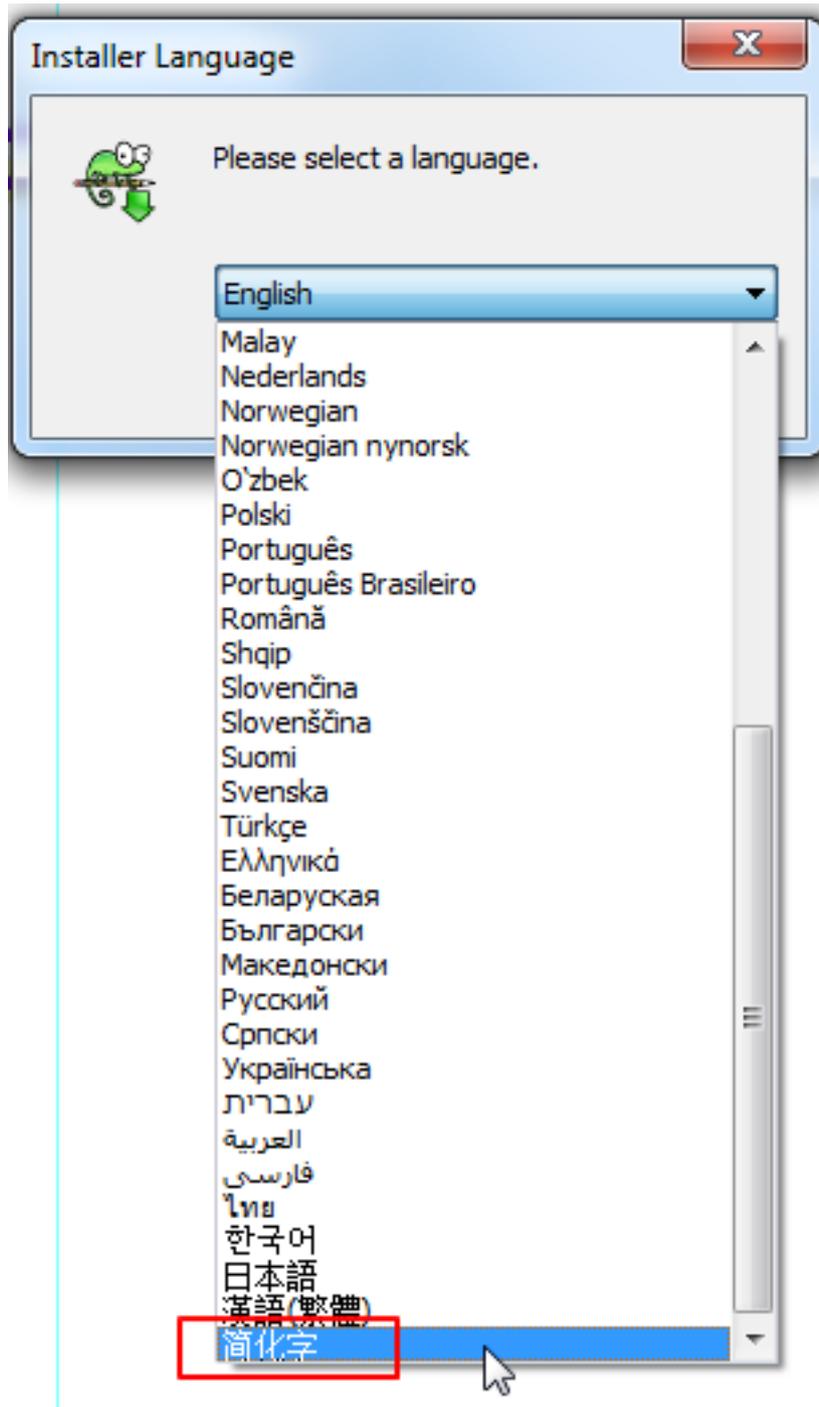
下载了notepad的可执行文件，比如npp.6.1.1.Installer.exe后，双击它，接着就可以安装了：

2.1. Notepad++ 安装：选择语言

其默认语言是English：



此处改为“简体字”：

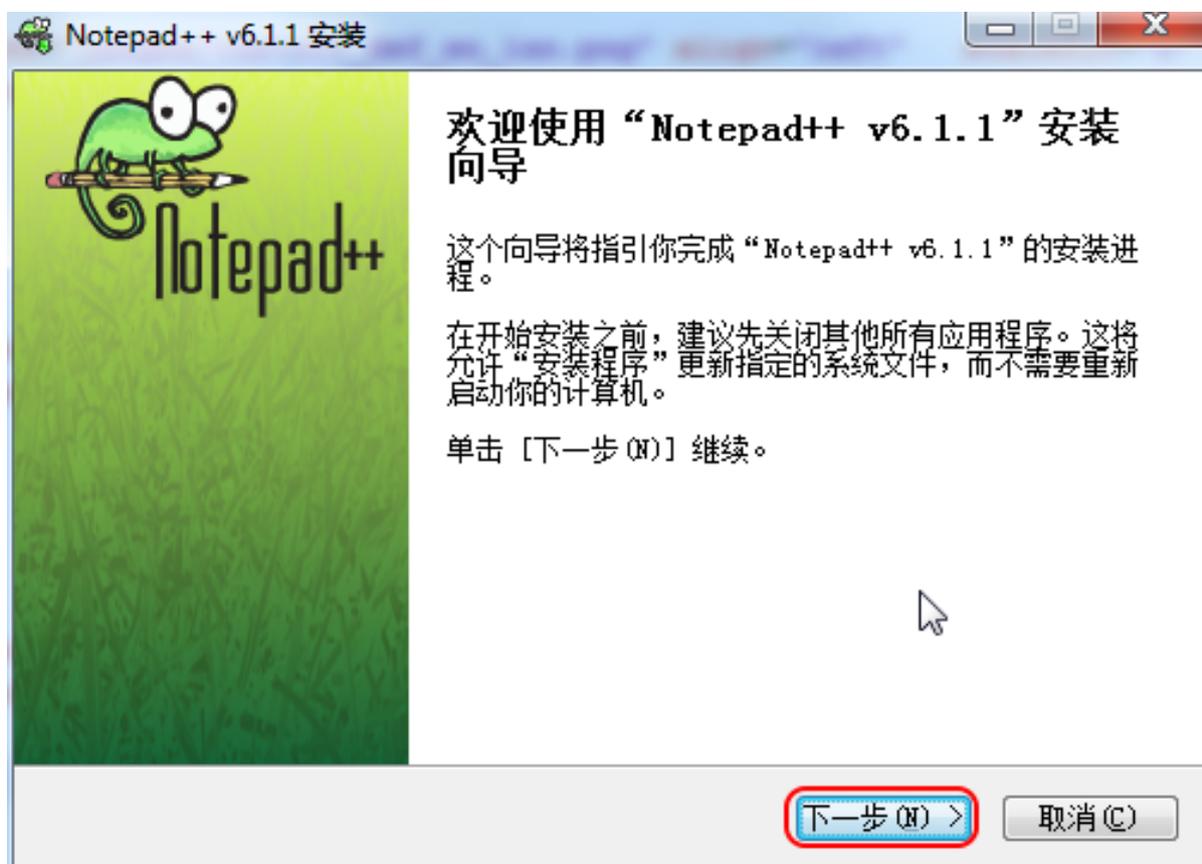


然后点击OK：



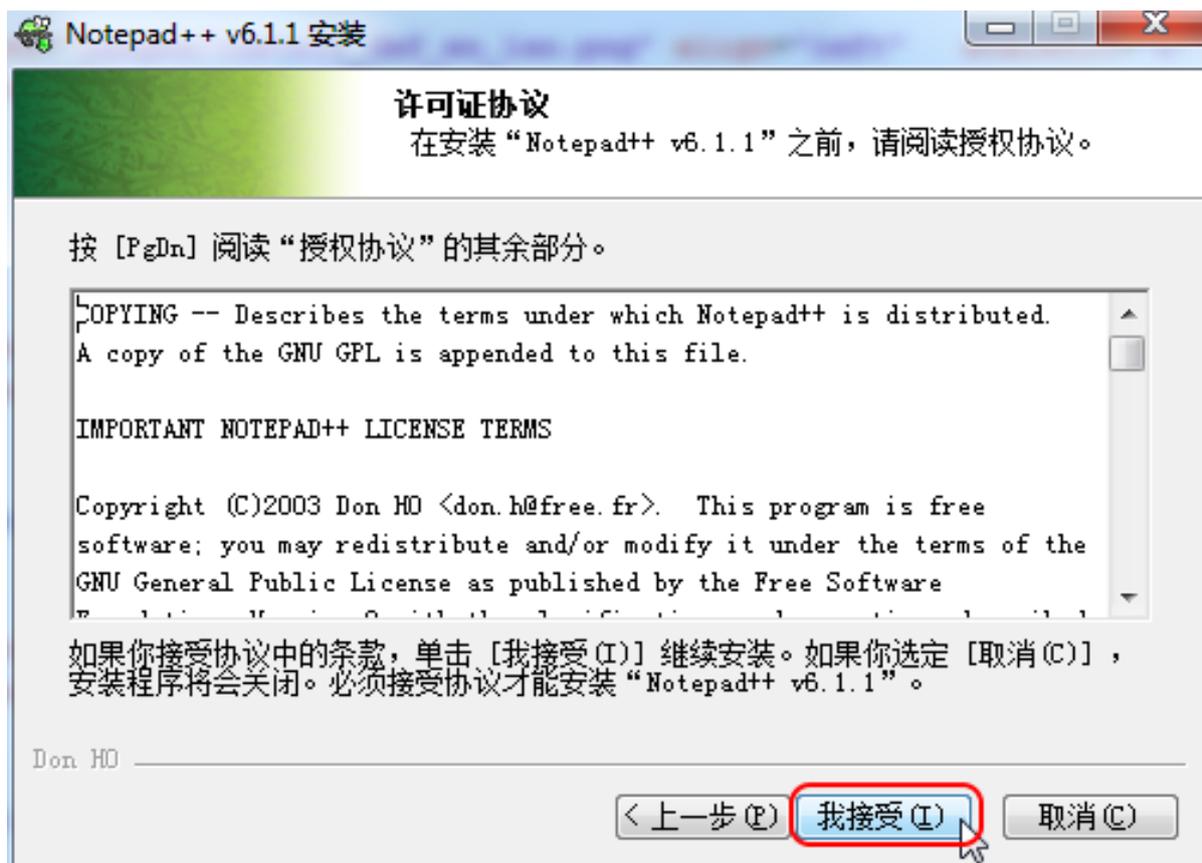
2.2. Notepad++安装：安装向导

然后出现欢迎界面，点击下一步：



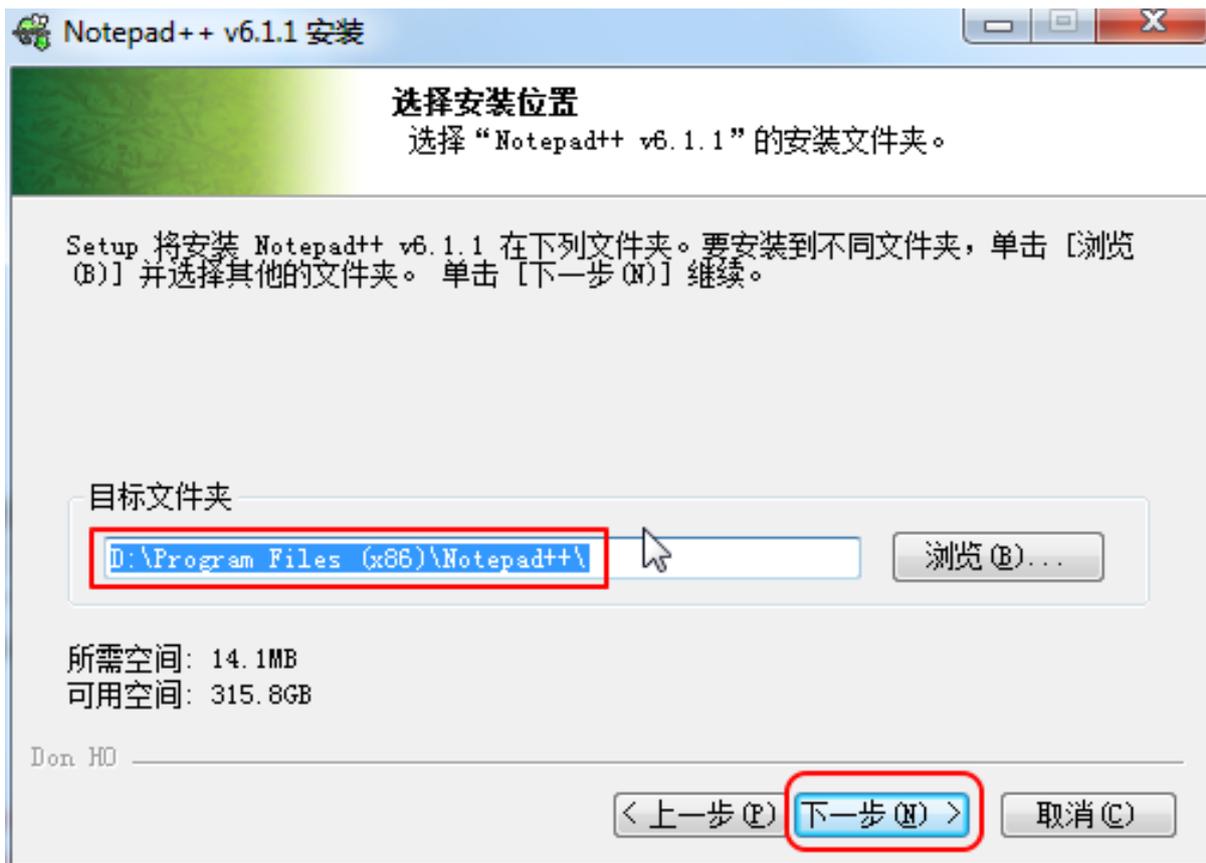
2.3. Notepad++安装：许可证协议

出现“许可证协议”，点击我接受：



2.4. Notepad++安装：选择安装位置

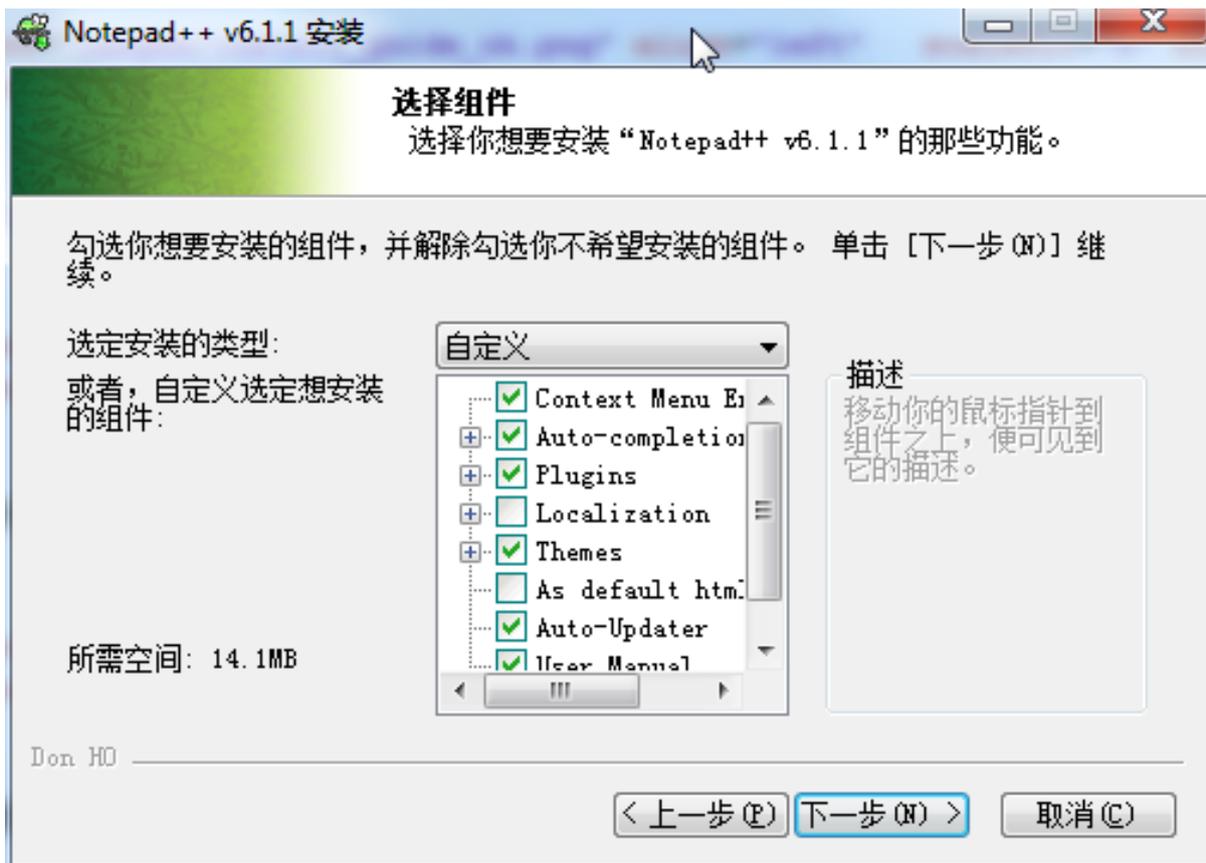
然后是“选择安装位置”的界面，对于其中的“目标文件夹”，可以使用默认设置的值，也可以选择一个别的路径，比如我此处改为了D盘的相应位置：D:\Program Files (x86)\Notepad++\



再点击下一步

2.5. Notepad++安装：选择组件

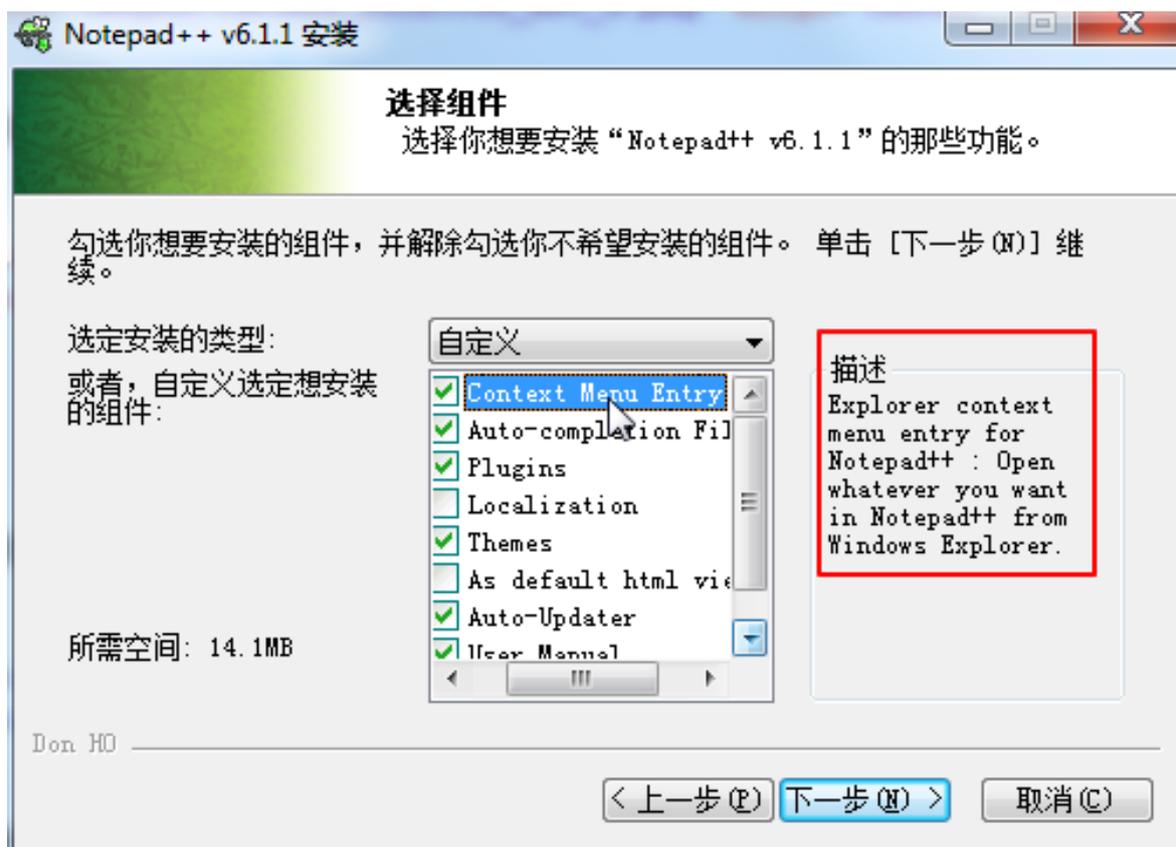
出现“选择组件”的界面：



可以看到，此处有好几个选项：

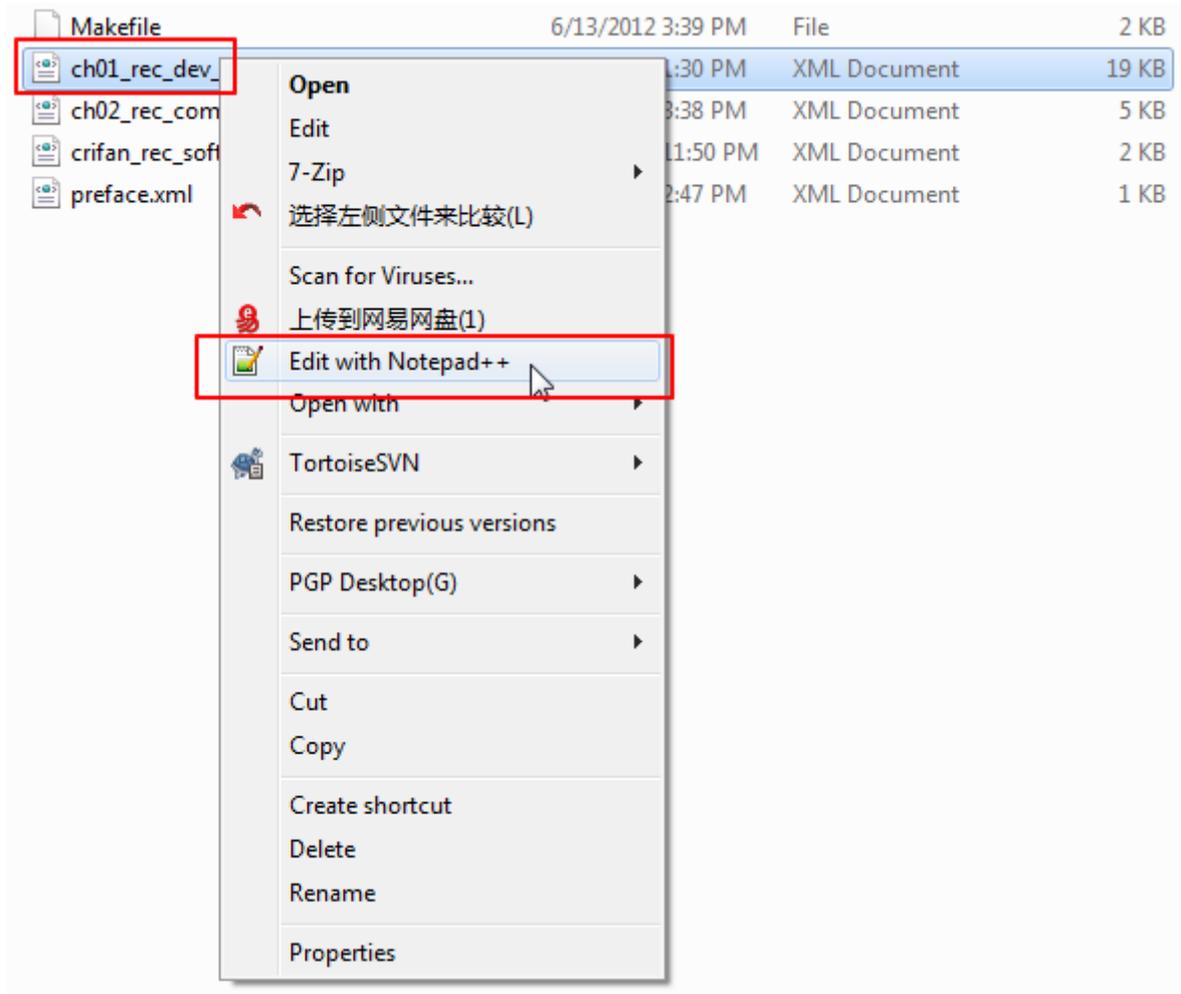
- Context Menu Entry

图 2.1. Notepad++ 的右击打开文件功能



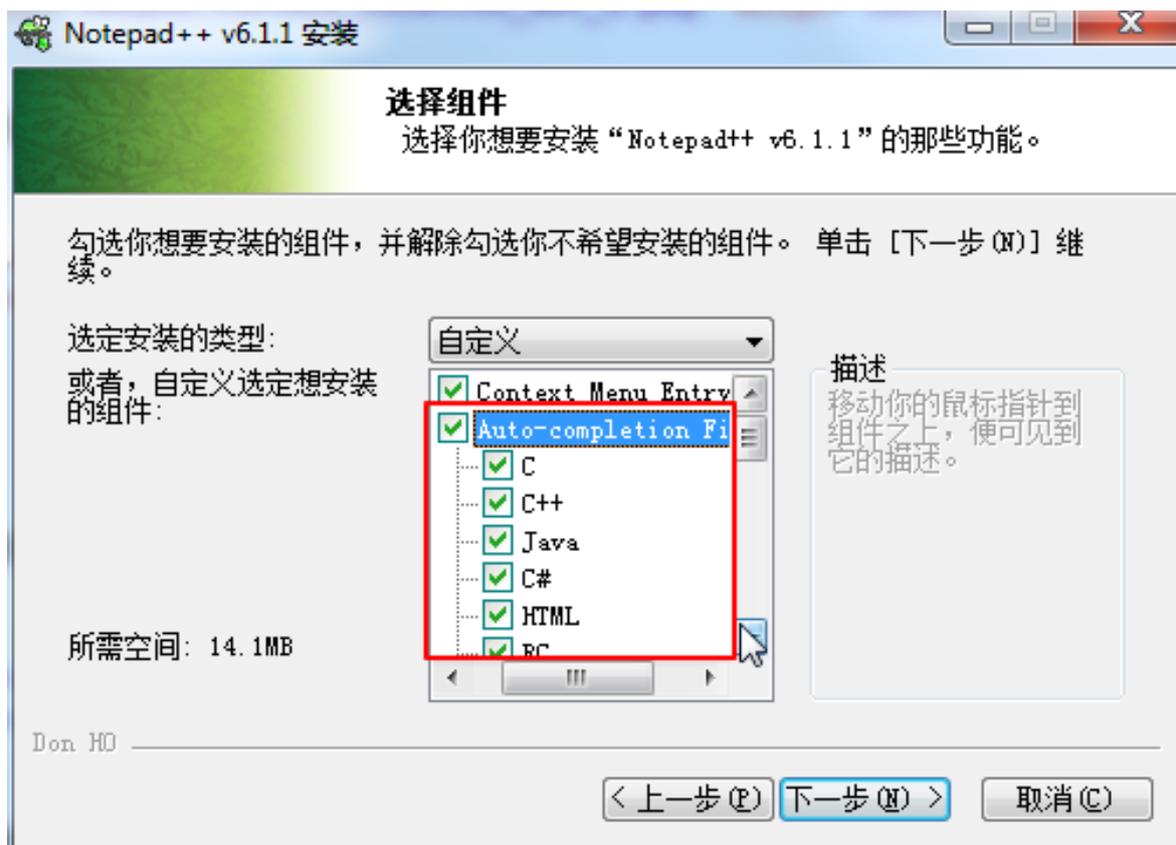
默认已勾选

其作用是添加右键快捷键。即，对于任何文件右击所出现的菜单中，都会有“Edit with Notepad++”：



其显而易见的好处是，可以很方便的启动Notepad++去打开相应的（任何）文件。

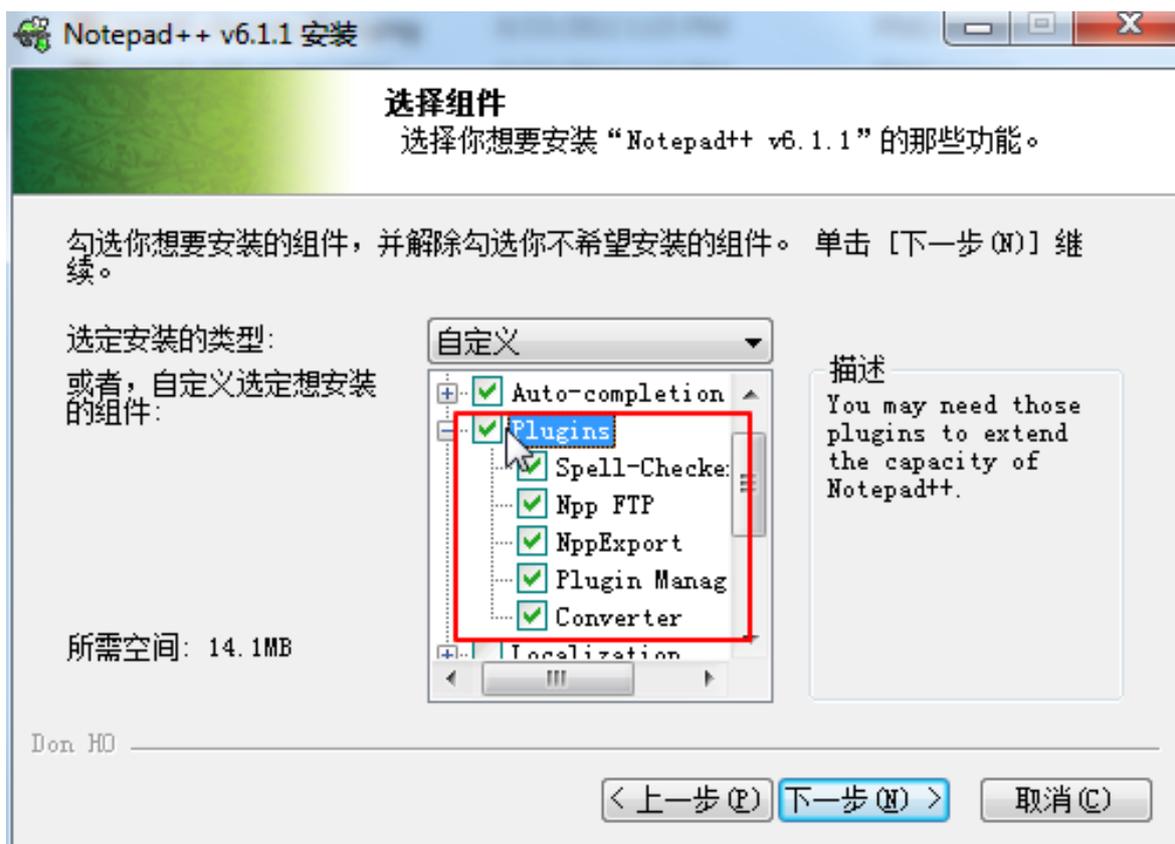
- Auto Complete Files
Notepad++安装选项：Auto Complete Files.



默认已勾选其作用是，对于各种语言或规范（C，C++，Docbook，Python等），支持[第 3.8 节 “Notepad++的自动完成”](#)的功能。

- Plugins

图 2.2. Notepad++ 默认已安装了一些插件

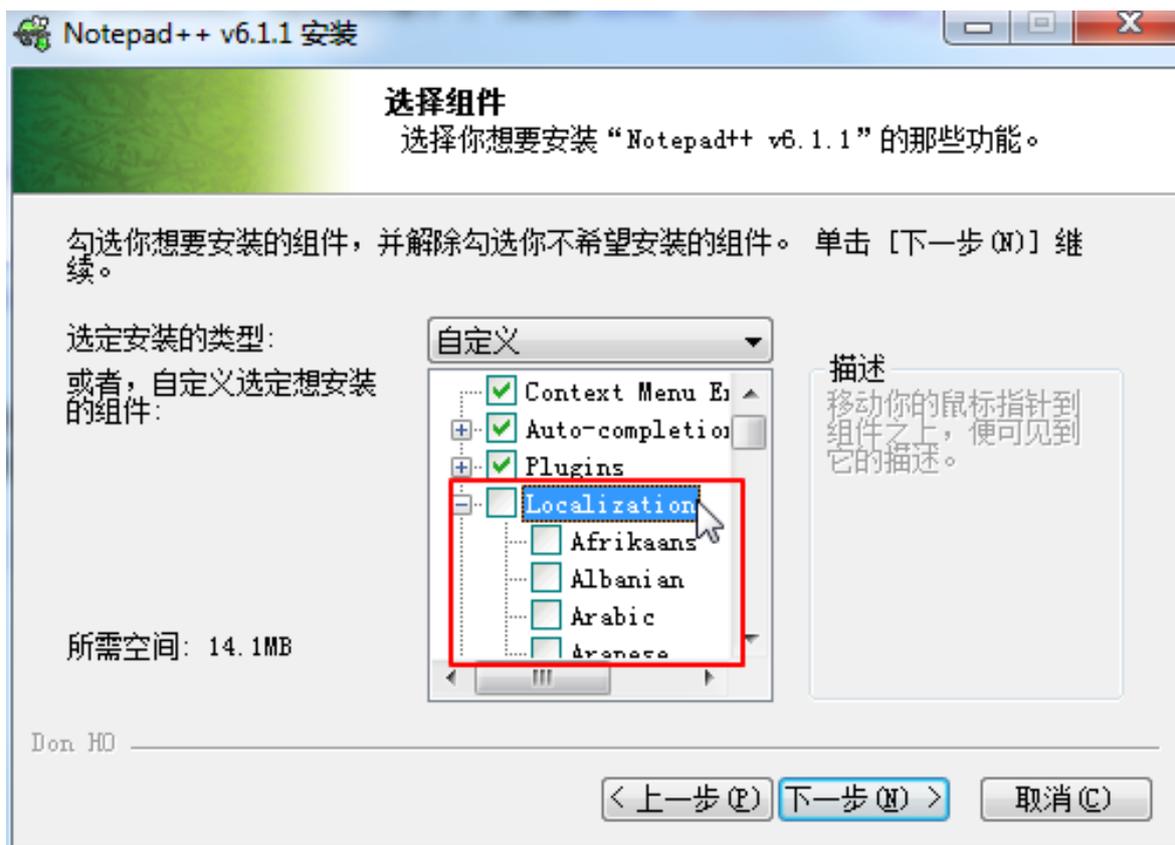


默认已勾选

其作用是，默认是否安装一些插件。当然，这些插件是足够的好用和常用，所以Notepad++才会默认将他们集成进来，默认勾选上的。

关于这些插件的详细功能的介绍，参见[第 1.4 节 “Notepad++ 中常用的插件”](#)

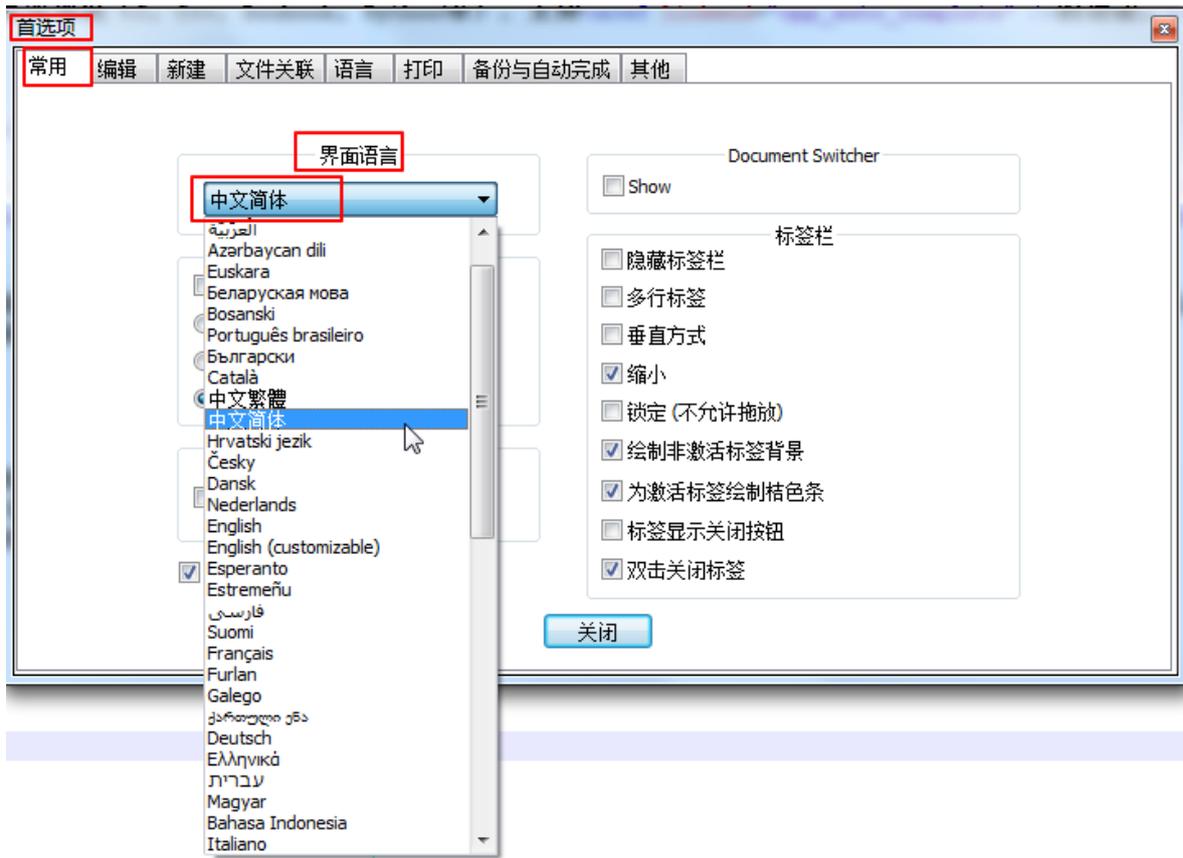
- Localization



默认无勾选

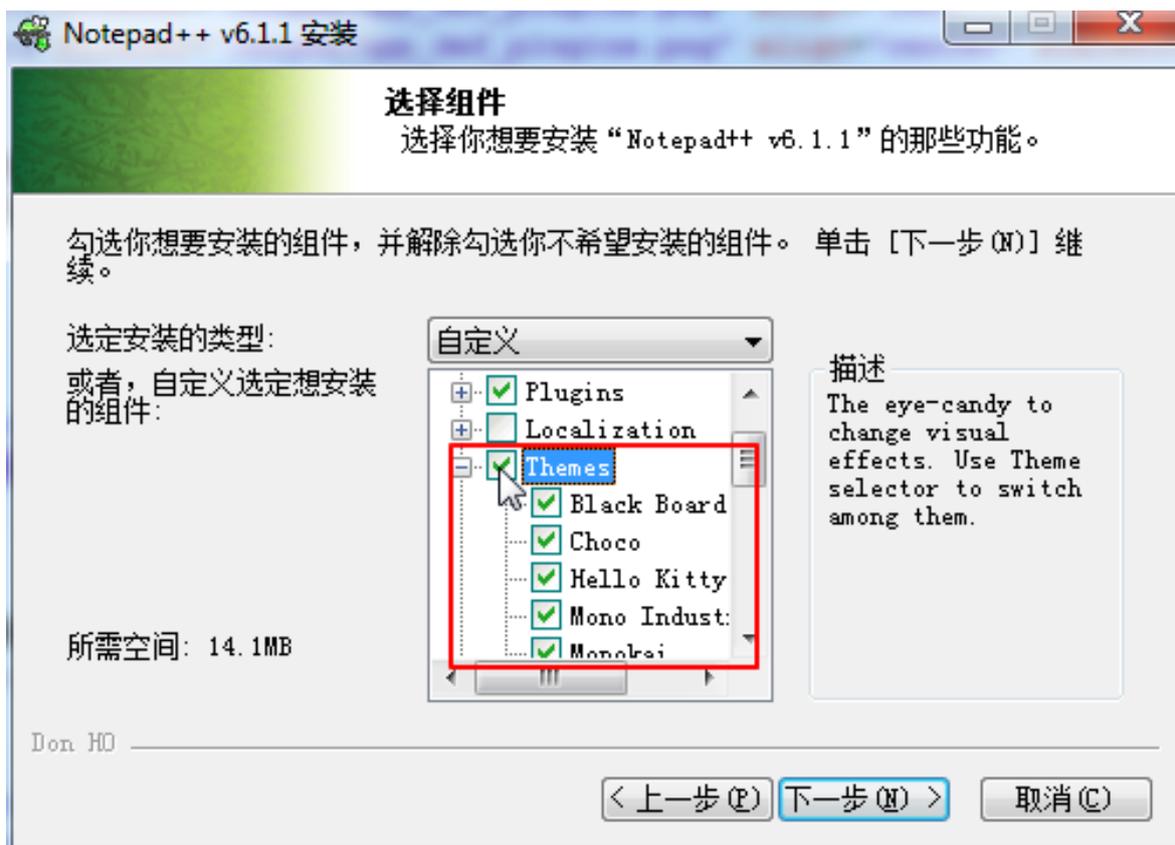
其作用是，在Notepad++的界面显示方面的对于各种语言的支持，即Notepad++本身软件的各个工具栏等部分的显示的文字所用的语言。

对应着Notepad++中的：设置(T) → 首选项... → 常用,中的“界面语言”：

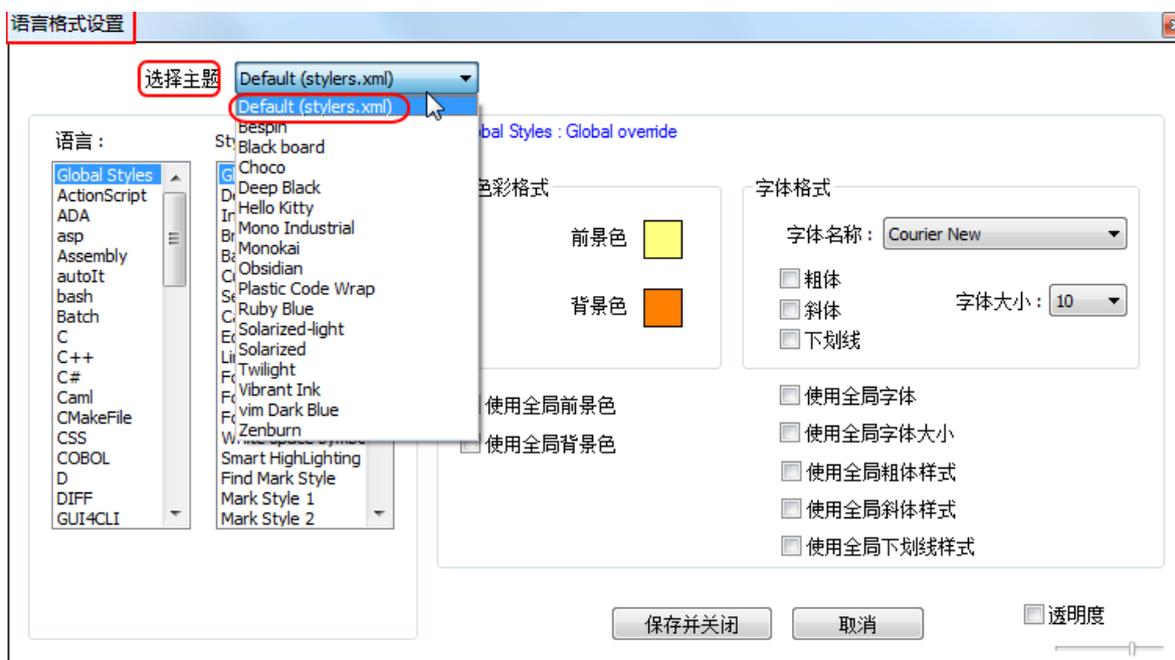


此处可选可不选，因为本身除了当前中文和默认英文，其他的也基本遇不到。

- Themes
Notepad++ 安装选项：Themes.

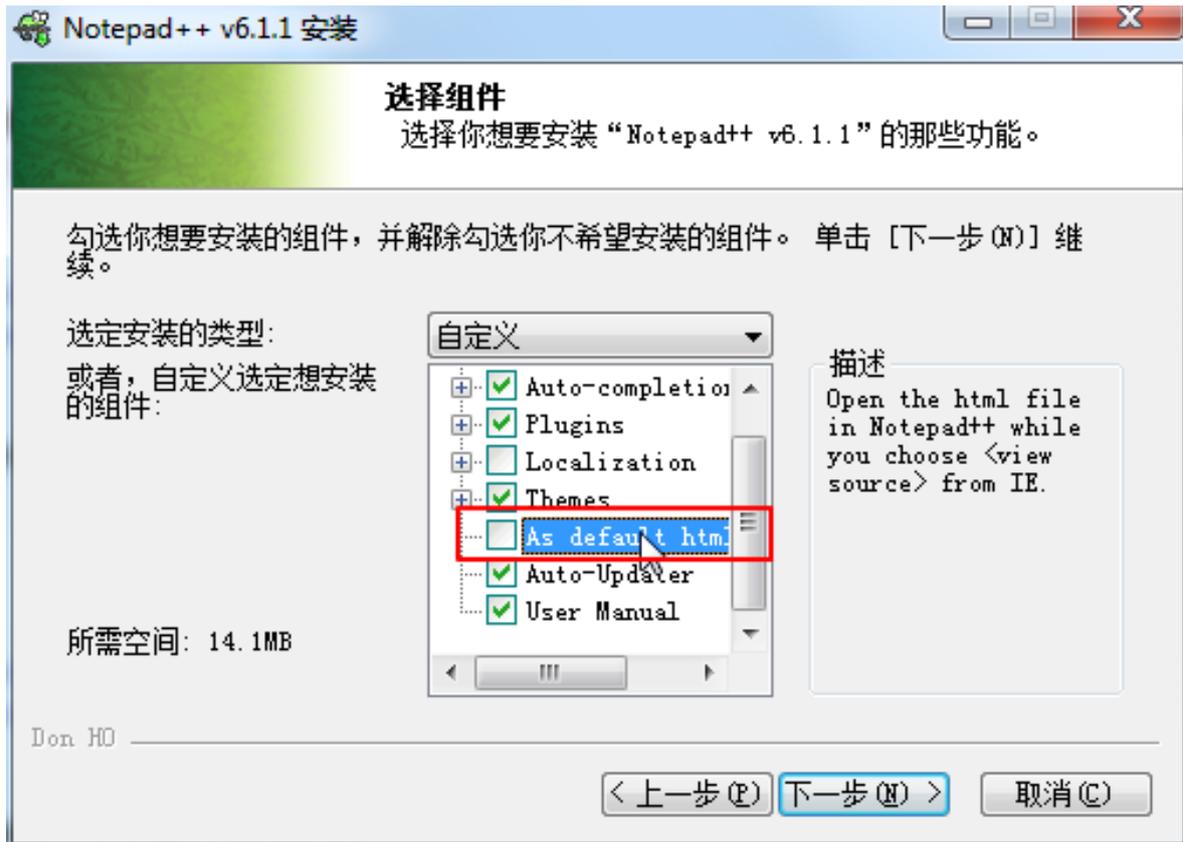


默认已勾选。即Notepad++中，是否支持各种主题。其对应着Notepad++中的：设置(T) → 语言格式设置...，中的“选择主题”：



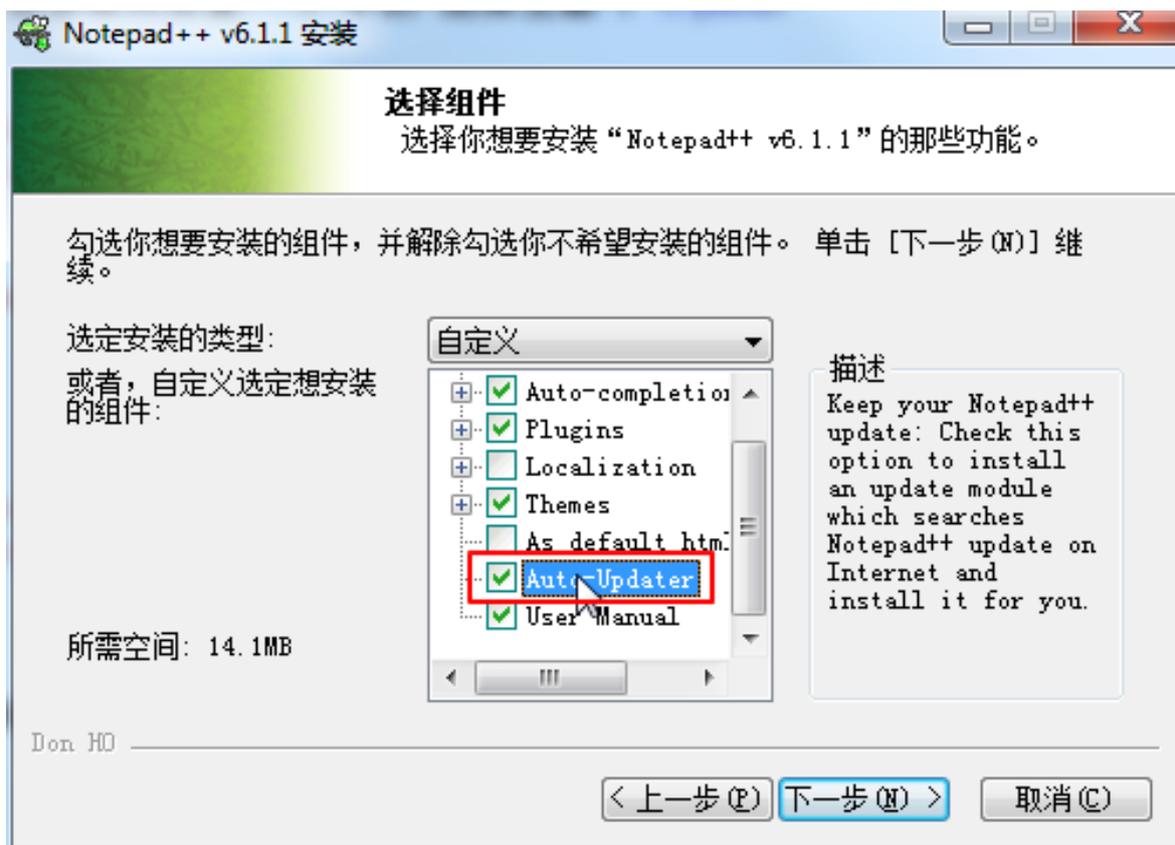
关于每种主题的效果，可以参看[第 3.18 节 “Notepad++的多主题”](#)

- As default html viewer
Notepad++安装选项：As default html viewer.



默认无勾选。建议勾选。选上后，其就可以用作默认的HTML网页源码编辑器了，详情参看：[第 3.9 节 “Notepad++的默认HTML查看器”](#)

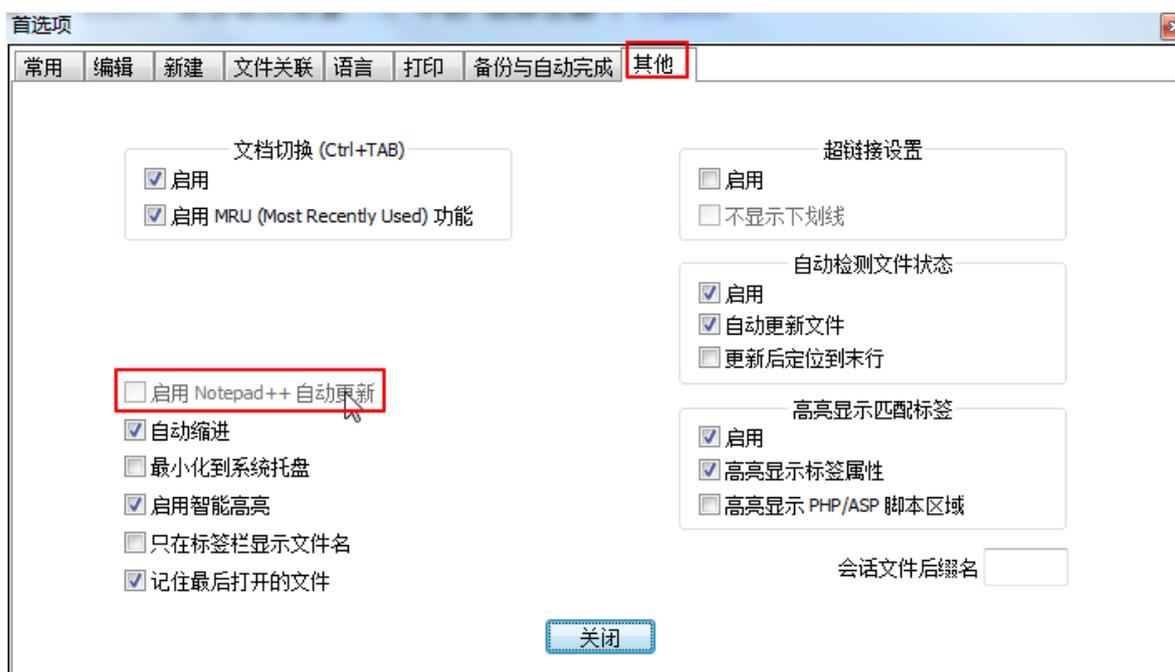
- Auto-Updater



默认已勾选。

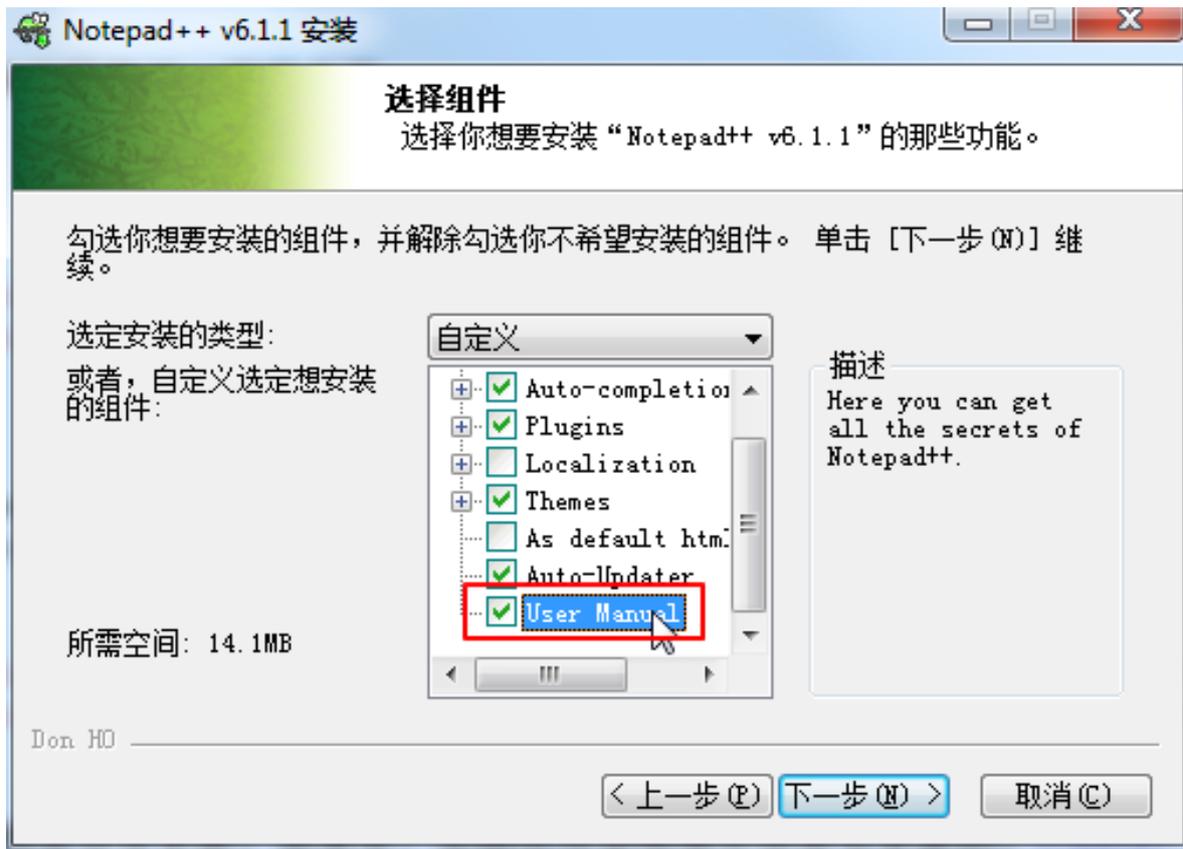
作用为，安装此模块后，就可以支持Notepad++的自动更新了。

其对应着：设置(T) → 首选项... → 其他,中的“启用Notepad++自动更新”：



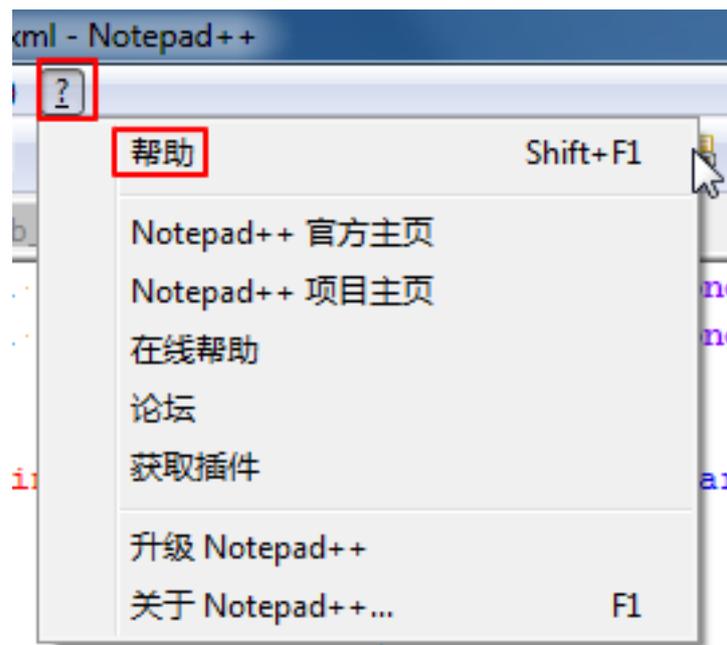
启用后，每次Notepad++启动时，都会去检测是否有更新。如果有，会跳出对话框提示你是否更新的。

- User Manual

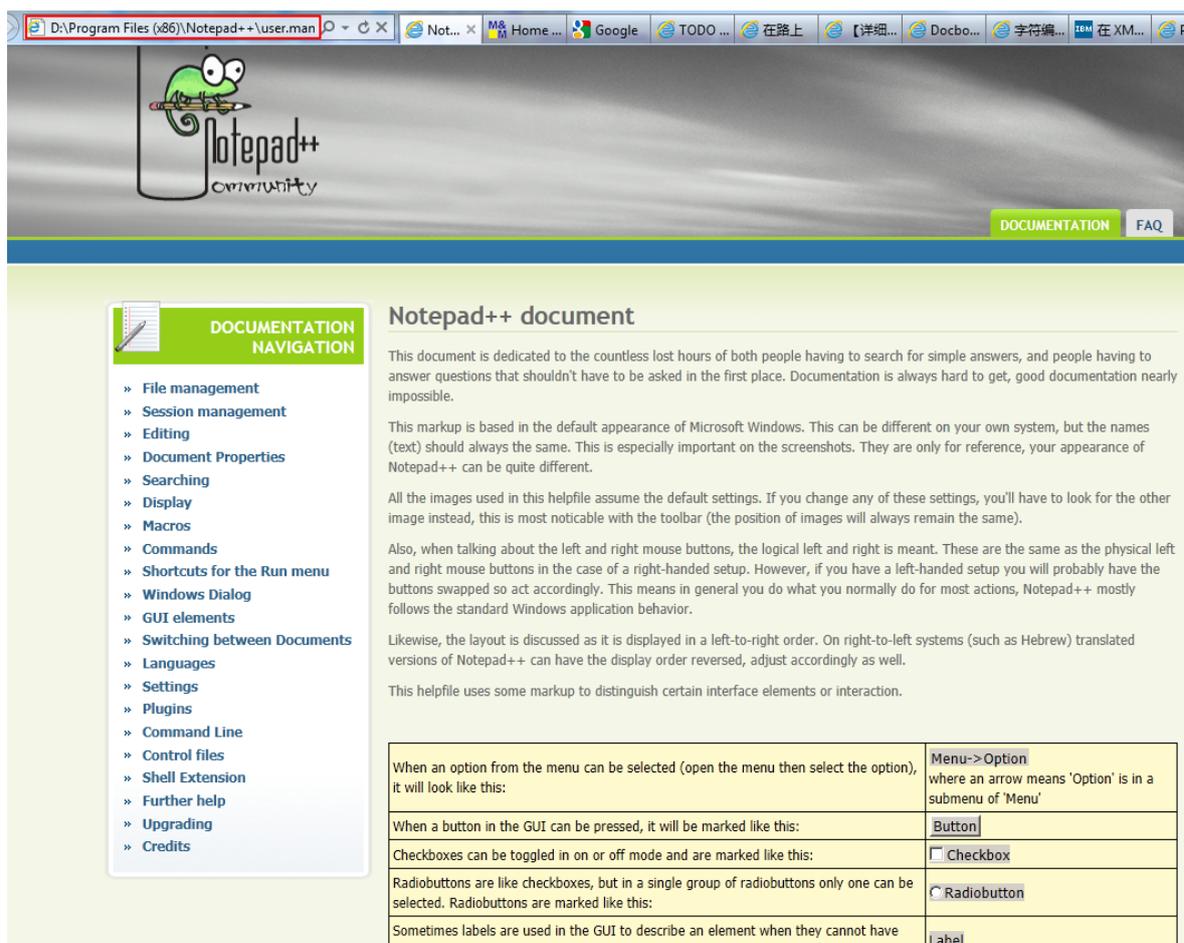


默认已勾选。

作用是，安装本地的帮助文件。其是网页的形式。Notepad++中通过点击：? → 帮助 Shift+F1：



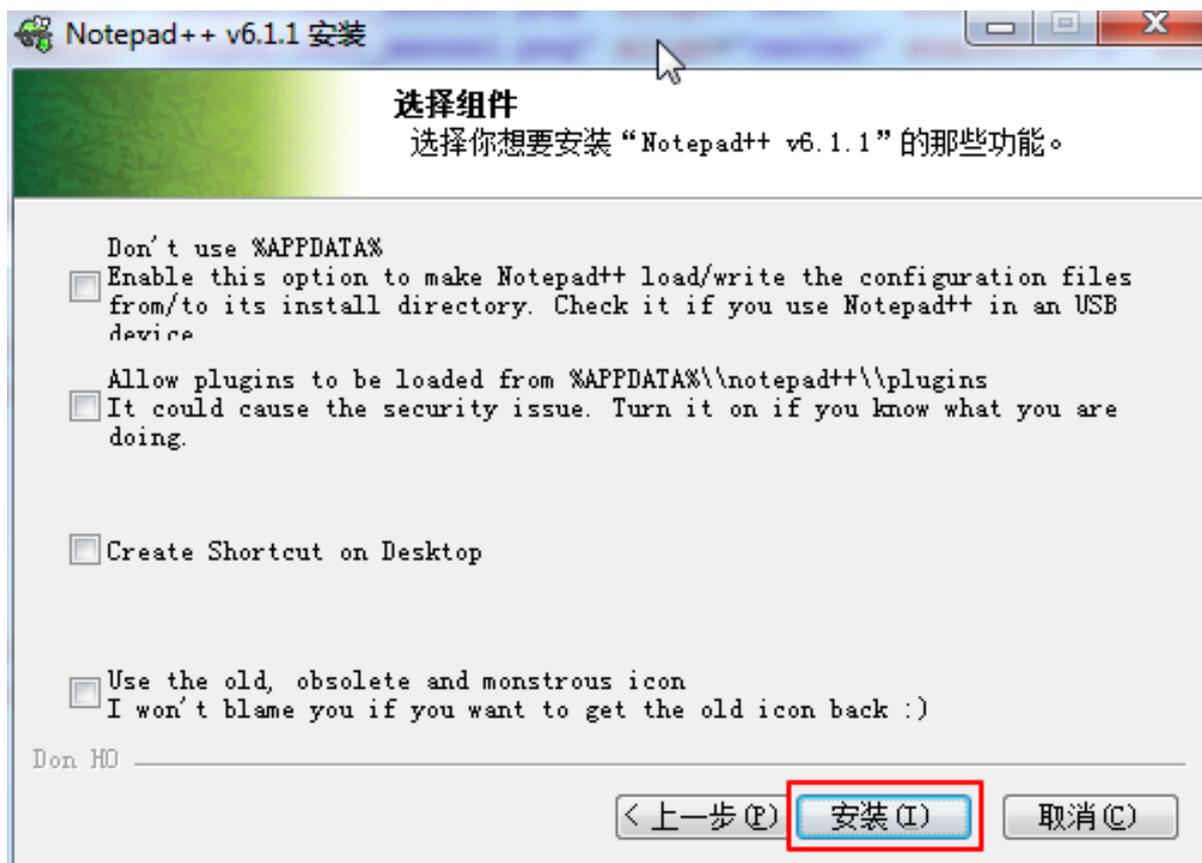
即可打开对应的网页文件：



根据需要，选择所需的组件后，点击"下一步"

2.6. Notepad++安装：选择组件(参数)

会出现另外一些组件参数选择：



其也有一些参数需要解释：

- Don't use %APPDATA%
默认没勾选。

作用是允许配置文件放在Notepad++的安装路径下。由此，可以实现支持把整个Notepad++都放到U盘中，到处带着跑了。

其用途主要是，如果本身自己对于Notepad++有很多自定义的配置，而换了个电脑，就得重新配置Notepad++，很是麻烦。

而此时，就可以在安装的时候，选择此项，这样以后所有的配置都是放在安装路径下，然后你可以把此Notepad++全部文件都放到U盘中。

这样，换了电脑，也都可以继续使用你自己的这个Notepad++了。

- Allow plugins to be loaded from %APPDATA%\\notepad++\\plugins
默认没勾选。

作用是不太了解。看起来好像是说，允许从%APPDATA%\\notepad++\\plugins的位置，载入插件，如此做法，有一定的安全隐患。

其估计指的是，有些不安全的插件，放到对应位置的话，其也会自动载入，所以可能会导致一些安全问题。

- Create Shortcut on Desktop
默认没勾选。

在桌面穿件一个Notepad++的快捷方式。

常见使用Notepad++的方式都是从文件右键中选择“Edit with Notepad++”，所以一般不需要在桌面创建快捷方式。

- Use the obsolete and monstrous icon
默认没勾选。

使用旧的风格图标。

估计旧的风格图标很难看，所以作者才说，即使你选择此项，其也不会责怪你的。要不然肯定是默认也勾选此项才对。

然后点击“安装”，即可。

2.7. Notepad++安装：安装完成

安装完毕后，会出现相应的安装完毕界面：



然后点击“完成”，即可启动默认勾选了的Notepad++了。

至此，Notepad++安装完毕。

第 3 章 Notepad++ 的功能详解

下面就来详细介绍Notepad++中的N多好用易用的功能：

下列功能的排名先后顺序，由个人所认为的该功能的重要性高低而决定

3.1. Notepad++ 支持右击打开所选文件

即，通过右击某文件，然后选择"Edit with Notepad++"，就可以用Notepad++打开该文件了。

虽然此功能很简单，但是相当的实用，而且也是最常用的功能。



注意

此功能是由开始安装Notepad++的时候，选择是否安装此功能的。

Notepad++安装程序中，默认已选择开启该功能了。

详情参考：[图 2.1 “Notepad++的右击打开文件功能”](#)

3.2. Notepad++的语法高亮

```

8 #
9 # this makefile should be located in the parent dir of src Makefile
10
11 LS_OUTPUT=$(shell ls)
12 #SUB_BOOKS=LS_OUTPUT
13 #SUB_BOOKS=$(shell for each_ls in $(LS_OUTPUT); do if [ -d $each_ls ];
then $(SUB_BOOKS)+=$each_ls; fi; done)
14 #SUB_BOOKS := $(foreach each_ls, $(LS_OUTPUT), $(if test -d
$(each_ls), $(each_ls)))
15
16 # all: make_all
17 # release: make_release
18 # clean: make_clean
19 # clean_all: make_clean_all
20 # clean_release: make_clean_release
21
22 #goto every sub folder(single book)'s src dir to do 'make xxx' action
23 clean clean_all clean_release all release release_copy:
24   → @echo "Current path=`pwd`;
25   → @echo "LS_OUTPUT="$(LS_OUTPUT);
26   → for each_item in $(LS_OUTPUT); \
27   → do \
28   →     if [ -d $$each_item/src ]; then \
29   →       cd $$each_item/src && \
30   →       echo `pwd` && \
31   →       if [ -f Makefile ]; then \
32   →         make $@; \
33   →       fi; \
34   →       cd ../../; \
35   →     fi; \
36   → done;
37
length:1181 lines:37 Ln:1 Col:1 Sel:0 UNIX ANSI INS

```

语法高亮，Syntax Highlight，也被叫做代码高亮

其实，其他很多文本编辑器也都支持此语法高亮功能，但是发现Notepad++支持的语言更多，更方便使用。

个人用过的，就有C,C++,Python,XML,HTML,xml,Javascript等，

甚至还支持一些相对很多人不是很常用的类型，比如makefile，tex/LaTex等。

而对于这些的支持，作为开发者的话，如果等你遇到了，有此需求了，就会发现，这些功能是多么有用。

3.2.1. 语法高亮的实现逻辑

Notepad++对于任何一个文件，想要实现对应的语法高亮的话，大概逻辑是：

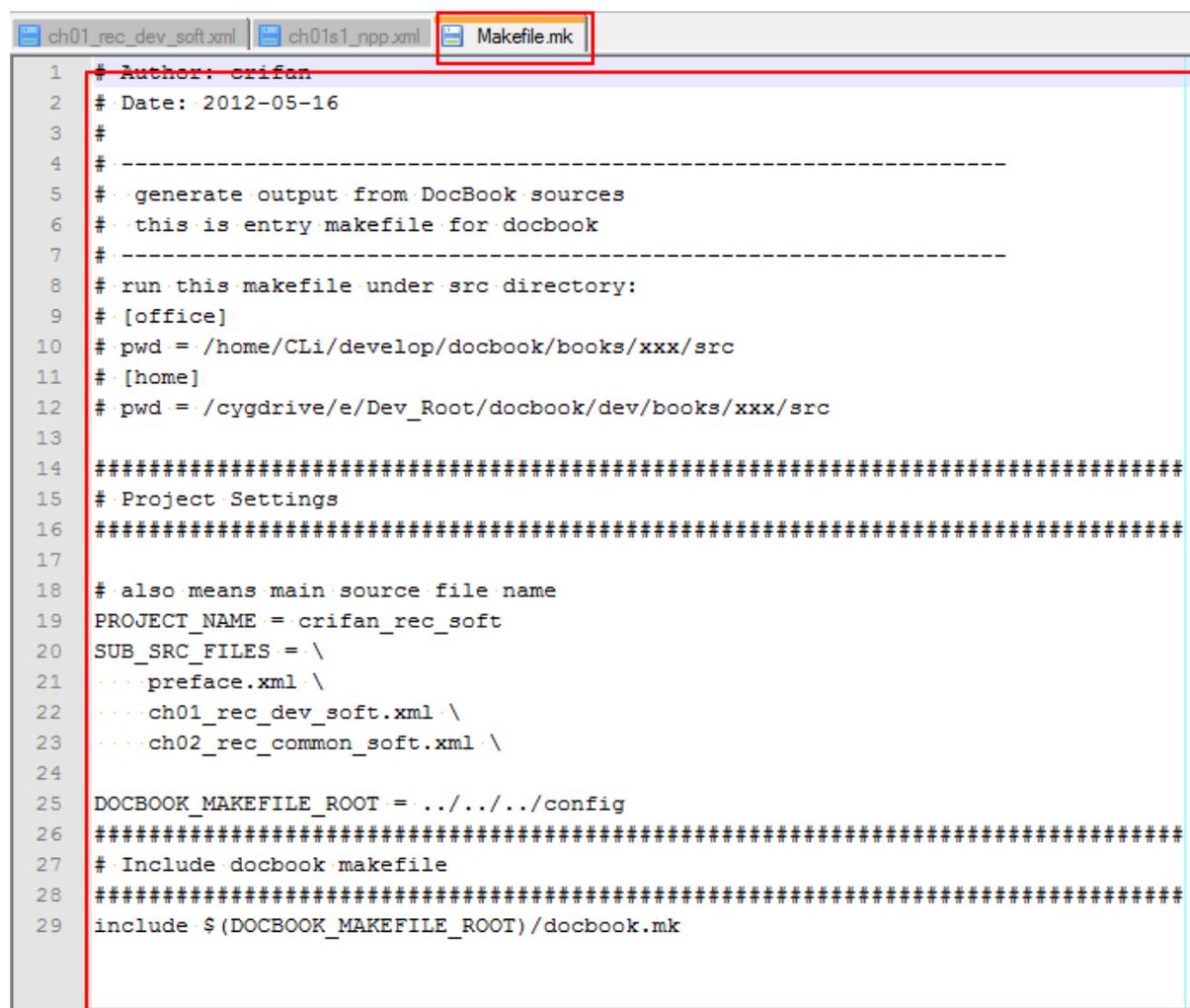
先是根据该文件后缀，去判断该文件属于哪种类型的文件

然后再调用对应的语法高亮配置，给不同的关键字（和符合对应逻辑的代码），设置不同的属性，包括字体粗细，改变颜色等

最终得到我们所看到的语法高亮后的代码的效果。

因此，如果文件没有后缀，或者不支持该文件的后缀名，则Notepad++打开该文件后，也没法自动对其实现语法高亮。

比如对于一个文件，Makefile.mk，其是一个makefile文件，用Notepad++打开后，由于无法识别后缀，不知道是何种类型文件，所以无法语法高亮：

A screenshot of the Notepad++ application window. The title bar shows three open files: 'ch01_rec_dev_soft.xml', 'ch01s1_npp.xml', and 'Makefile.mk'. The 'Makefile.mk' tab is selected and highlighted with a red box. The main editing area contains the text of a Makefile, with line numbers 1 through 29 on the left. The text is not syntax-highlighted. The content includes comments like '# Author: crifan', '# Date: 2012-05-16', and project settings such as 'PROJECT_NAME = crifan_rec_soft' and 'SUB_SRC_FILES = \'. It also includes an 'include' statement at the bottom: 'include \$(DOCBOOK_MAKEFILE_ROOT)/docbook.mk'.

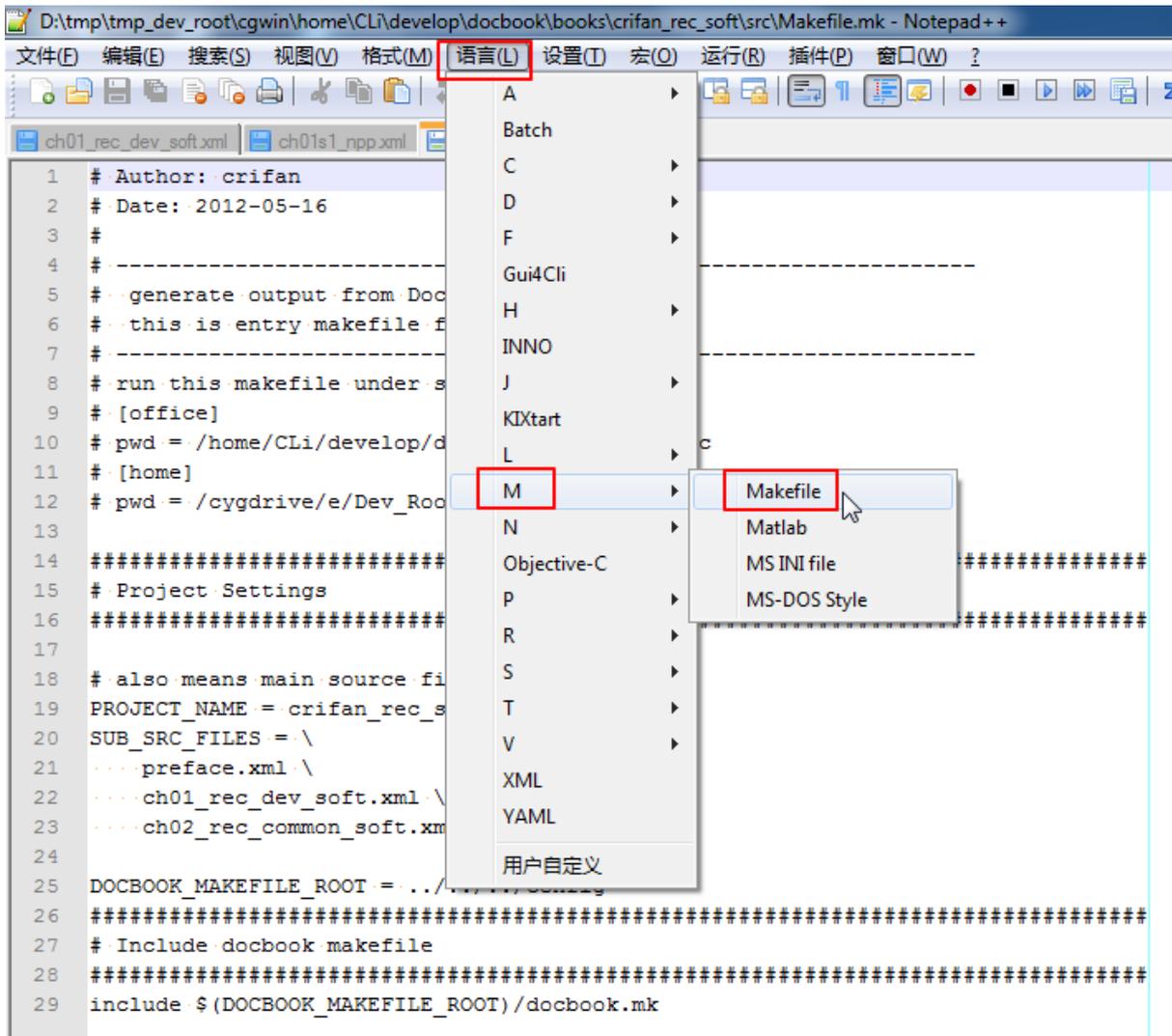
```
1 # Author: crifan
2 # Date: 2012-05-16
3 #
4 # -----
5 # generate output from DocBook sources
6 # this is entry makefile for docbook
7 # -----
8 # run this makefile under src directory:
9 # [office]
10 # pwd = /home/CLi/develop/docbook/books/xxx/src
11 # [home]
12 # pwd = /cygdrive/e/Dev_Root/docbook/dev/books/xxx/src
13
14 #####
15 # Project Settings
16 #####
17
18 # also means main source file name
19 PROJECT_NAME = crifan_rec_soft
20 SUB_SRC_FILES = \
21   \preface.xml \
22   \ch01_rec_dev_soft.xml \
23   \ch02_rec_common_soft.xml \
24
25 DOCBOOK_MAKEFILE_ROOT = ../../../../config
26 #####
27 # Include docbook makefile
28 #####
29 include $(DOCBOOK_MAKEFILE_ROOT)/docbook.mk
```

而如何对不支持的文件类型实现语法高亮，就是下面[第 3.2.2 节 “实现未知类型（不支持的后缀名）的文件的语法高亮”](#)要介绍的内容了。

3.2.2. 实现未知类型（不支持的后缀名）的文件的语法高亮

3.2.2.1. 通过手动设置语言，实现未知类型文件的语法高亮

语言(L)⇒M⇒Makefile#



即可看到语法高亮的效果了：

```

1 # Author: crifan
2 # Date: 2012-05-16
3 #
4 # -----
5 # generate output from DocBook sources
6 # this is entry makefile for docbook
7 # -----
8 # run this makefile under src directory:
9 # [office]
10 # pwd = /home/CLi/develop/docbook/books/xxx/src
11 # [home]
12 # pwd = /cygdrive/e/Dev_Root/docbook/dev/books/xxx/src
13
14 #####
15 # Project Settings
16 #####
17
18 # also means main source file name
19 PROJECT_NAME = crifan_rec_soft
20 SUB_SRC_FILES = \
21   ../preface.xml \
22   ../ch01_rec_dev_soft.xml \
23   ../ch02_rec_common_soft.xml \
24
25 DOCBOOK_MAKEFILE_ROOT = ../../../../config
26 #####
27 # Include docbook makefile
28 #####
29 include $(DOCBOOK_MAKEFILE_ROOT)/docbook.mk

```



注意

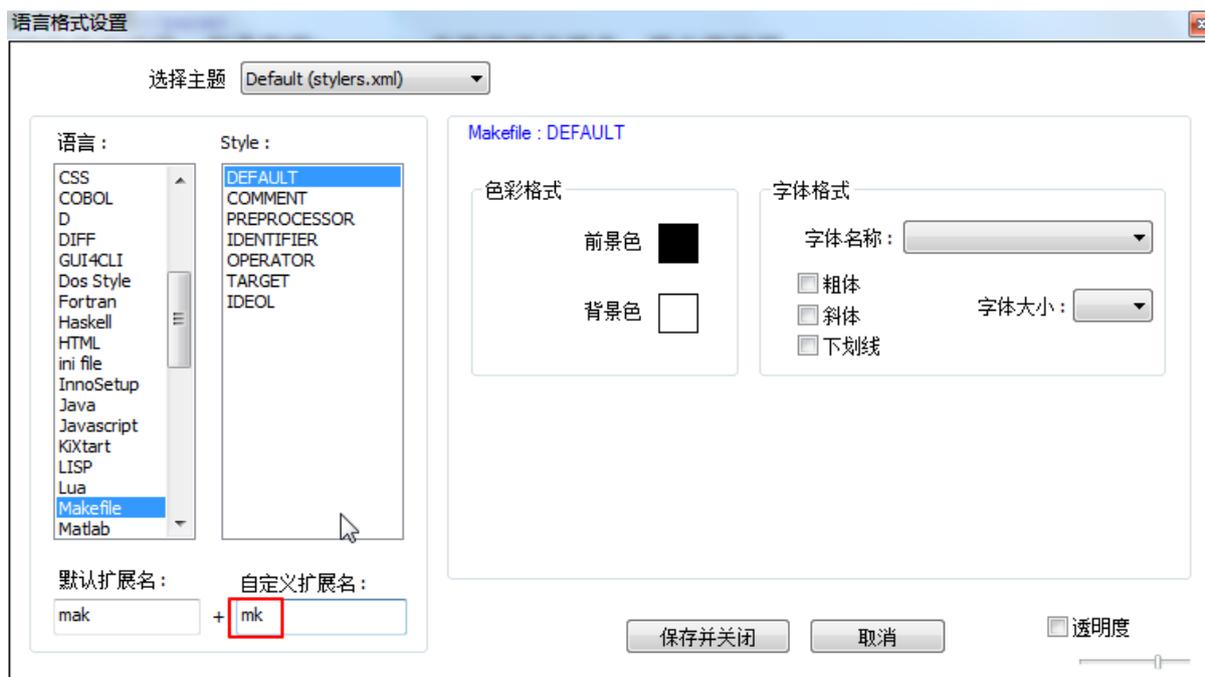
通过手动设置语言的方式实现语法高亮，只对当前打开的文件有效，关闭再打开，就失效了，就需要再手动操作一次，相对比较麻烦。推荐用下面的办法：[第 3.2.2.2 节 “通过添加文件后缀名，实现未知类型文件的语法高亮”](#)

3.2.2.2. 通过添加文件后缀名，实现未知类型文件的语法高亮

设置⇒语言格式设置⇒语言，选择Makefile

可以看到左下角有默认扩展名为mak，意思为，当Notepad++打开mak后缀的文件的时候，就识别为Makefile文件，调用Makefile的语法高亮设置，对此文件进行颜色设置。

对应的，想要支持mk后缀的Makefile文件的话，在自定义扩展名中，加上对应的**mk**



同样可以实现对于后缀名为mk的文件，去实现对应的Makefile的语法高亮的效果。



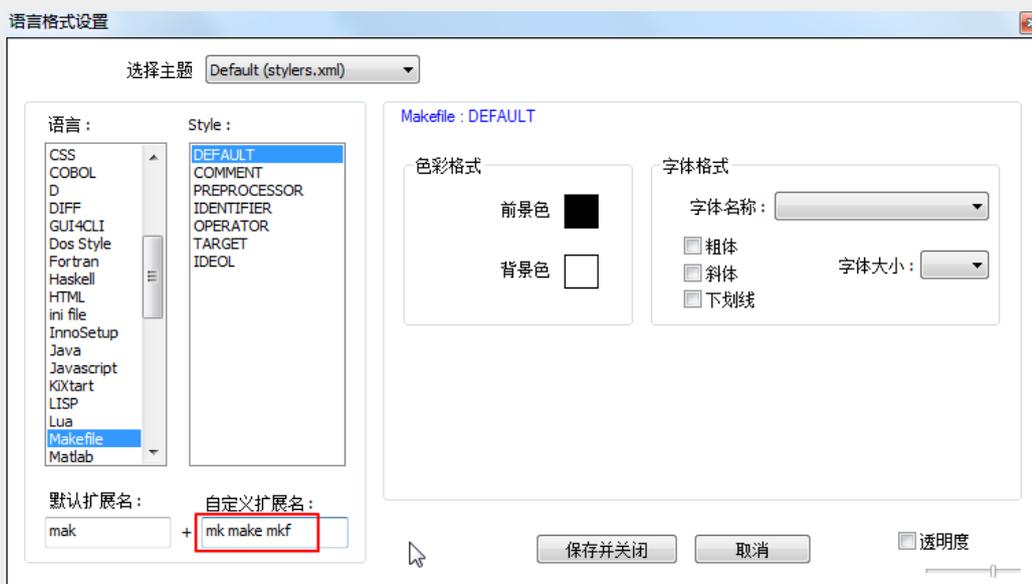
提示

通过添加后缀名的方式，则可以实现之后的每次打开此后缀的文件名，都可以自动识别，并实现语法高亮的效果了。即一劳永逸的效果。



注意

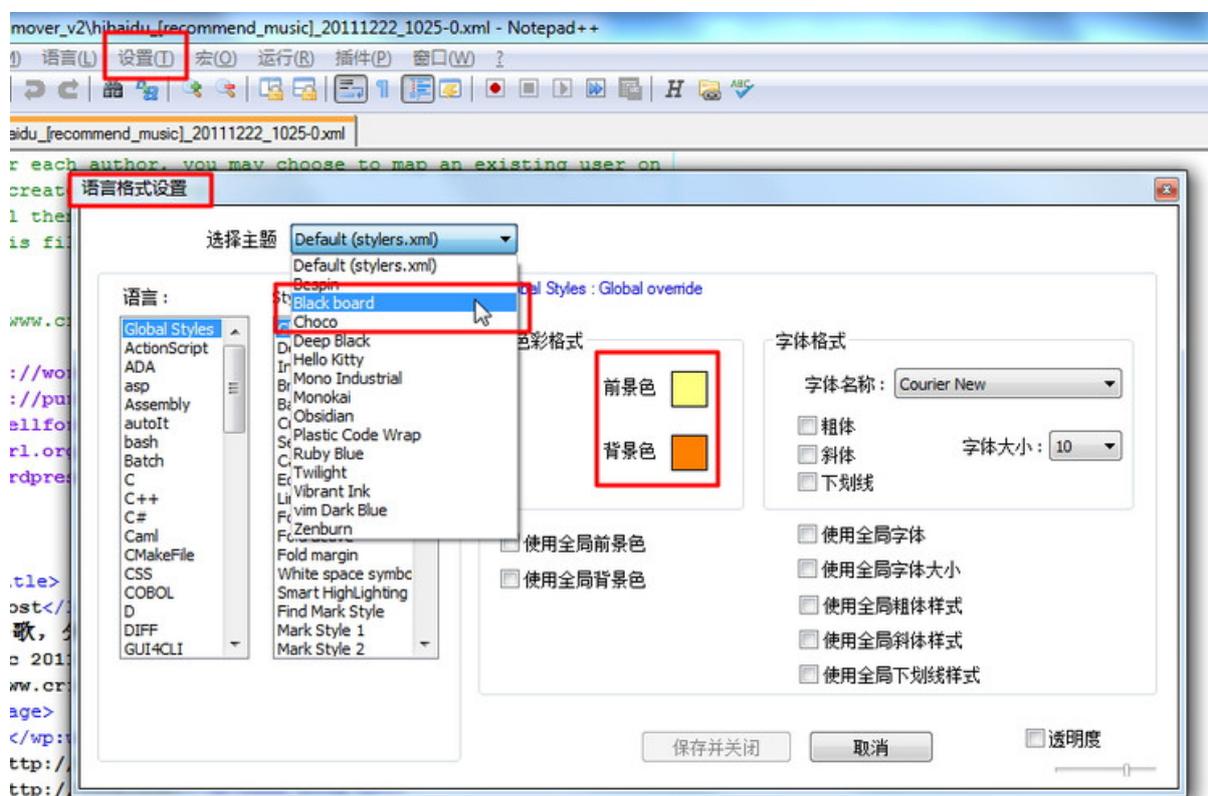
- 不要将mk写成.mk或*.mk
添加自定义扩展名的时候，不需要加"."，也不需要加星号点"*."，否则是无法识别的。
- 想要给自定义扩展名中，添加多个后缀名的话，经过测试，正确做法是：多个文件名后缀之间以空格区分：



而不是（我）原以为的，分号";"或者冒号非分隔符":".

3.2.3. 自定义语法高亮的颜色

对于语法高亮时显示的颜色，也可以自定义：设置⇒语言格式设置：



接下来，就可以根据自己需要配置不同的颜色了。

3.3. Notepad++的多种编码支持



关于字符编码的知识

关于字符编码方面的知识，如果不是很了解的话，建议先去看看[7]

以及相关的中文字符编码：[8]

在对什么字符编码，以及常见的一些字符编码，比如ISO 8859¹，UTF-8²，GB2312³，GBK⁴等，有个最基本的了解之后，我们再来看看Notepad++在字符编码方面，有哪些功能：

3.3.1. 用Notepad++查看当前文件编码

对于想要知道当前文件所用的字符编码类型，可以如下操作：

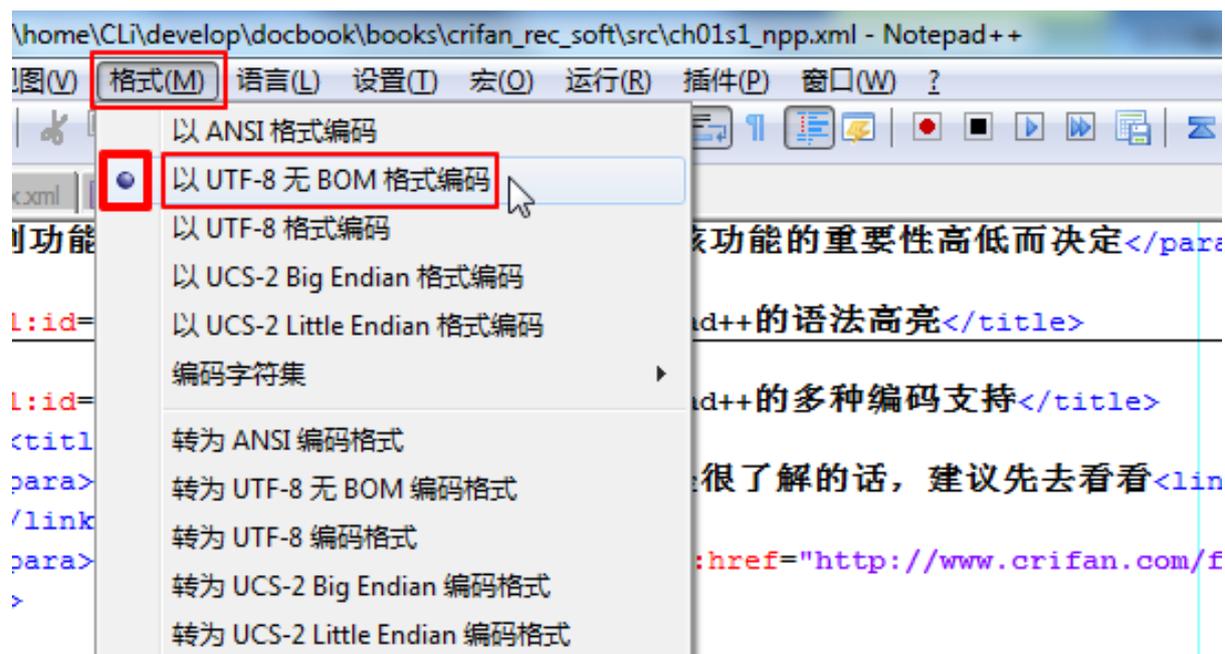
选择 格式(M)，然后就可以看到当前字符编码类型了：

¹ http://www.crifan.com/files/doc/docbook/char_encoding/release/html/char_encoding.html#enc_iso8859

² http://www.crifan.com/files/doc/docbook/char_encoding/release/html/char_encoding.html#enc_utf8

³ http://www.crifan.com/files/doc/docbook/char_encoding/release/html/char_encoding.html#enc_gb2312

⁴ http://www.crifan.com/files/doc/docbook/char_encoding/release/html/char_encoding.html#enc_gbk



在对什么字符编码，以及常见的一些字符编码，比如UTF-8，GBK等，有个

4<title>用Notepad++查看当前文件编码</title>

para>对于想要知道当前文件所用的字符编码类型，可以如下操作： </para

para>选择 格式 (M)，然后就可以看到当前字符编码类型了： </para>

para></para>

可以看到，当前xml文件所用字符编码类型为"以 UTF-8 无 BOM 格式编码"。



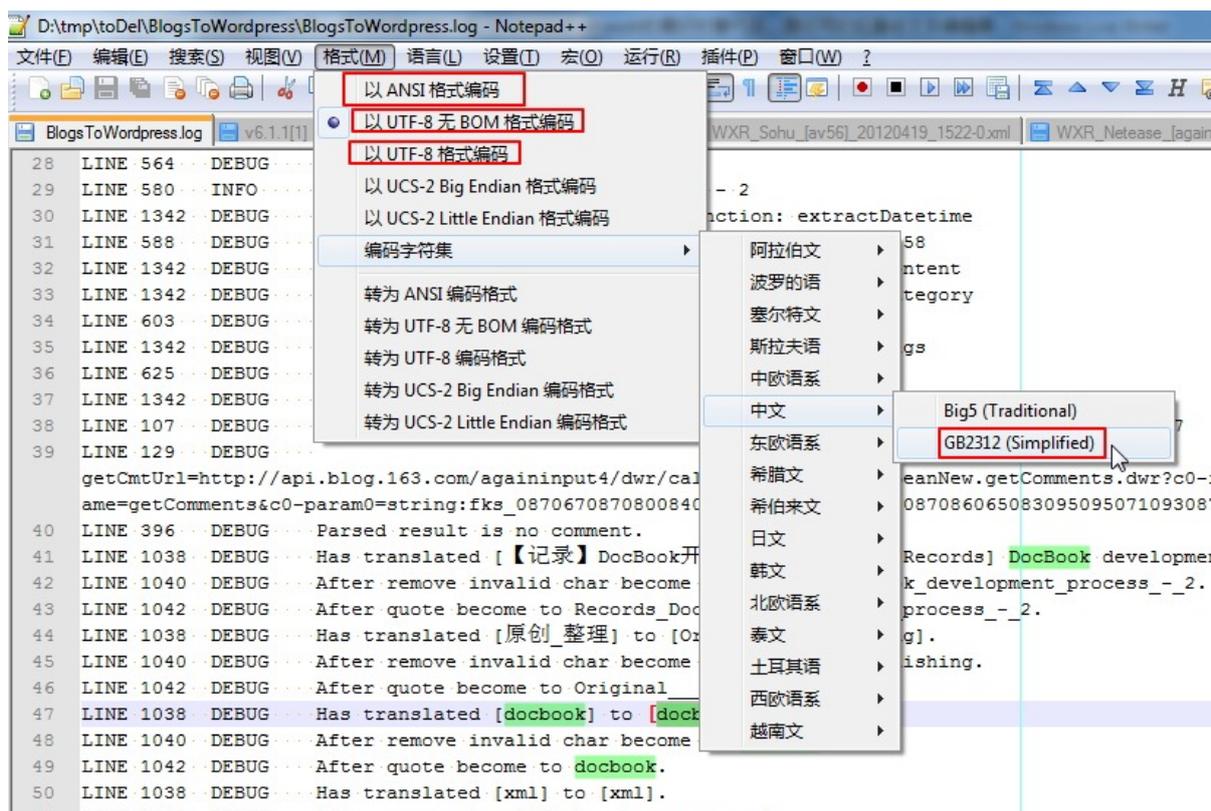
提示

关于UTF-8的BOM，不了解的可参考[\[9\]](#)

需要提及的一些是，一般我们中文和英文，最常用的几个字符编码，大概有：

- 最通用的UTF-8，包含了（全世界几乎）所有的字符
- 双字节的Unicode/UTF-16/UCS-2 LE(Little Endian)
- 简体中文：GB18030 > GBK > GB2312
- 繁体中文：Big5

对应的编码，截图如下：



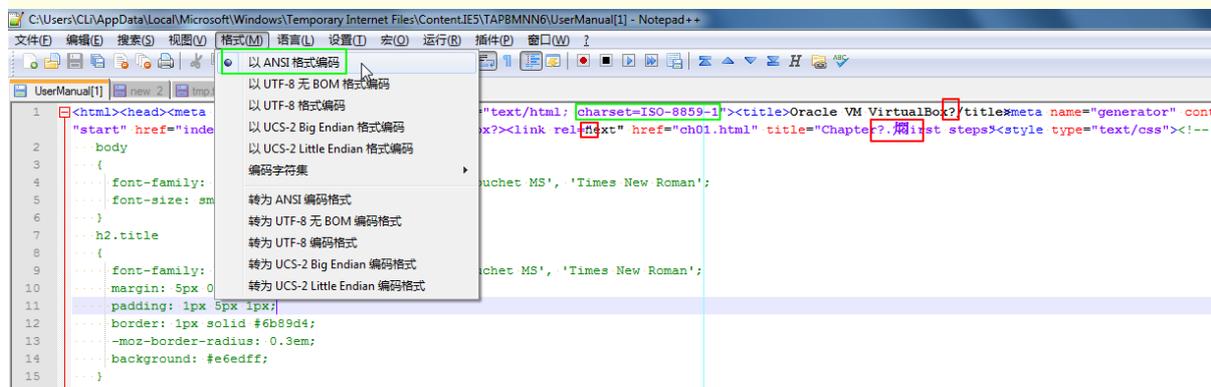
3.3.2. 用Notepad++选用合适的编码打开对应的文件

当你打开某个文件时，可能会遇到一些乱码。

此时，如果知道当前文本本身是用的是何种编码，则可在Notepad++中选择对应编码打开，就可以正确显示文件内容了。

例 3.1. 在Notepad++中使用ISO-8859-1 编码打开VirtualBox的Manual的HTML源码

比如，在打开VirtualBox⁵官网中的用户手册(UserManual)⁶的HTML源码的时候，默认是用ANSI编码打开的，所以会有乱码：



⁵ http://www.crifan.com/category/work_and_job/virtual_machine/virtualbox-virtual_machine/

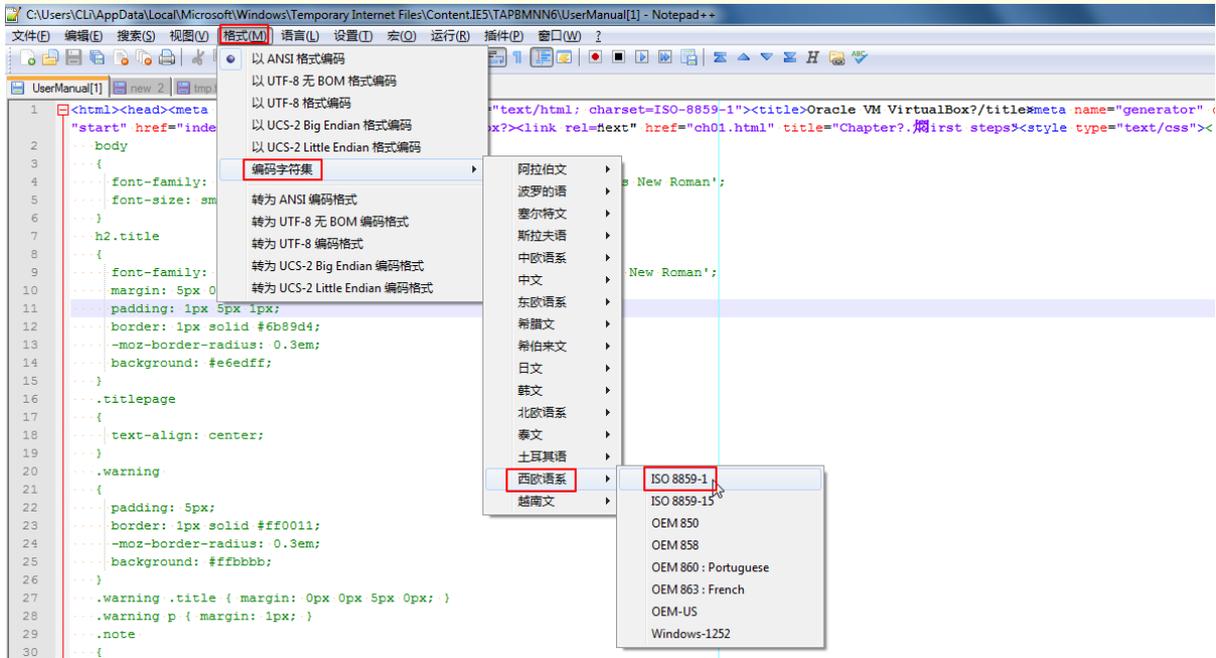
⁶ <https://www.virtualbox.org/manual/UserManual.html>



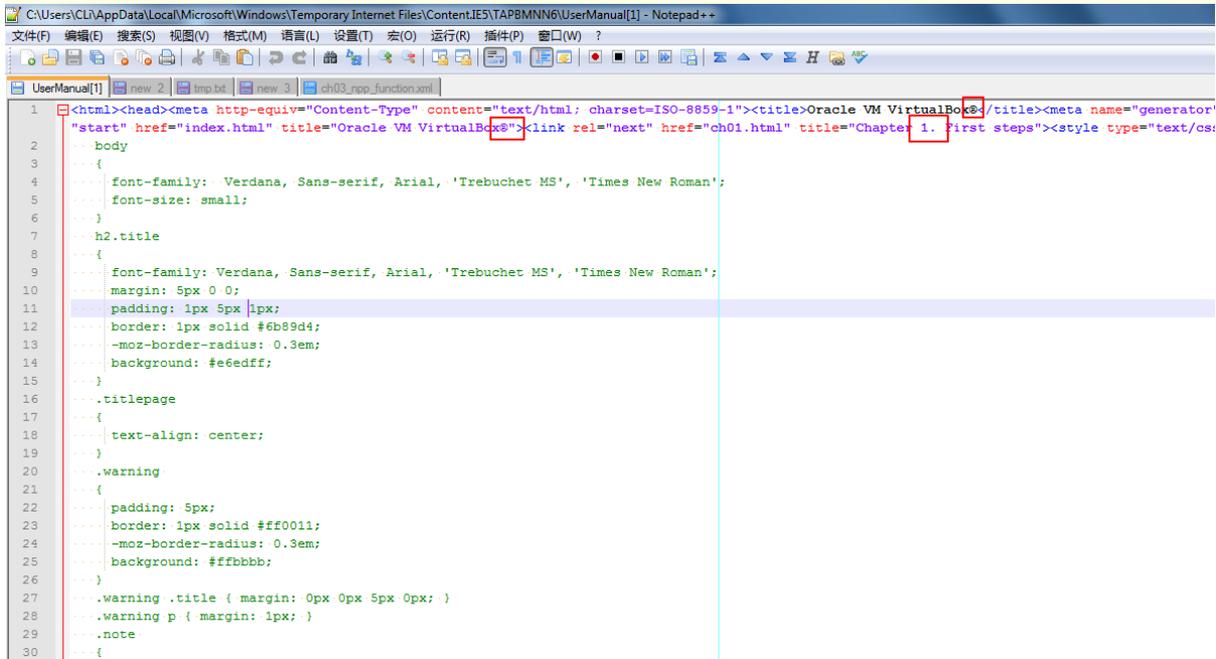
提示

对于如何用Notepad++打开该HTML并高亮显示，可参考第 3.9 节 “Notepad++的默认HTML查看器”

此时，注意到该HTML源码已经通过charset=ISO-8859-1标明了使用的编码是ISO-8859-1了，所以，此时可以去改为对应的ISO-8859-1编码：



就可以看到对应的乱码的字符，都可以正常显示了：



由此，就可以实现了，在Notepad++中，使用正确的编码打开相应的文件，解决了乱码显示问题。



通过html中的charset推测编码

此处由于HTML的特殊性，正常情况下，都会在头部加上charset=xxx来指明当前HTML所用编码

所以此处可以利用此信息去以正确编码显示出内容，而避免了乱码。

对于其他类型文件，在不知道文件编码的情况下，就只能自己手动多试试不同编码了。

最常见的几种，其实也就是那几种，比如UTF-8,ANSI(此处为GBK),GB2312之类的。

3.3.3. 如果乱码则切换到所推测的正确的编码直至消除乱码

如上所述，当HTML源码时，可以通过charset去得知文件编码。

但是，很多时候，我们去打开一个文件时，

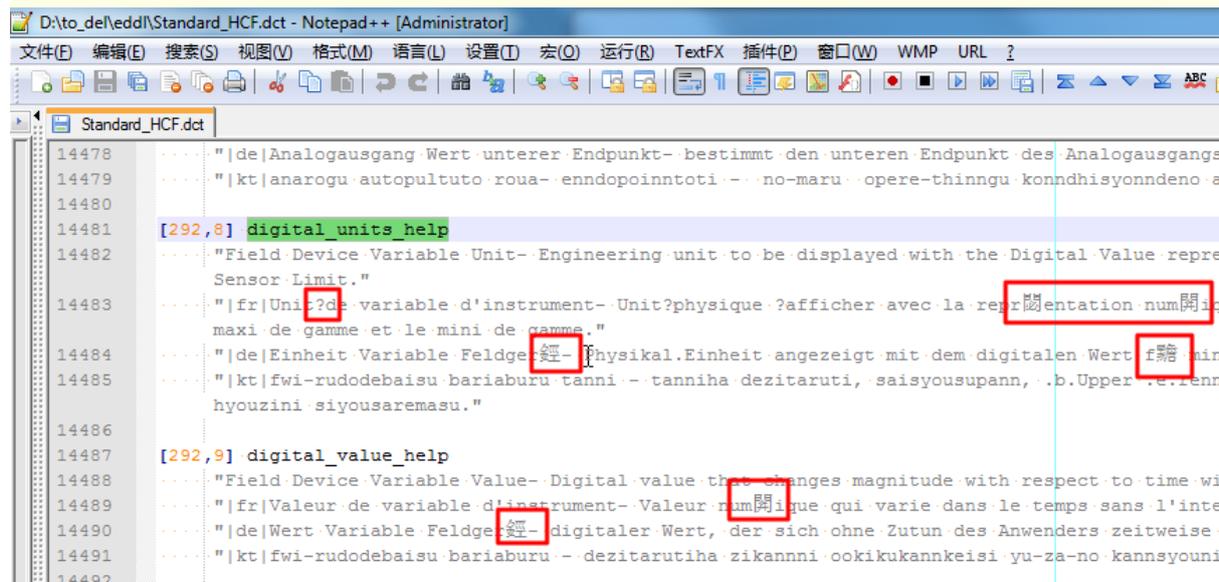
可能会遇到乱码，但是由于未必立刻就已知其文件编码是什么

所以，只能去猜测其编码是什么，然后再切换到对应的编码类型，去查看内容是否可以正常显示。

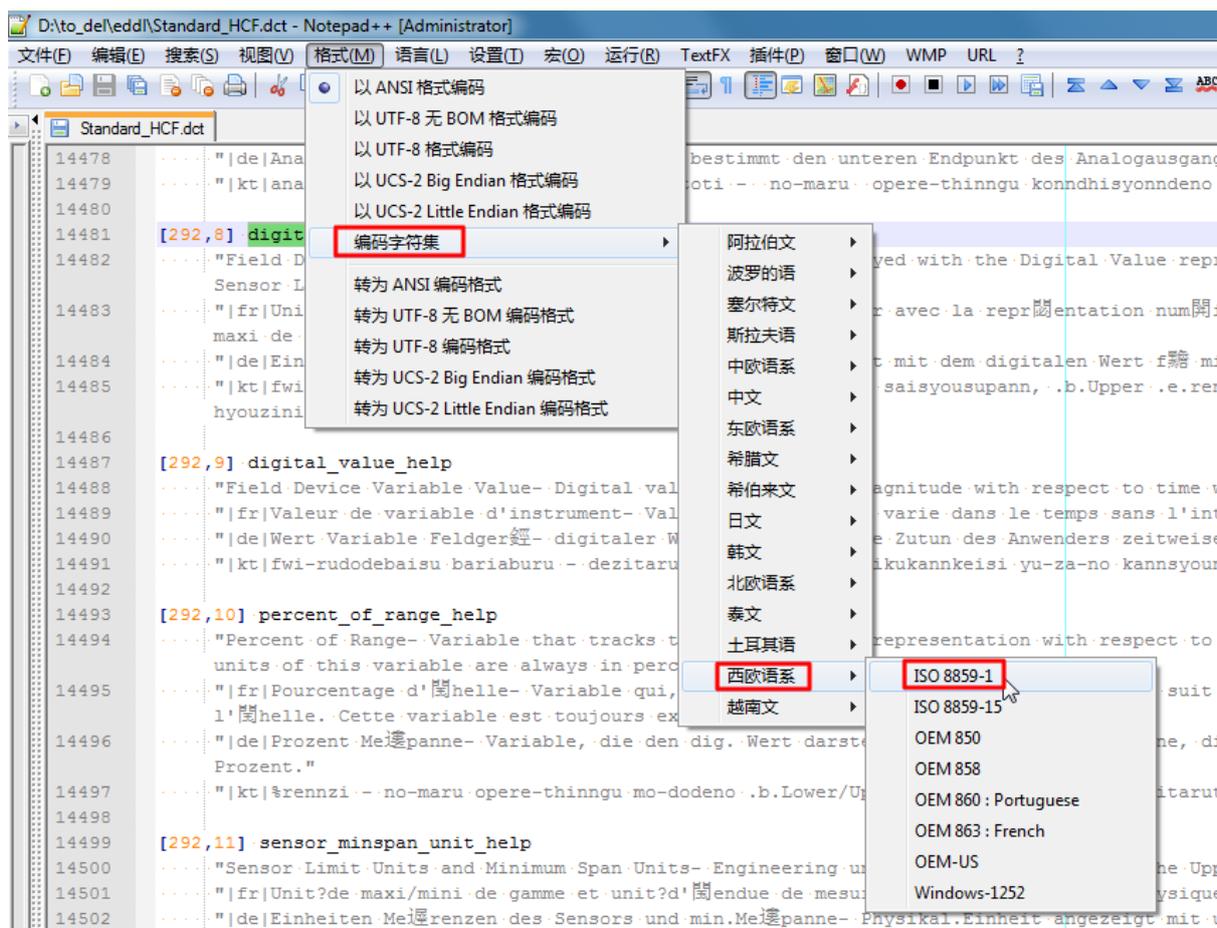
例 3.2. 出现乱码，猜测出是西欧编码，切换到ISO 8859-1而消除乱码

比如遇到一个例子：

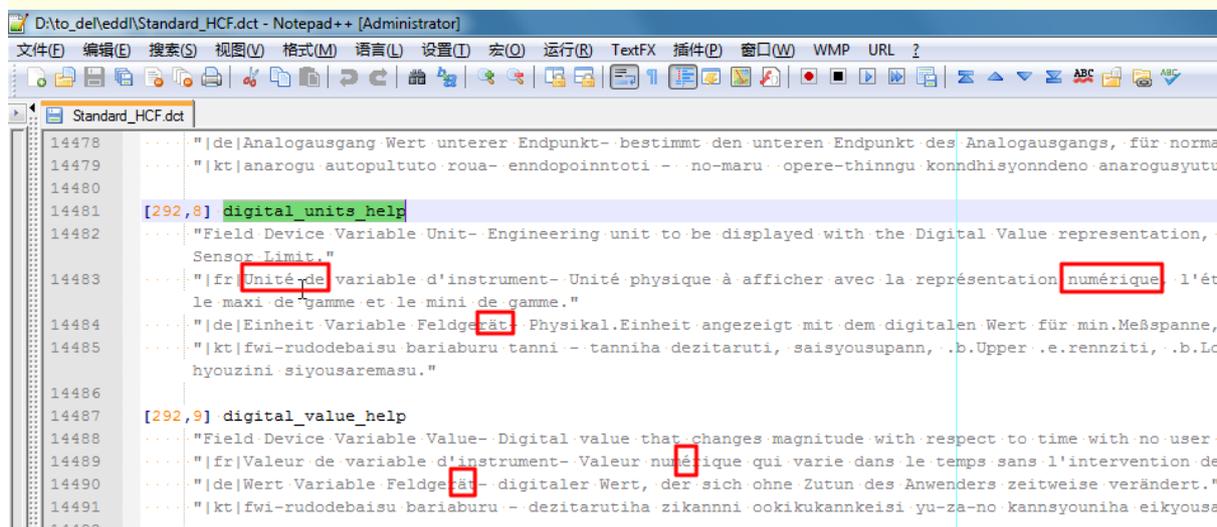
打开文件时，出现是乱码：



看起来，就像是西欧类的字符，所以，去切换到对应的ISO 8859-1编码：



然后真的就消除了乱码，可以正常显示出对应的一些特殊的西欧字符了：



此处，很明显，由于对于编码稍微熟悉，所以一次就猜对了编码，而使得快速消除了乱码。

如果，你对于编码不是很熟悉，则可以多去尝试不同的编码，

最后，肯定也还是可以切换到正确的编码，可以正常显示字符的。

而随着对于字符编码的了解越来越深入，则自然会越加熟悉的，越容易一次或几次就猜对文件的正确的编码的。

3.3.4. 用Notepad++实现不同字符编码之间的转换

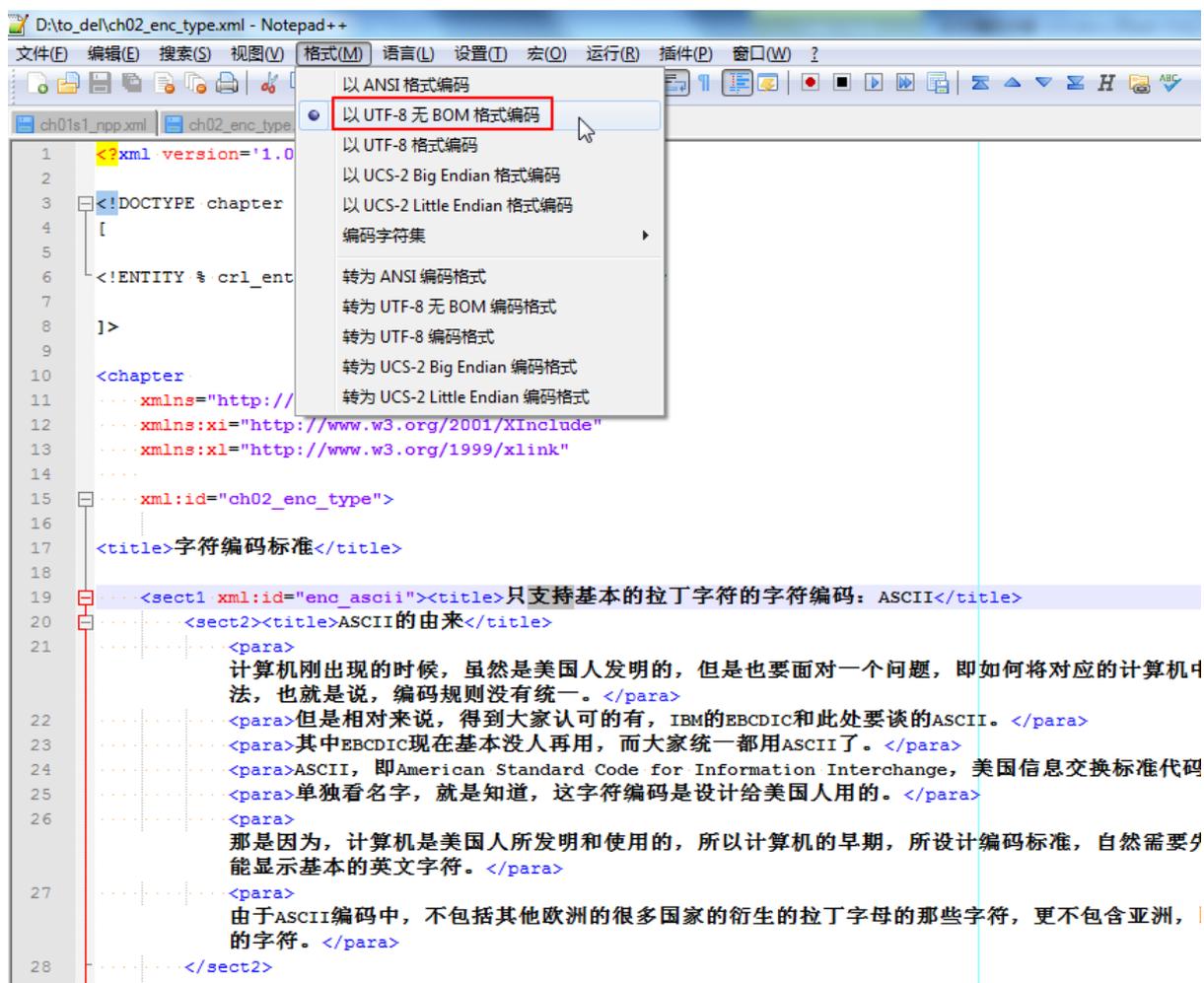
很多时候，我们需要在不同字符编码之间，进行相互转换。

或者由于某些需要，要建立对应的编码的文件。

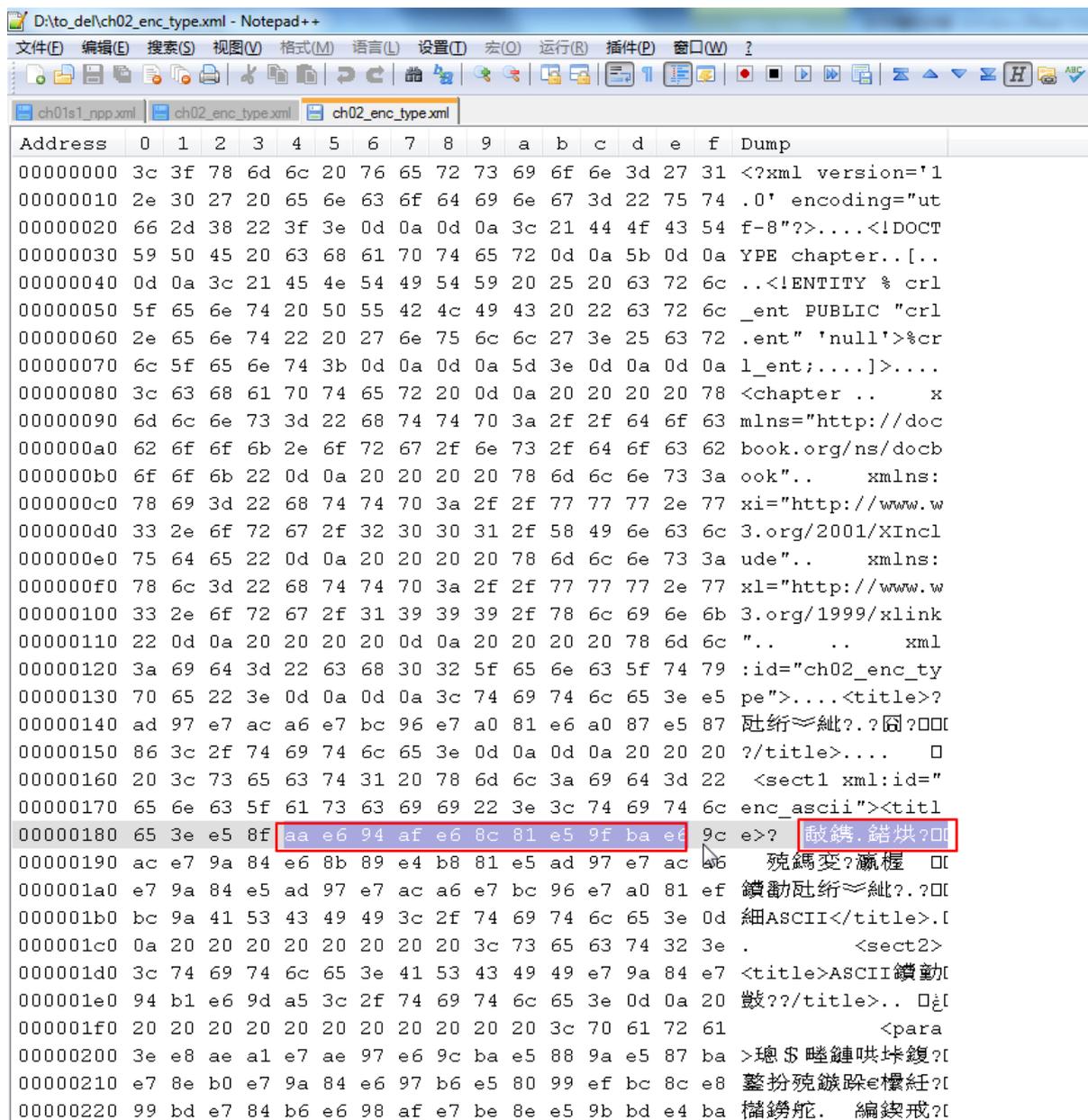
比如，写Python代码的时候，常需要文件本身的编码就是UTF-8的，

此时，就可以用Notepad++的字符编码转换方面的功能了。

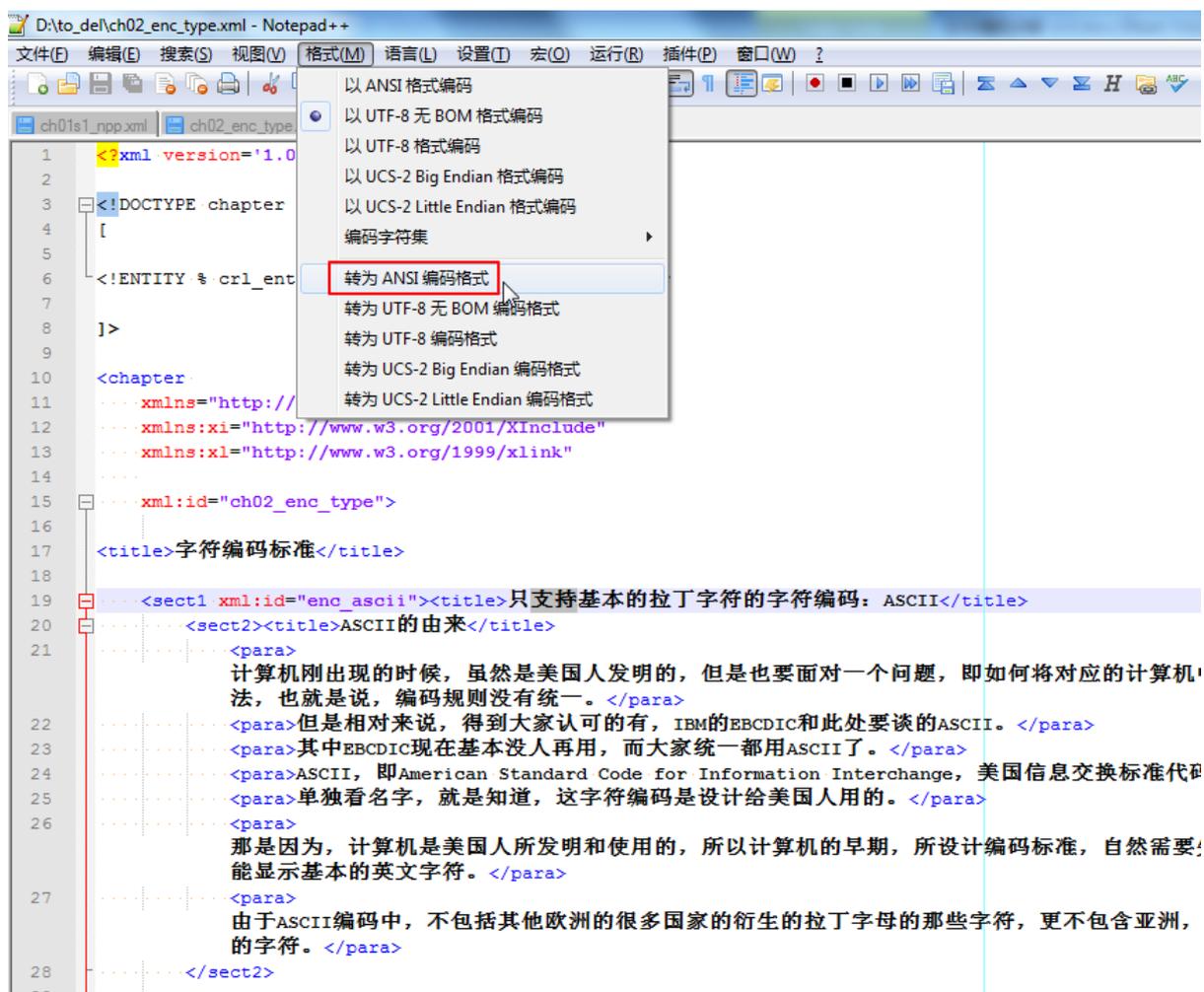
比如，此处将本地一个UTF-8的xml文件：



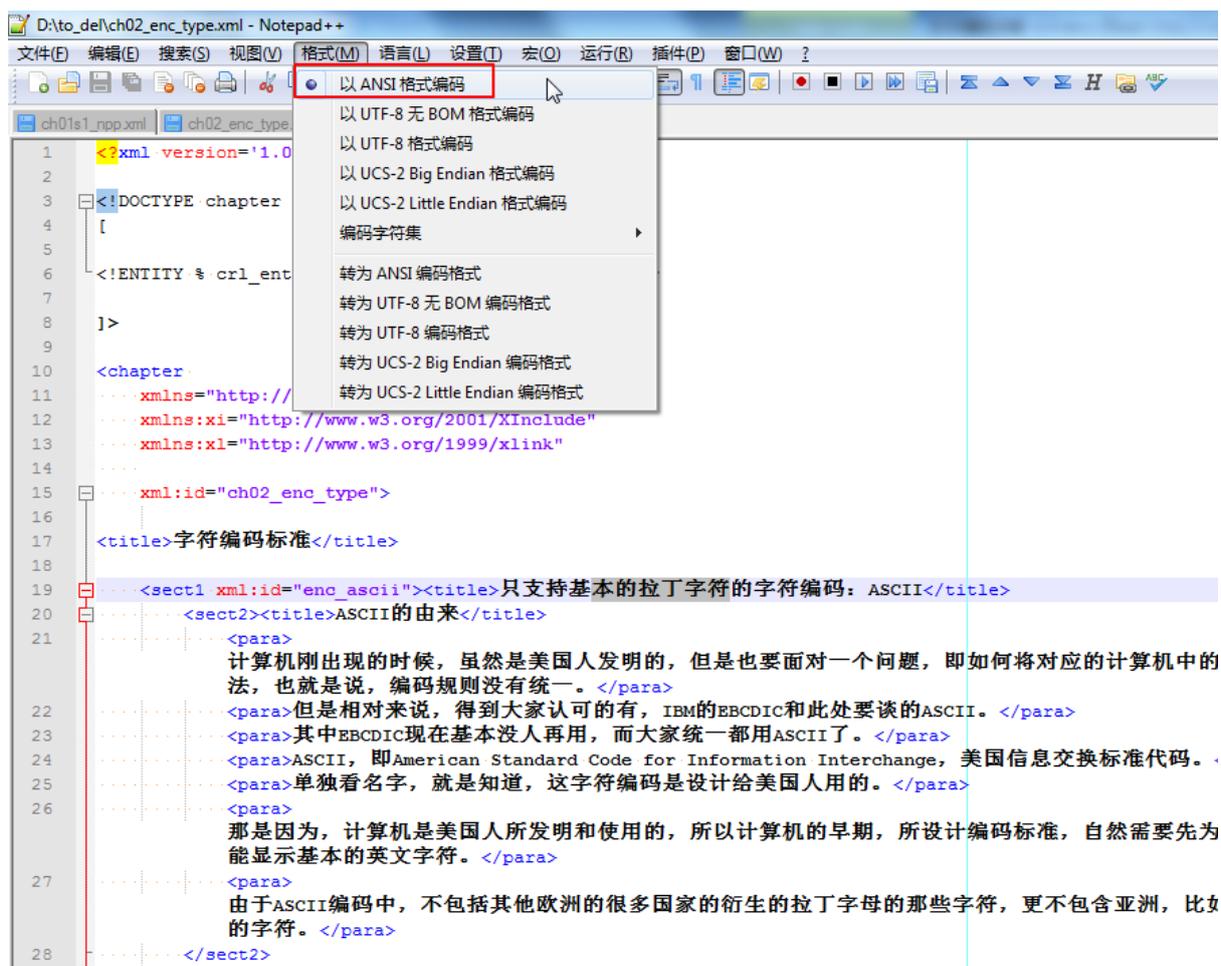
其中，中文字符所对应的16进制的如下：



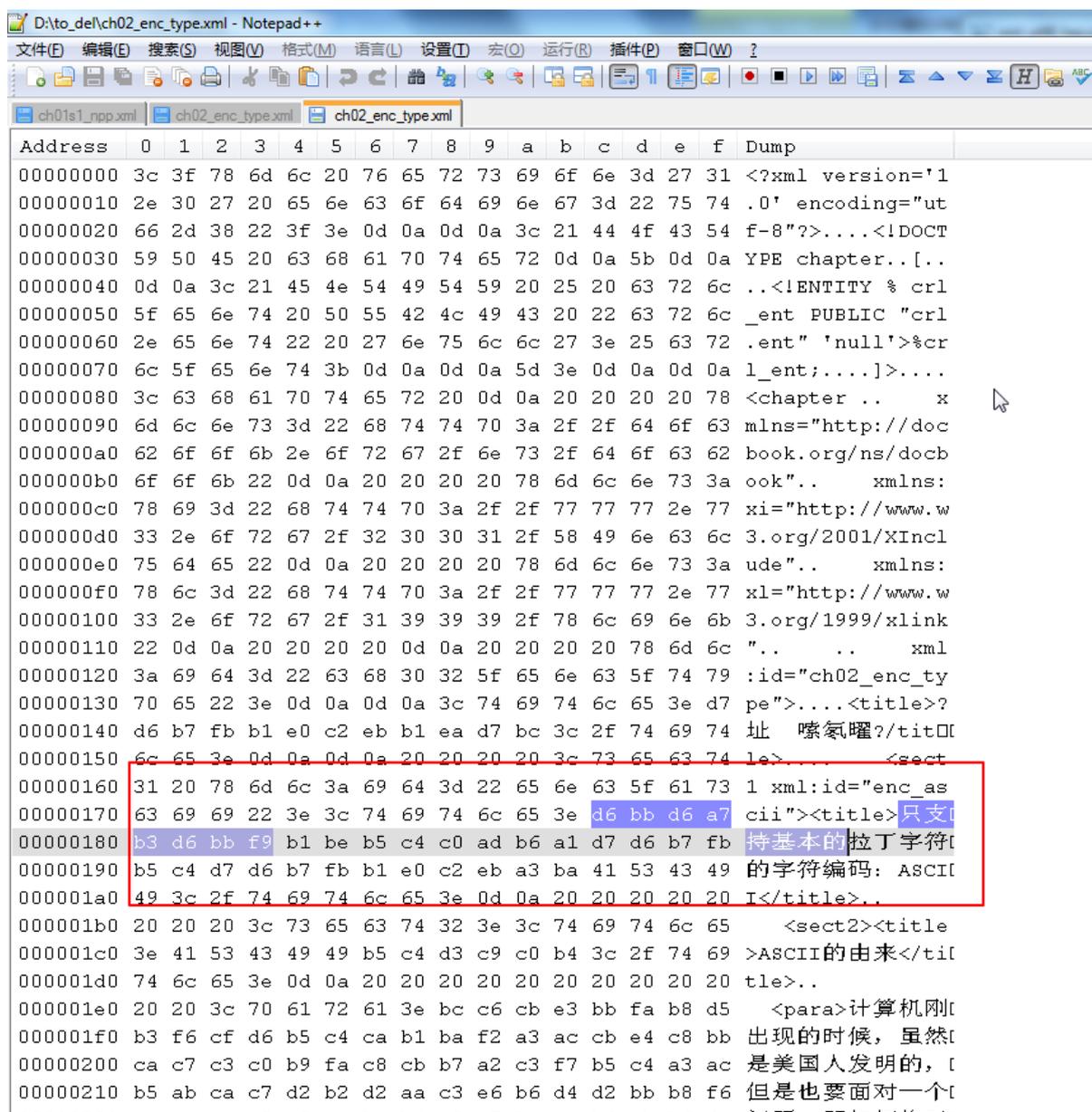
然后点击“转为 ANSI 编码格式”：



即可转为ANSI编码了，此时文件已被修改，所以先保存一下该文件，然后再查看编码，就变为ANSI编码了：



此时再去查看对应的中文字符所对应的16进制的值，就变了：

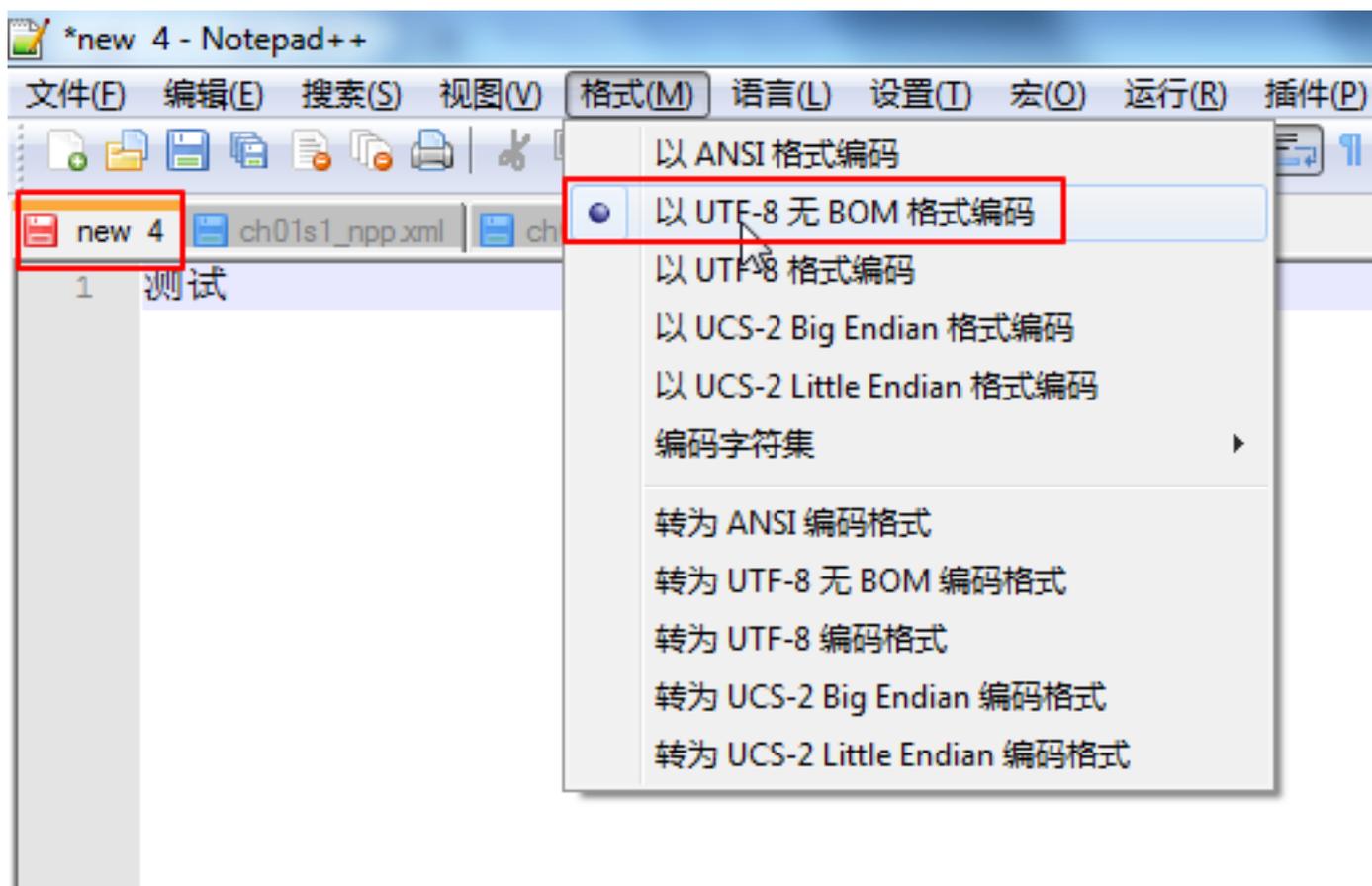


而此处的ANSI编码，可以简单的理解为“本地”编码，而此处是本地编码是中文的GBK，所以此处ANSI即为GBK中文编码。

相应地，可以根据自己需要，在多个不同的字符编码之间互相转换。

3.3.5. 设置Notepad++新建文件（默认）的编码类型

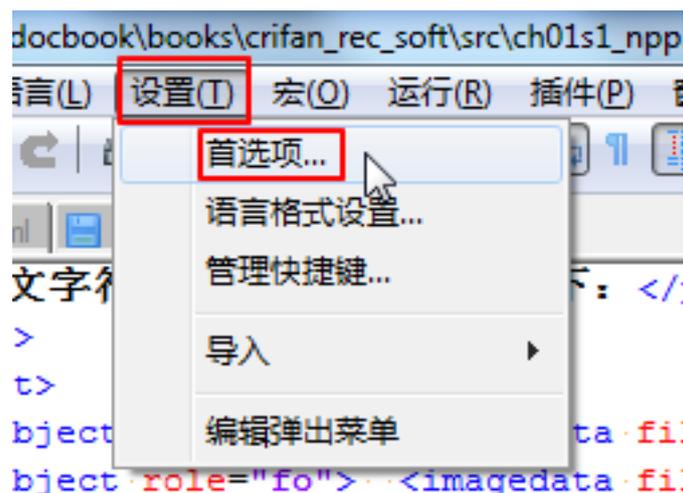
当我们在Notepad++中新建一个文件时，可以通过[第 3.3.1 节 “用Notepad++查看当前文件编码”](#)看到新建的文件所使用的字符编码：



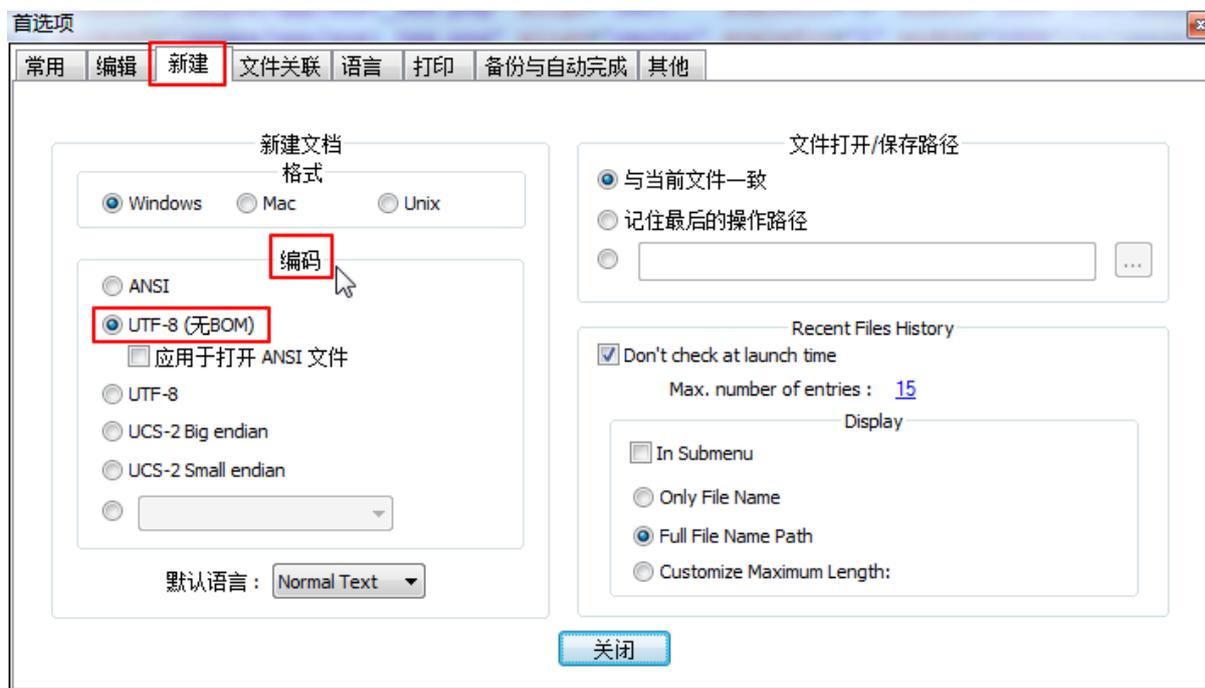
此处为UTF-8。

而想要改变新建文件的默认所用字符编码，可以通过：

设置(T) ⇒ 首选项...



新建，中的“编码”：



然后设置为自己所需要的编码格式。



Notepad++的bug：有时候执行编码转换会导致内容丢失

我遇到过好多次了，在执行代码转换的时候，结果是当前Notepad++打开页面变成空白了，即内容瞬间丢失了，按Ctrl+Z，也无法撤销此操作。

如果不是有备份文件的话，则就会导致文件内容丢失。

如果是很重要的文件的话，文件内容丢失，损失还是很严重的。

尽管Notepad++此编码转换导致文件内容丢失的bug，出现的概率很小，但是也还是建议，对于重要文件的编码转换，转换之前，先备份一下。

3.4. Notepad++的正则表达式替换和替换

下面就来详细介绍Notepad++中关于正则表达式部分，主要是查找和替换。

对于替换功能，一般的文本编辑器，都具有此功能，但是对于高级的正则表达式替换，则很多都不支持。而此处Notepad++支持此功能。

正则表达式的替换，在很长一段时间内，我都没有用到过。而后来有此需求的时候，由于不熟悉，导致也没去折腾具体如何使用的。

后来有空去弄了下，终于搞懂了。对此类功能不了解的人，会没啥感觉，但是看了下面的介绍，你就会发现这类功能的强大之处。

例 3.3. Notepad++正则表达式替换举例：一次性替换多个文件的后缀

举例说明，此处我有个xml文件，其中原始的内容为：

```
<informalfigure>
  <mediaobject>
    <imageobject role="html"><imagedata fileref="images/env_var/win/
right_click_then_property.jpg" align="left" scalefit="0" width="100%"/></imageobject>
```

```

    <imageobject role="fo"><imagedata fileref="images/env_var/win/
right_click_then_property.jpg" align="center" scalefit="1" width="100%"/></imageobject>
  </mediaobject>
</informalfigure>
.....
<informalfigure>
  <mediaobject>
    <imageobject role="html"><imagedata fileref="images/env_var/win/
advance_enviroment.jpg" align="left" scalefit="0" width="100%"/></imageobject>
    <imageobject role="fo"><imagedata fileref="images/env_var/win/
advance_enviroment.jpg" align="center" scalefit="1" width="100%"/></imageobject>
  </mediaobject>
</informalfigure>

```

其中images/env_var/win/xxx.jpg出现的次数，有好几十处。

此处需要把images/env_var/win/xxx.jpg，全部替换为images/env_var/win/xxx.png，即替换文件的后缀。

但是呢，如果手动改的话，改动量很大，效率很低，所以要尽量避免手动改。

另外，此处也不能通过全局的那个替换功能，因为全局替换只适用于固定的文字xxx替换为yyy，而此处文件名都不一样，所以无法实现统一的替换。

此时，才会考虑用Notepad++的正则表达式替换，去实现复杂的，非规则性的替换功能。

而对于正则表达式的替换，最开始，由于不了解其语法，错写成：

查找目标：

```
images/env_var/win/\w+\.jpg
```

替换为(P):

```
images/env_var/win/\w+\.png
```

则替换结果是错的，原来的文件名都被替换为w+这两个字符了：

```

<informalfigure>
  <mediaobject>
    <imageobject role="html"><imagedata fileref="images/env_var/win/w+.png"
align="left" scalefit="0" width="100%"/></imageobject>
    <imageobject role="fo"><imagedata fileref="images/env_var/win/w+.png"
align="center" scalefit="1" width="100%"/></imageobject>
  </mediaobject>
</informalfigure>
.....
<informalfigure>
  <mediaobject>
    <imageobject role="html"><imagedata fileref="images/env_var/win/w+.png"
align="left" scalefit="0" width="100%"/></imageobject>
    <imageobject role="fo"><imagedata fileref="images/env_var/win/w+.png"
align="center" scalefit="1" width="100%"/></imageobject>
  </mediaobject>
</informalfigure>

```

后来，参考了[\[10\]](#)

然后得知是用反斜杠，加上数字，实现向后引用（back reference）。



Notepad++的正则表达式的语法

不过后来也找到了其他更专业和全面的解释：[\[11\]](#)

\digit

Backreference:

\1 matches an additional occurrence of a text matched by an earlier part of the regex. Example: This regular expression: `([Cc][Aa][Ss][Ee]).*` \1 would match a line such as `Case matches Case` but not `Case doesn't match cASE`. A regex can have multiple subgroups, so \2, \3, etc can be used to match others (numbers advance left to right with the opening parenthesis of the group). So \n is a synonym for \gn, but doesn't support the extension syntax for the latter.



注意

之前就知道Notepad++底层是使用SciTE的库的，也顺便找到了SciTE的关于正则表达式的解释[\[12\]](#)

不过其中关于backreference的解释很不清楚。

最后写出正确的语法：

查找目标：

`images/env_var/win/(\w+)\.jpg`

替换为(P):

`images/env_var/win/\1\.png`



可以成功替换为：

```
<informalfigure>
  <mediaobject>
    <imageobject role="html"><imagedata fileref="images/env_var/win/
right_click_then_property.png" align="left" scalefit="0" width="100%"/></imageobject>
    <imageobject role="fo"><imagedata fileref="images/env_var/win/
right_click_then_property.png" align="center" scalefit="1" width="100%"/></imageobject>
  </mediaobject>
</informalfigure>
.....
<informalfigure>
  <mediaobject>
    <imageobject role="html"><imagedata fileref="images/env_var/win/
advance_enviroment.png" align="left" scalefit="0" width="100%"/></imageobject>
    <imageobject role="fo"><imagedata fileref="images/env_var/win/
advance_enviroment.png" align="center" scalefit="1" width="100%"/></imageobject>
  </mediaobject>
</informalfigure>
```

如此，如果有类似需要，想要实现批量的，非规则性的替换，就可以好好利用Notepad++中的正则表达式去替换了。

例 3.4. Notepad++ 正则表达式替换举例：一次性替换多个路径

又比如，由于我把很多jpg，png等类型的图片，从images文件夹移动到了images下面的npp文件夹下了，

所以需要把一个文件中所有的：

images/xxx.yyy

其中xxx为文件名（此处文件名全部都是只包含字母和下划线），yyy=jpg或png，都替换为

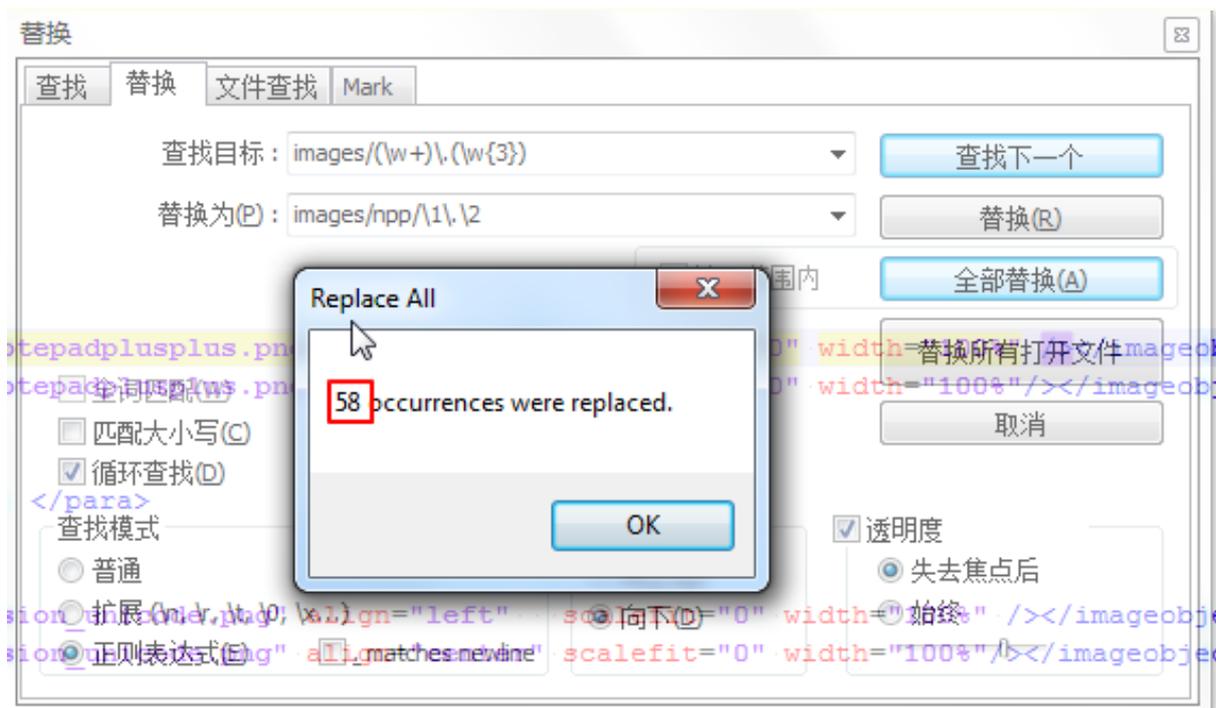
images/npp/xxx.yyy

此时，就可以写成：

images/(w+)\.(w{3})

images/npp^1\.\2

此处提示我成功替换了58处：



相比之下，如果手动去改这58个地方，那真的是累死了不说，还容易由于手误而出错，效率太低。通过正则表达式去替换，则是高效，又准确。

例 3.5. Notepad++ 正则表达式替换举例：一次性替换多个listitem为sect4

又比如，在docbook中，想要把原先的listitem部分的内容，都替换为sect4:



并且也注意到，其中除了上层的listitem，其内部还有一些特殊的子listitem：

```

1048 <listitem>调用搜索引擎搜索选定文字
1049 <para>选择一段文字后，然后：运行(R) &rArr; Google Search Alt+F2</para>
1050 <informalfigure>
1056 <para>就会调用google去搜索选定文字了：</para>
1057 <informalfigure>
1063 <para>与google search的，还有类似的：</para>
1064 <itemizedlist>
1065 <listitem>Get php help Alt+F1: 调用php帮助，对查找php函数等有帮助</listitem>
1066 <listitem>Wikipedia Search :调用维基百科搜索选定文字</listitem>
1067 </itemizedlist>
1068 <para>在同一界面的其他功能有：</para>
1069 <itemizedlist>
1070 <listitem>Open current dir cmd: 打开cmd且定位到当前文件夹，功能等价于：<link xl:href=
      "http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#
1071 <listitem>Open cotaining folder: 打开当前文件所在文件夹</listitem>
1072 </itemizedlist>
1073 </listitem>
    
```

需要在替换的时候，考虑到此点，不要将子listitem也替换掉了。

然后就是去想办法，写对应的正则表达式。

此时，注意到每个要替换的listitem的标题部分之后，是有回车和换行的，所以参考第 3.10 节 “Notepad++支持显示回车符，换行符，TAB键，行首，行尾等特殊字符” 去“显示所有字符”：

```

968 <sect4 xml:id="npp_func_multi_themes"><title>Notepad++的多主题</title>
969 <para></para>
970 <para></para>
971 <para></para>
972 <para></para>
973 </sect4>
974
975 <orderedlist>
976 <listitem>支持双击Tab标签关闭文件
977 <para>想要快速关闭某个文件，可以通过双击该文件的Tab：</para>
978 <informalfigure>
984 <note>
994 </listitem>
995 <listitem>显示文件行号
996 <para>设置(T) &rArr; 首选项... &rArr; 编辑 &rArr; 勾选 "显示行号栏"</para>
997 <informalfigure>
1003 <para>这样，就可以显示行号了：</para>
1004 <informalfigure>
1010 </listitem>
1011 <listitem>支持是否打印行号
1012 <para>设置(T) &rArr; 首选项... &rArr; 打印 &rArr; 勾选 "打印行号"</para>
1013 <para>这样，再去：文件 &rArr; 打印</para>
1014 <informalfigure>
1020 <para>选择一个pdf打印机(BullZip PDF Printer)，然后打印：</para>
1021 <informalfigure>
1027 <para>输出的pdf中，就包括了行号了：</para>
1028 <informalfigure>
1034 <tip>关于pdf打印机，不了解的可参考：<xref linkend="common.bullzip_pdf_printer" /></tip>
1035 </listitem>
1036 <listitem>编辑文件后，是否备份
1037 <para>和Ultra Editor类似，Notepad++在编辑一个文件后，默认会生成.bak文件的。</para>
1038 <para>不过对于notepad++来说，此功能默认是关闭的。需要的话可以去开启：</para>
1039 <para>设置(T) &rArr; 首选项... &rArr; 备份 &rArr; 备份与自动完成 &rArr; 勾选 "简单备份"</para>
1040 <informalfigure>
1046 <para>可见，另外还有禁止和复杂备份，可根据自己的实际情况而设置。</para>
1047 </listitem>
    
```

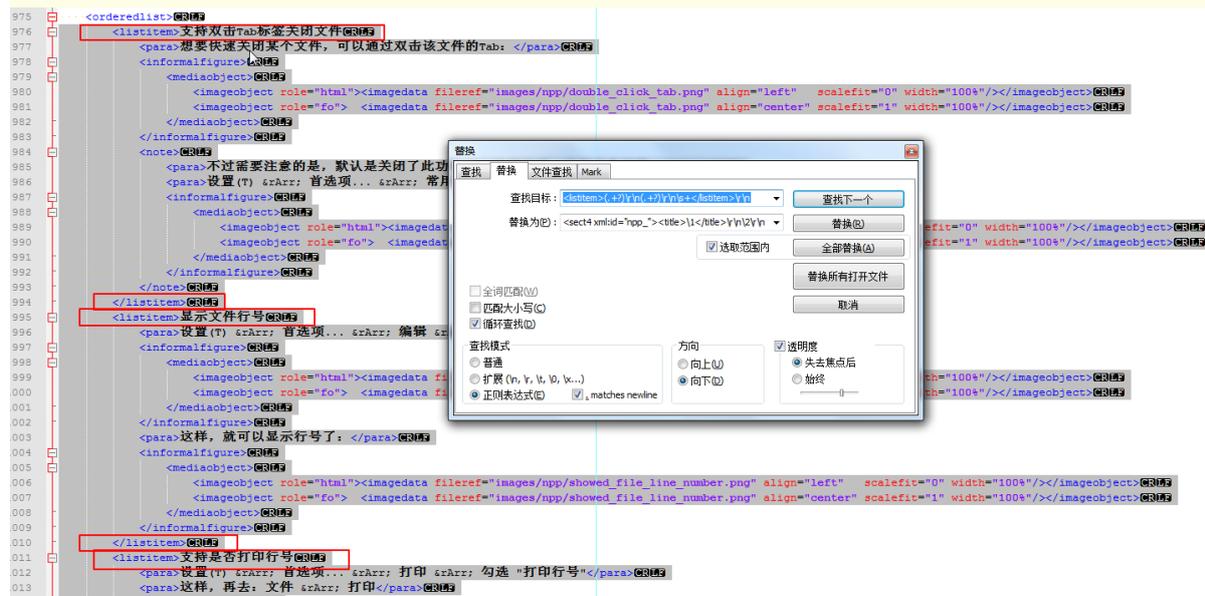
这样就清楚，到底包含哪些字符，方便我们接下来去写正则表达式了。

经过折腾，用如下的正则表达式：

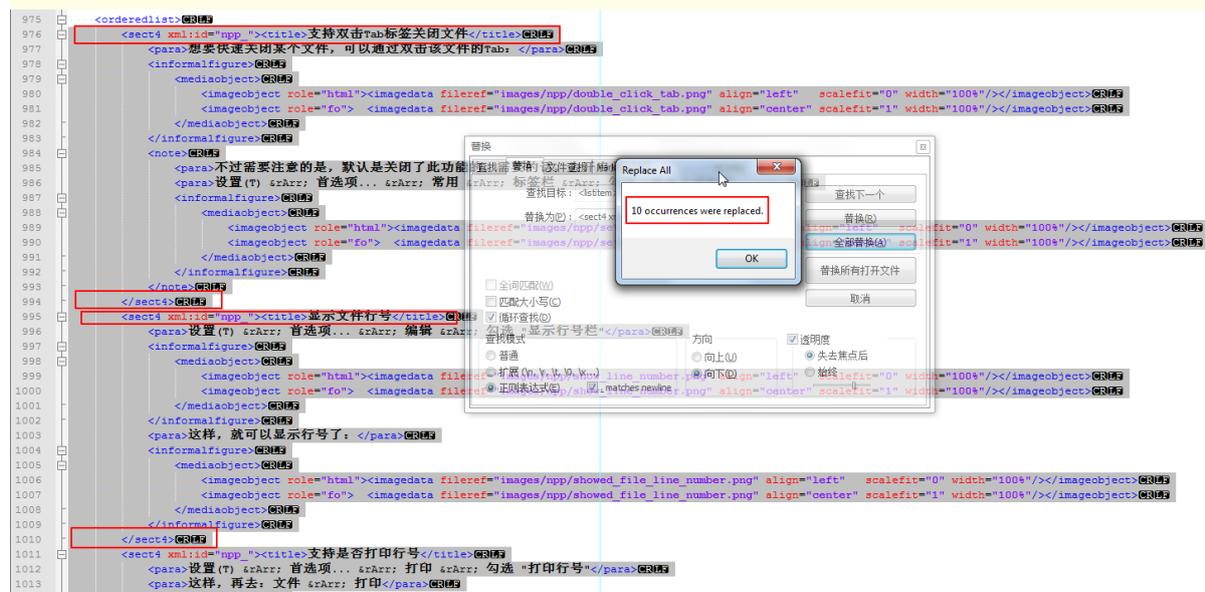
```
<listitem>(.*?)\r\n(.*?)\r\n\s+ </listitem>\r\n
```

```
<sect4 xml:id="npp_" ><title>\1</title>\r\n\r\n </sect4>\r\n
```

将原先内容：



替换为：



可以看到，成功替换了10个。

对应的，也可以看到，对于那些特殊的子listitem来说，也没有被替换掉：

```

1058 <mediaobject>CR1F3
1059 <imageobject role="html"><imagedata fileref="images/npp/call_google_search.png" align="left" sca
1060 <imageobject role="fo"> <imagedata fileref="images/npp/call_google_search.png" align="center" sca
1061 </mediaobject>CR1F3
1062 </informalfigure>CR1F3
1063 <para>与google search的, 还有类似的: </para>CR1F3
1064 <itemizedlist>CR1F3
1065 <listitem>Get php help Alt+F1: 调用php帮助, 对查找php函数等有帮助</listitem>CR1F3
1066 <listitem>Wikipedia Search :调用维基百科搜索选定文字</listitem>CR1F3
1067 </itemizedlist>CR1F3
1068 <para>在同一界面的其他功能有: </para>CR1F3
1069 <itemizedlist>CR1F3
1070 <listitem>Open current dir cmd: 打开cmd且定位到当前文件夹, 功能等价于: <link xl:href=
1071 "http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#right_menu_ru
1072 <listitem>Open containing folder: 打开当前文件所在文件夹</listitem>CR1F3
1073 </itemizedlist>CR1F3
1074 </sect4>CR1F3
1075 <sect4 xml:id="npp"><title>暂时隐藏某些行</title>CR1F3
1076 <para>如果在查看文件的时候, 暂时觉得需要将某些行隐藏掉, 在需要的时候再正常显示, 就可以先选中那些行,
1077 <para>视图(V) &rArr; 隐藏行 Alt+H</para>CR1F3
1078 <informalfigure>CR1F3
1079 <mediaobject>CR1F3
1080 <imageobject role="html"><imagedata fileref="images/npp/hide_line.png" align="left" scalefit="0"
1081 <imageobject role="fo"> <imagedata fileref="images/npp/hide_line.png" align="center" scalefit="1"
1082 </mediaobject>CR1F3

```

如此, 就可以避免了手动的去一点点的修改了。

例 3.6. Notepad++ 正则表达式替换举例：给每一行都添加AddIcon的前缀

需要把每一行中的文件类型的图片名, 替换成加上前缀和后缀。

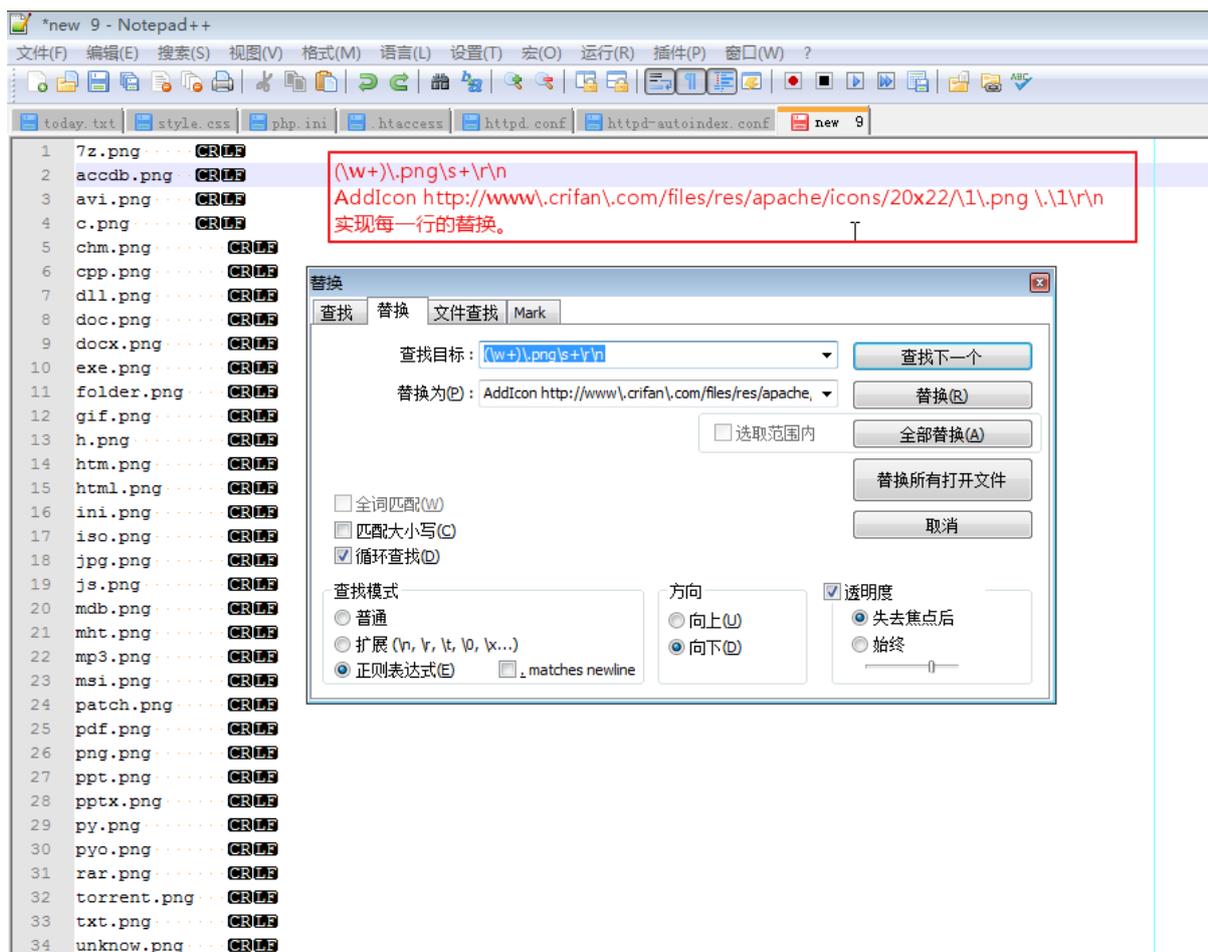
然后就可以去写正则表达式了：

```
(w+)\.png\s+\r\n
```

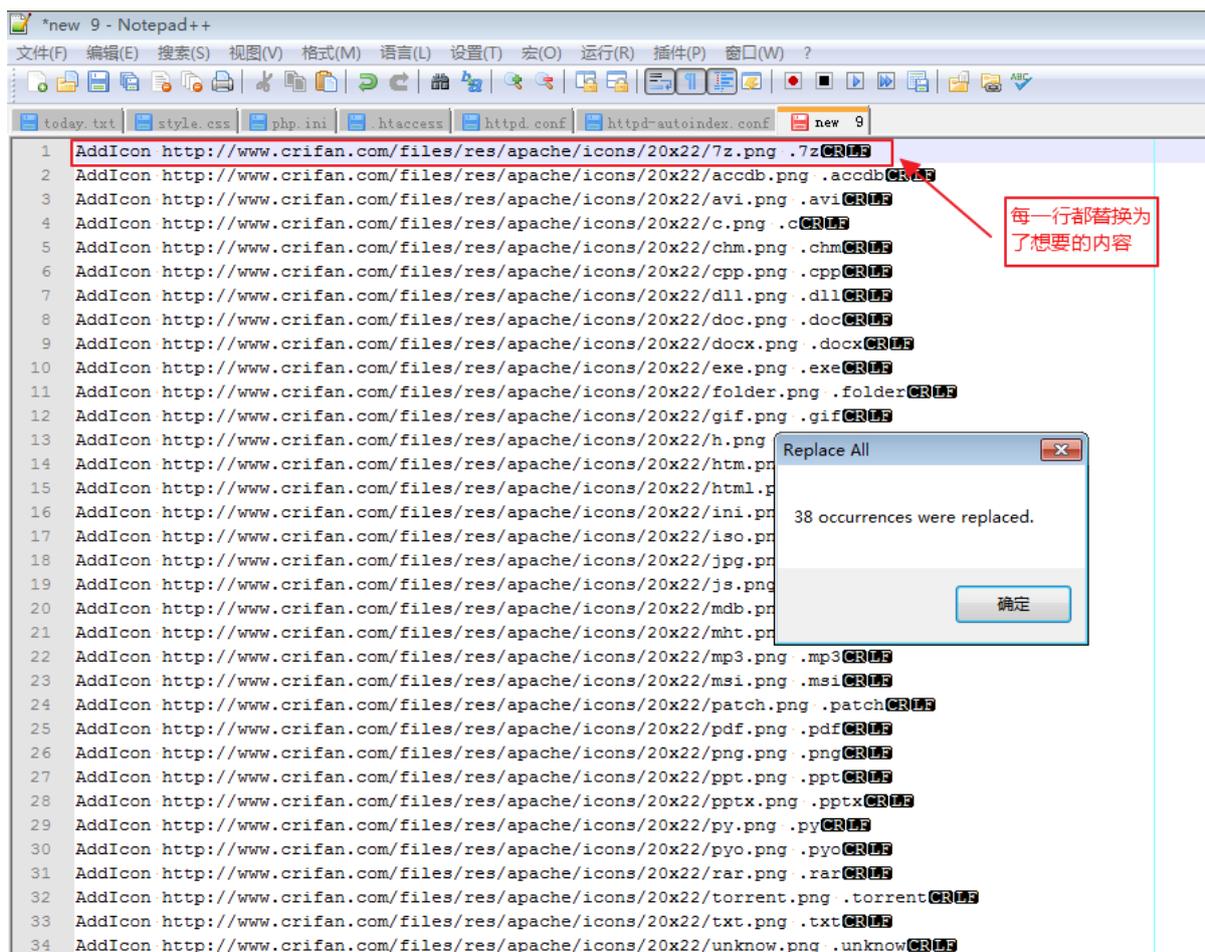
```
AddIcon http://www\.\.crifan\.\.com/files/res/apache/icons/20x22/\1\.\.png \.\.1\r\n
```

就可以从：

Notepad++的功能详解



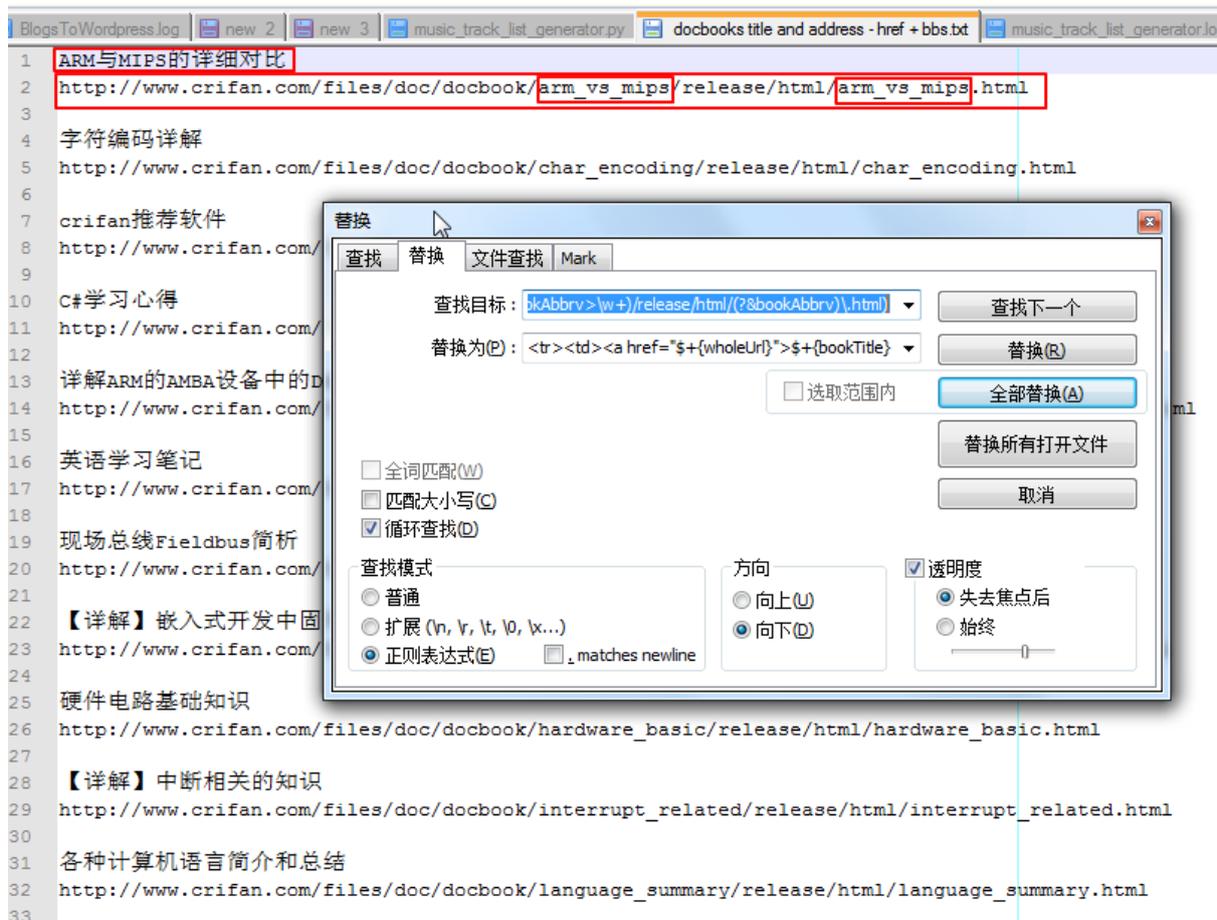
替换为：



所以，如果能利用到此种功能，还是可以很大地提高工作效率的。

例 3.7. Notepad++ 正则表达式替换举例：给book的标题和地址添加html代码

又比如，对于原先是这样的代码：



即，有很多组，每组分别是标题和地址。

而想要做的事情是，把对应的标题和地址，以及其中缩写，都提取出来，并且添加相关的html代码。

最后经过参考：[\[11\]](#)

而写出了相应的正则表达式：

```
(?<bookTitle>.+?)\r\n(?<wholeUrl>http://www.crifan.com/files/doc/docbook/(?<bookAbbrev>\w+)/release/html/(?&bookAbbrev)\.html)
```

```
<tr> <td> <a href="$+{wholeUrl}"> $+{bookTitle}</a> </td> <td> <a href="http://www.crifan.com/bbs/categories/$+{bookAbbrev}"> $+{bookTitle}</td> </tr>
```

然后就可以成功实现28处的替换，替换出来的效果为：

Notepad++的功能详解



替换后的html代码，再添加上相应的html代码：头部的：

```
<table border="1" cellspacing="0" cellpadding="0">
<tbody>
<tr> <td> <strong>单个Book</strong> </td> <td> <strong>对应的讨论区</strong> </td> </tr>
```

和尾部的：

```
</tbody>
</table>
```

最终就可以实现需要显示出来的完整的表格信息了：

单个Book	对应的讨论区
ARM与MIPS的详细对比	ARM与MIPS的详细对比
字符编码详解	字符编码详解
crifan推荐软件	crifan推荐软件
C#学习心得	C#学习心得
详解ARM的AMBA设备中的DMA设备PL08X的Linux驱动	详解ARM的AMBA设备中的DMA设备PL08X的Linux驱动
英语学习笔记	英语学习笔记
现场总线Fieldbus简析	现场总线Fieldbus简析
【详解】嵌入式开发中固件的烧录方式	【详解】嵌入式开发中固件的烧录方式
硬件电路基础知识	硬件电路基础知识
【详解】中断相关的知识	【详解】中断相关的知识
各种计算机语言简介和总结	各种计算机语言简介和总结
【详解】如何编写Linux下Nand Flash驱动	【详解】如何编写Linux下Nand Flash驱动
如何在Linux下写无线网卡的驱动	如何在Linux下写无线网卡的驱动
MPEG简介 + 如何计算CBR和VBR的MP3的播放时间	MPEG简介 + 如何计算CBR和VBR的MP3的播放时间
Linux MTD下获取Nand flash各个参数的过程的详细解析	Linux MTD下获取Nand flash各个参数的过程的详细解析
Python语言总结	Python语言总结
【Crifan推荐】轻量级文本编辑器，Notepad最佳替代品：Notepad++	【Crifan推荐】轻量级文本编辑器，Notepad最佳替代品：Notepad++
正则表达式学习心得	正则表达式学习心得
在Linux运行期间升级Linux系统 (Uboot+kernel+Rootfs)	在Linux运行期间升级Linux系统 (Uboot+kernel+Rootfs)
软件开发基础知识	软件开发基础知识
Code 128 Symbology Introduction	Code 128 Symbology Introduction
GS1-128条形码和相关的AI及FNC1的详解	GS1-128条形码和相关的AI及FNC1的详解
Plessey & MSI Symbology Introduction	Plessey & MSI Symbology Introduction
UPC/UPC-A/UPC-E & EAN Barcode Symbology	UPC/UPC-A/UPC-E & EAN Barcode Symbology
Uboot中start.S源码的指令级的详尽解析	Uboot中start.S源码的指令级的详尽解析
USB基础知识概论	USB基础知识概论
如何实现Linux下的U盘 (USB Mass Storage) 驱动	如何实现Linux下的U盘 (USB Mass Storage) 驱动
USB HID Learning Record	USB HID Learning Record

则又一次地，极大地提高了工作效率。否则要一个个的复制和粘贴，累死了不说，也还容易出错。

例 3.8. Notepad++ 正则表达式替换举例：查找单个的CR或LF

后来一次，遇到一个需求是：对于单个文件中，查找所谓的不一致的换行符

去看了一下当前文件中，正常都是Windows类型的回车换行，CRLF

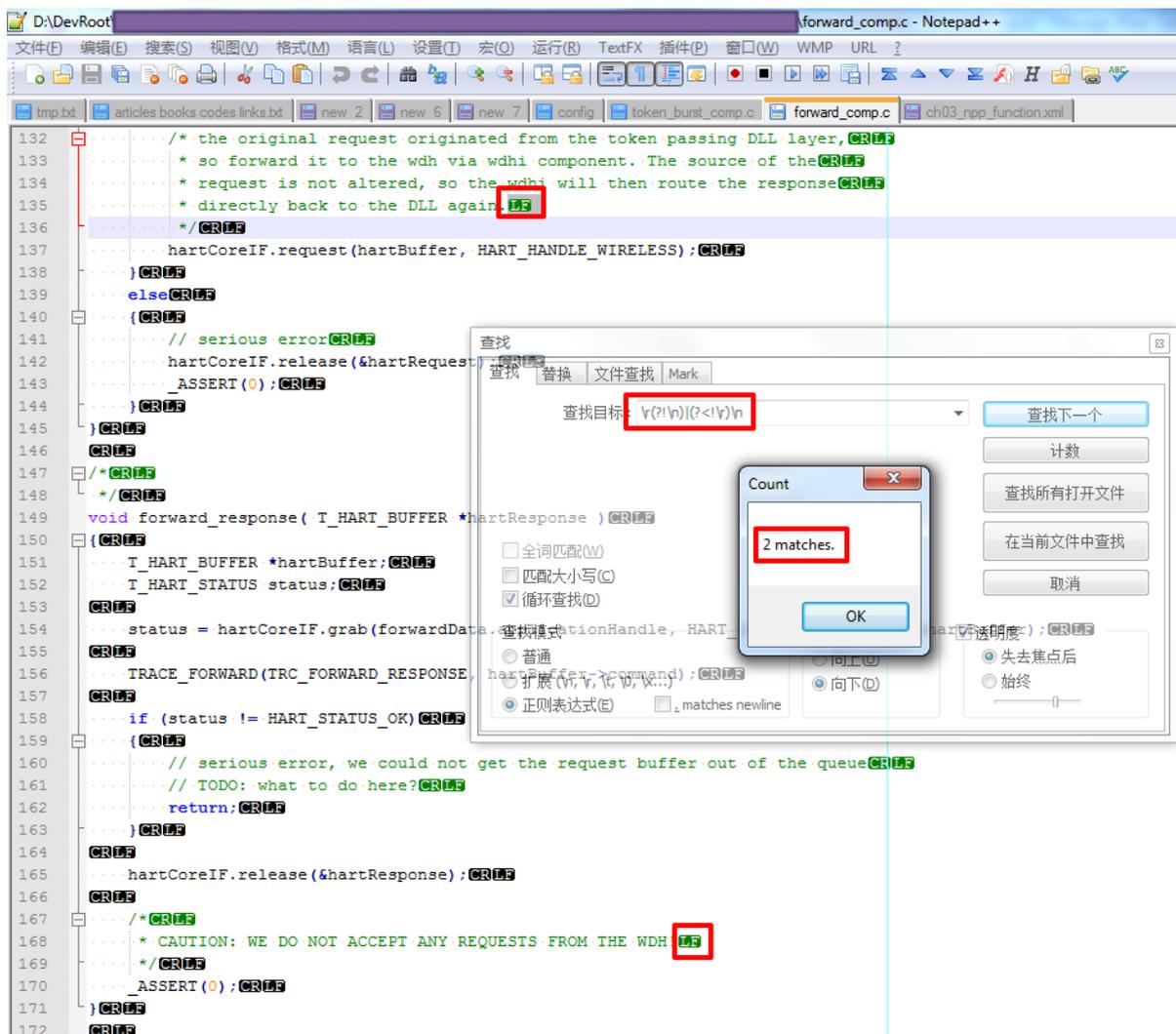
想要找到哪里出现了单个的CR或LF

而由于文件有很多行，里面有N个CRLF，而要一眼看出哪里有单个的CR或LF，很困难，或者说不可能，所以只能靠正则表达式去搜索寻找了

然后最终使用正则表达式是：

```
\r(?!\n)|(?!r)\n
```

然后找到了对应的2处，单个的CR或者LF：



此处，顺便简单解释一下此处的正则表达式的含义：

- 具体的语法，还是去参考这个最全的语法：[\[11\]](#)
- 关于CR LF不清楚的，还是先去看：[\[13\]](#)
- 此处的

`\r(?:\n)|(?:<\r)\n`

是xxx|yyy的格式，表示匹配xxx或yyy

- 其中xxx是

`\r(?:\n)`

，yyy是

`(?:<\r)\n`

- 其中\r和\n，分别表示回车CR，换行LF
- 而

`\r(?:\n)`

用于匹配一个\r，但是后面不是\n

- 而

`(?<!\r)\n`

用于匹配一个\n，但是前面不是\r

例 3.9. Notepad++ 正则表达式替换举例：去除href链接

遇到一个问题是，想要把一个表格中的内容，粘贴到WLW中，结果由于其中N个内容，都包含了对应的链接，想要把链接去掉。

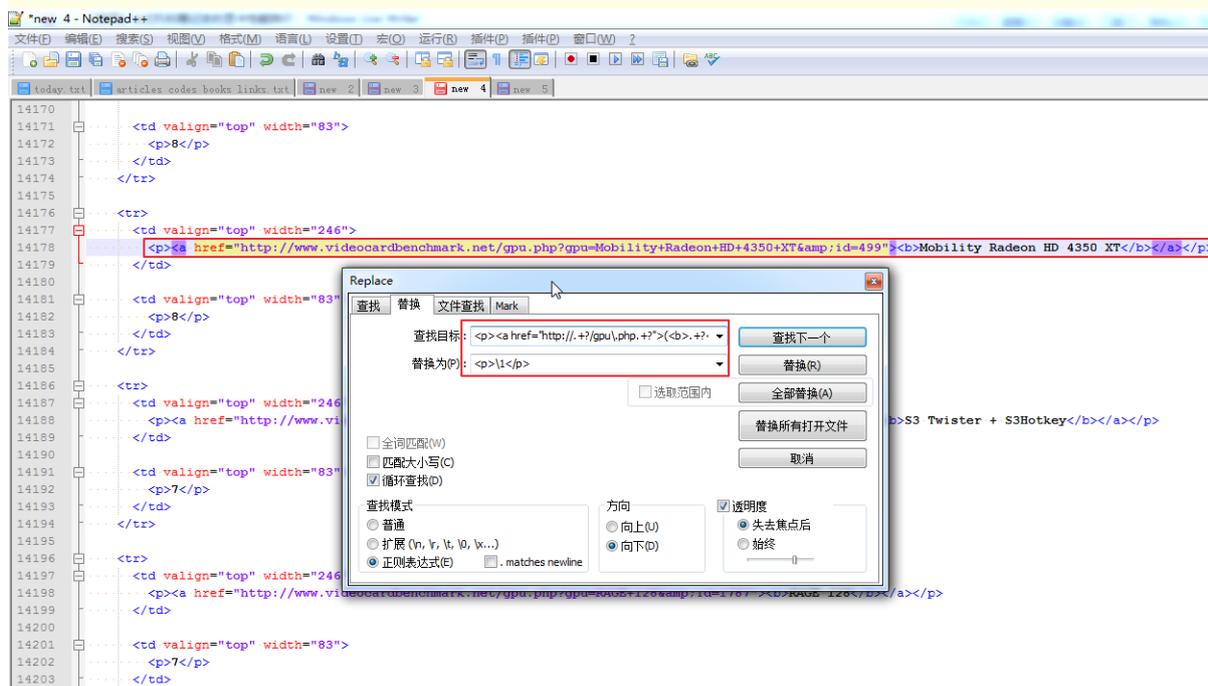
所以就去切换到源码模式，然后把html代码拷贝出来，然后用Notepad++去替换

所用正则则是：

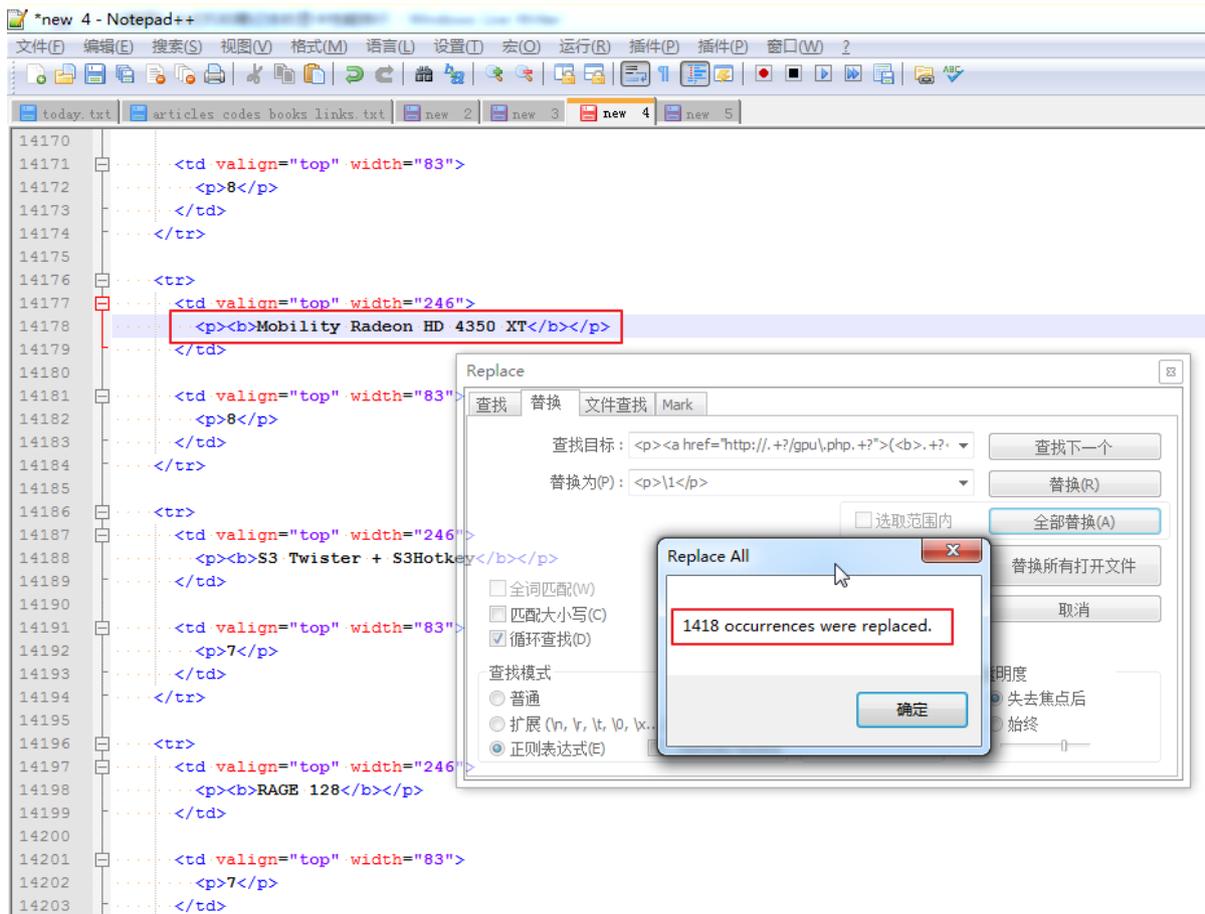
`<p> (.+?) </p>`

`<p>\1</p>`

替换前是：



替换后，变成这样的：



效率不是一般的高。

例 3.10. Notepad++ 正则表达式替换举例：把标题和地址转换为link格式

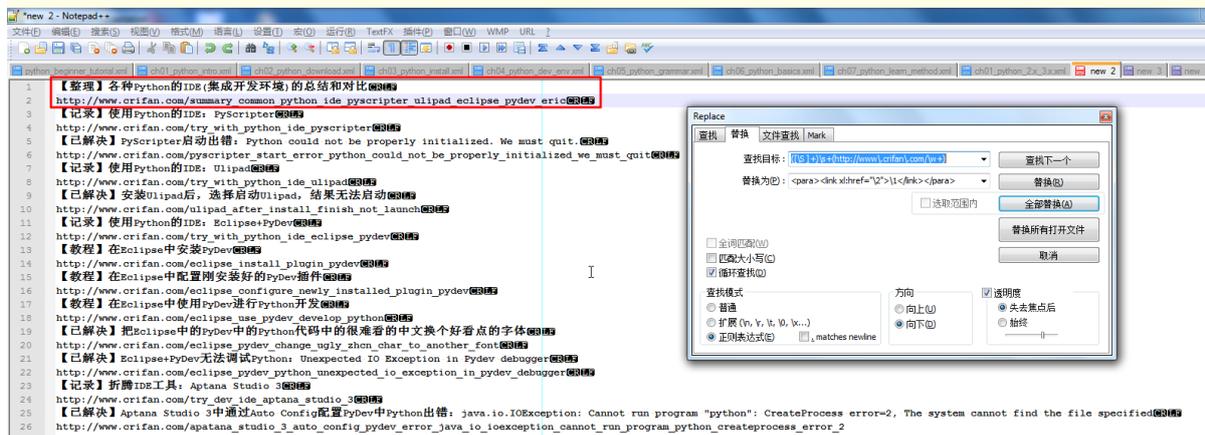
遇到一个问题是，想要把对应的，帖子的标题和地址，变成docbook中的link格式，一般拷贝到Docbook的xml中直接使用，就省的自己一点点复制粘贴和修改了。

所用Notepad++的正则替换的如下：

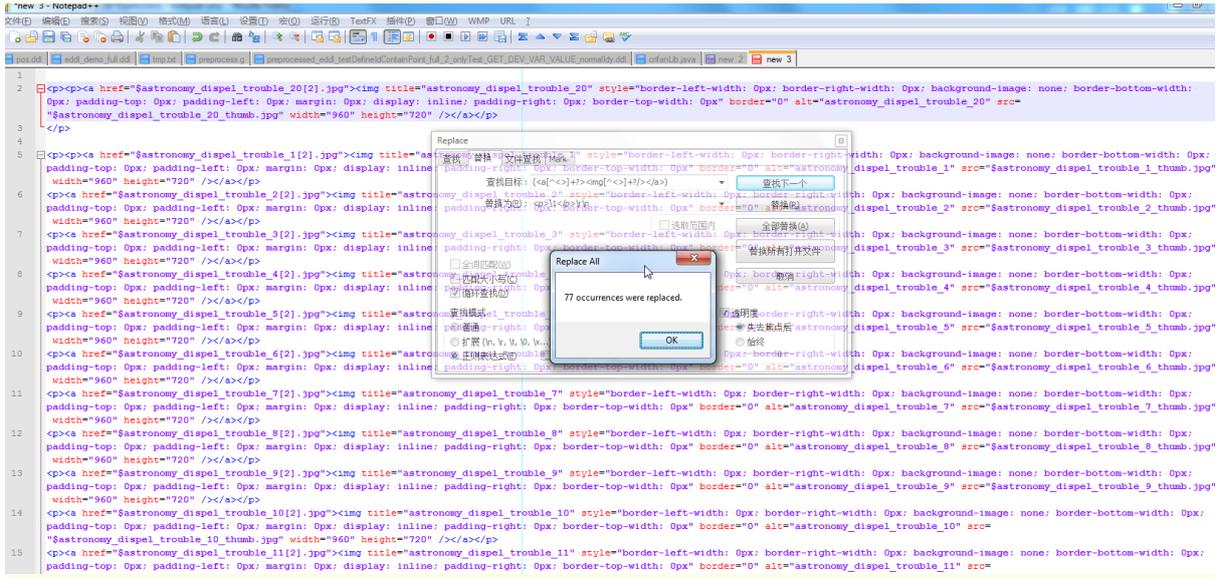
`([S]+)(http://www\.crifan\.com\/w+)`

`<para><link xl:href="\"/>`

替换前是：



Notepad++ 的功能详解



如此，高效的解决了问题。

例 3.13. Notepad++ 正则表达式替换举例：给sect2添加xml:id和title

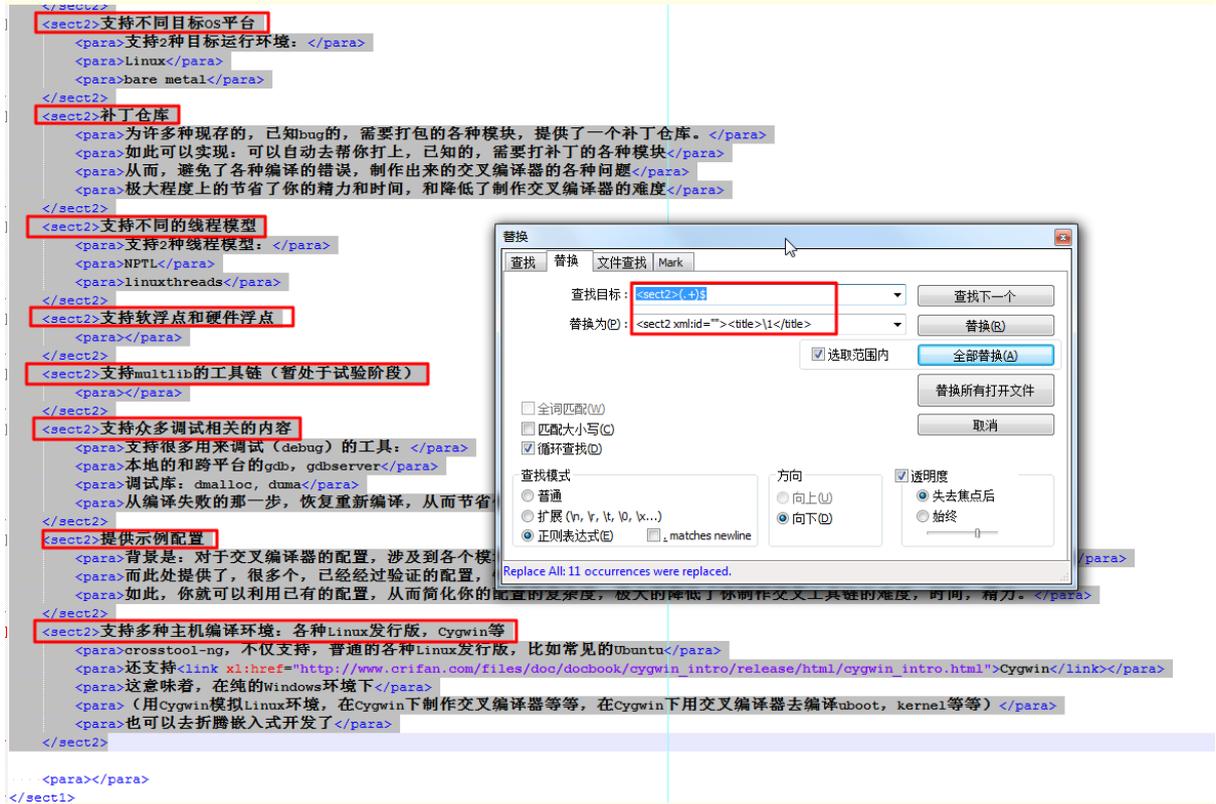
背景是，为crosstool_ng写docbook的xml，然后想要对于之前的itemizedlist，改为sect2，已经手动替换itemizedlist为sect2了。

剩下的，还需要给sect2添加对应的xml:id和title，所以，就去Notepad++中写替换的正则

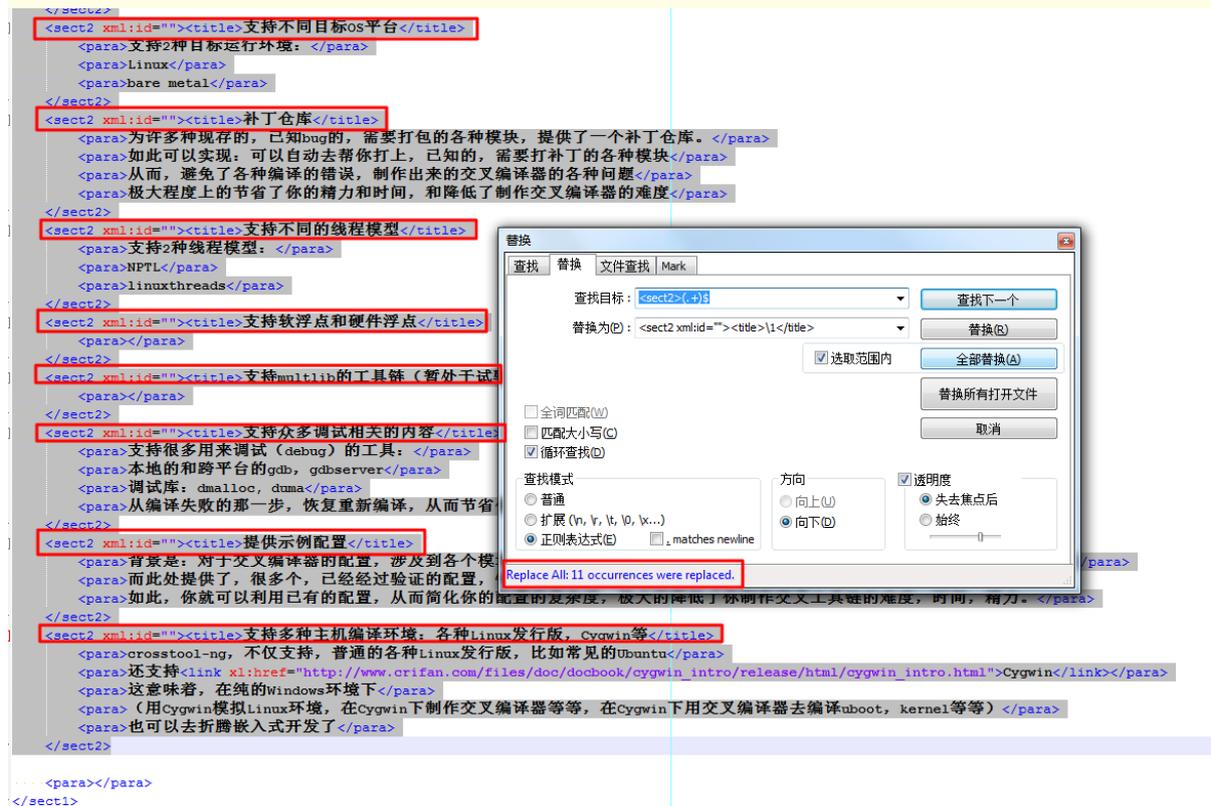
`<sect2>(.)$`

`<sect2 xml:id="" ><title>\1</title>`

将：



替换为：



如此，省去了几十次的，手动输入xml:id和title，以及复制和粘贴的繁琐工作。

简直就是一秒变格格的节奏啊，^_^

例 3.14. Notepad++ 正则表达式替换举例：保持sect2和title添加xml:id

在写Python教程的book的xml时，需要编辑对应的xml，涉及到处理sect2：

用正则：

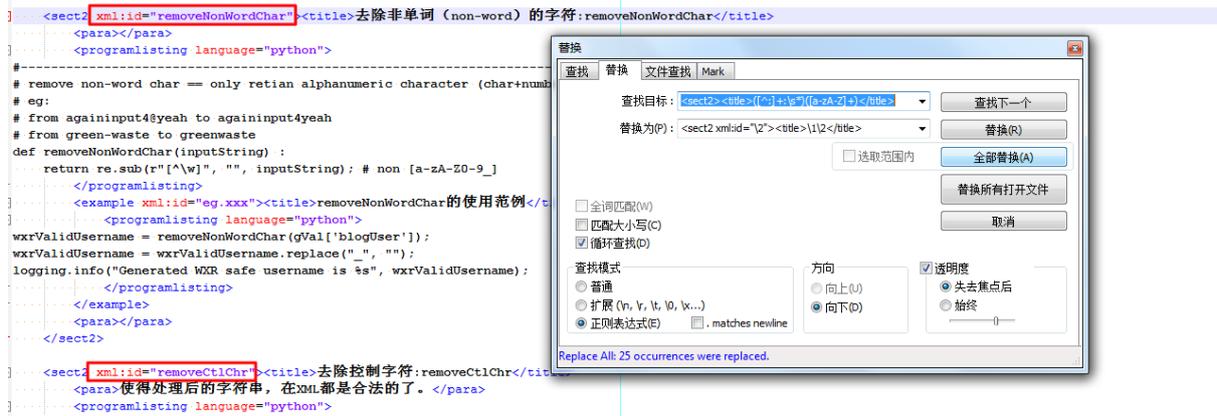
```
<sect1> <title>([^\:]+\:s*)([a-zA-Z]+)</title>
```

```
<sect1 xml:id="\2"> <title>\1\2</title>
```

将：



替换为：



又省去很多复杂的手动的复制粘贴的体力活了。

例 3.15. Notepad++ 正则表达式替换举例：去除单引号变成antlr的token

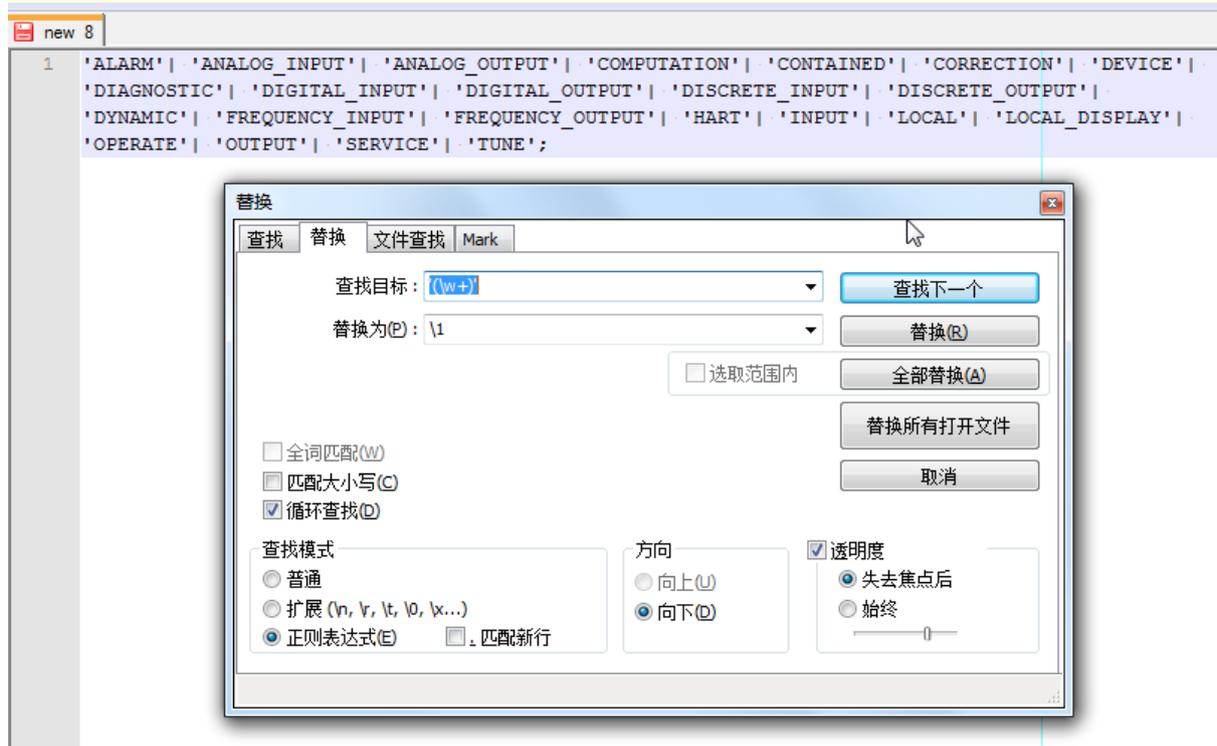
在折腾antlr的grammar时，将原先rule中有的literal都处理为token，所以去把部分单引号，都去掉，变成对应的token：

用正则：

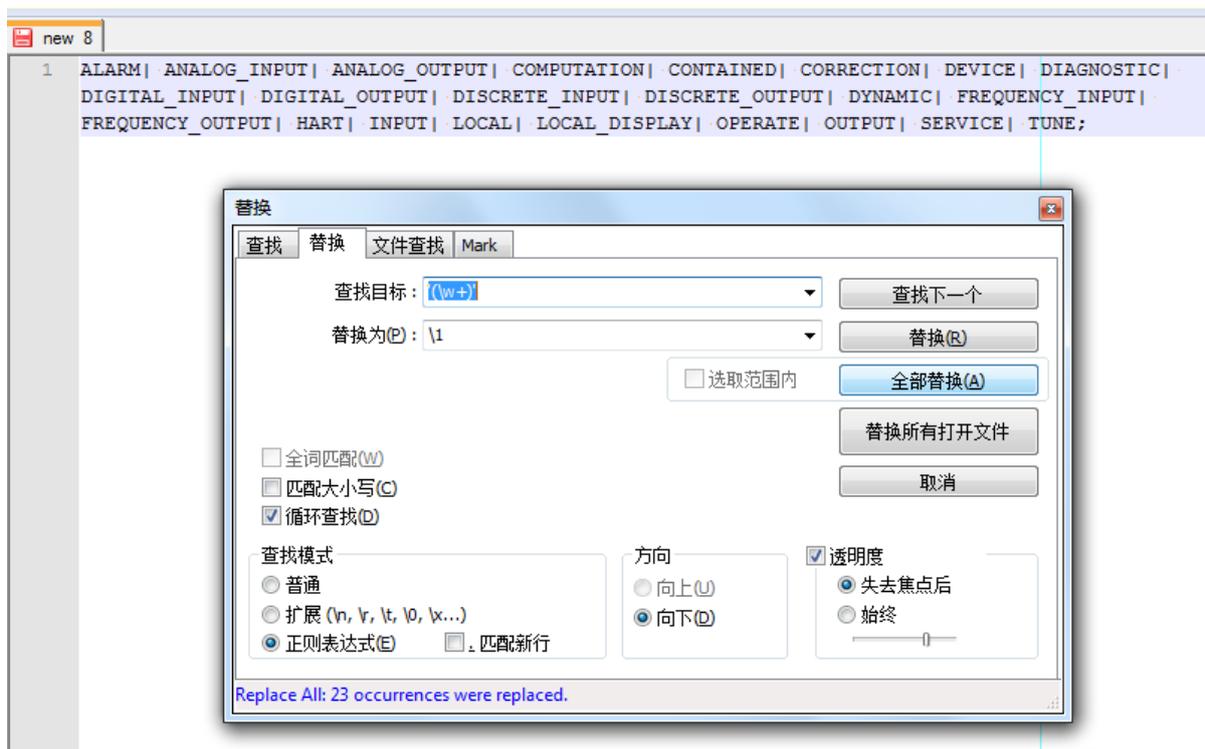
'(\w+)

\1

将：



替换为：



不用自己手动地去掉对应的单引号了。

例 3.16. Notepad++ 正则表达式替换举例：将单引号加ID变成antlr的token的定义

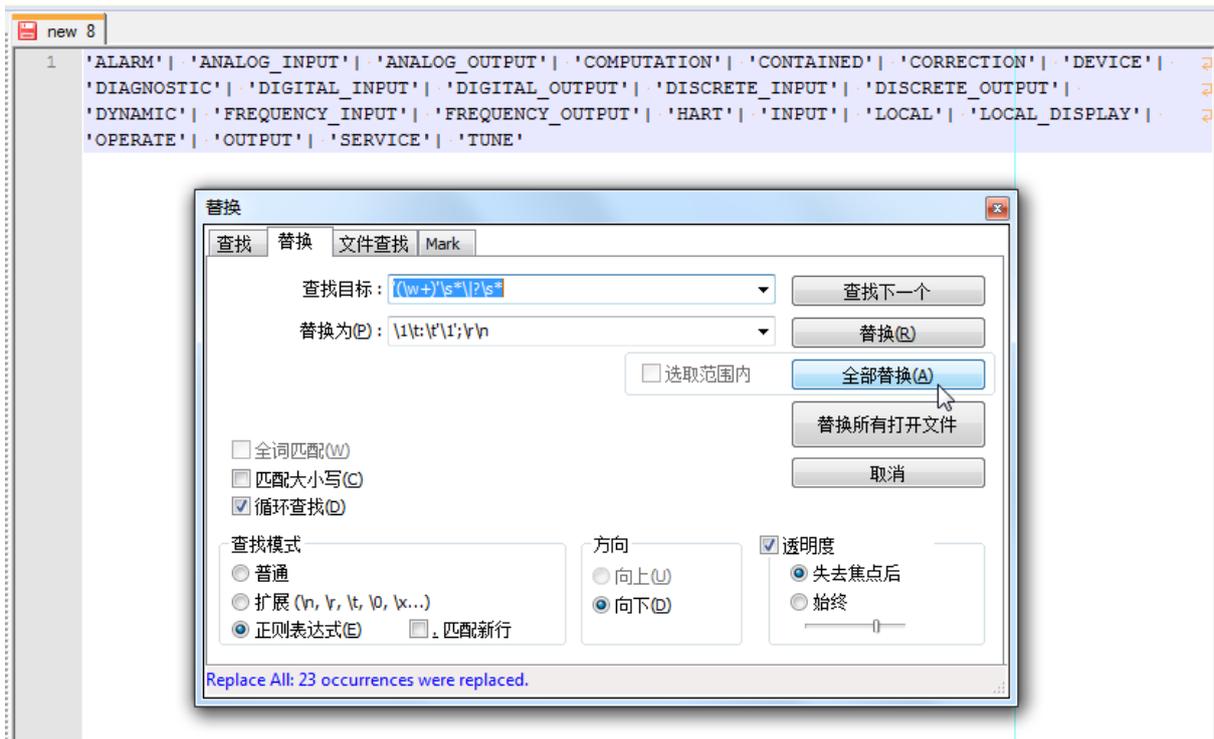
在折腾antlr的grammar时，将原先单引号加上id的写法，直接变成对应的token的定义，并且加上对应的换行：

用正则：

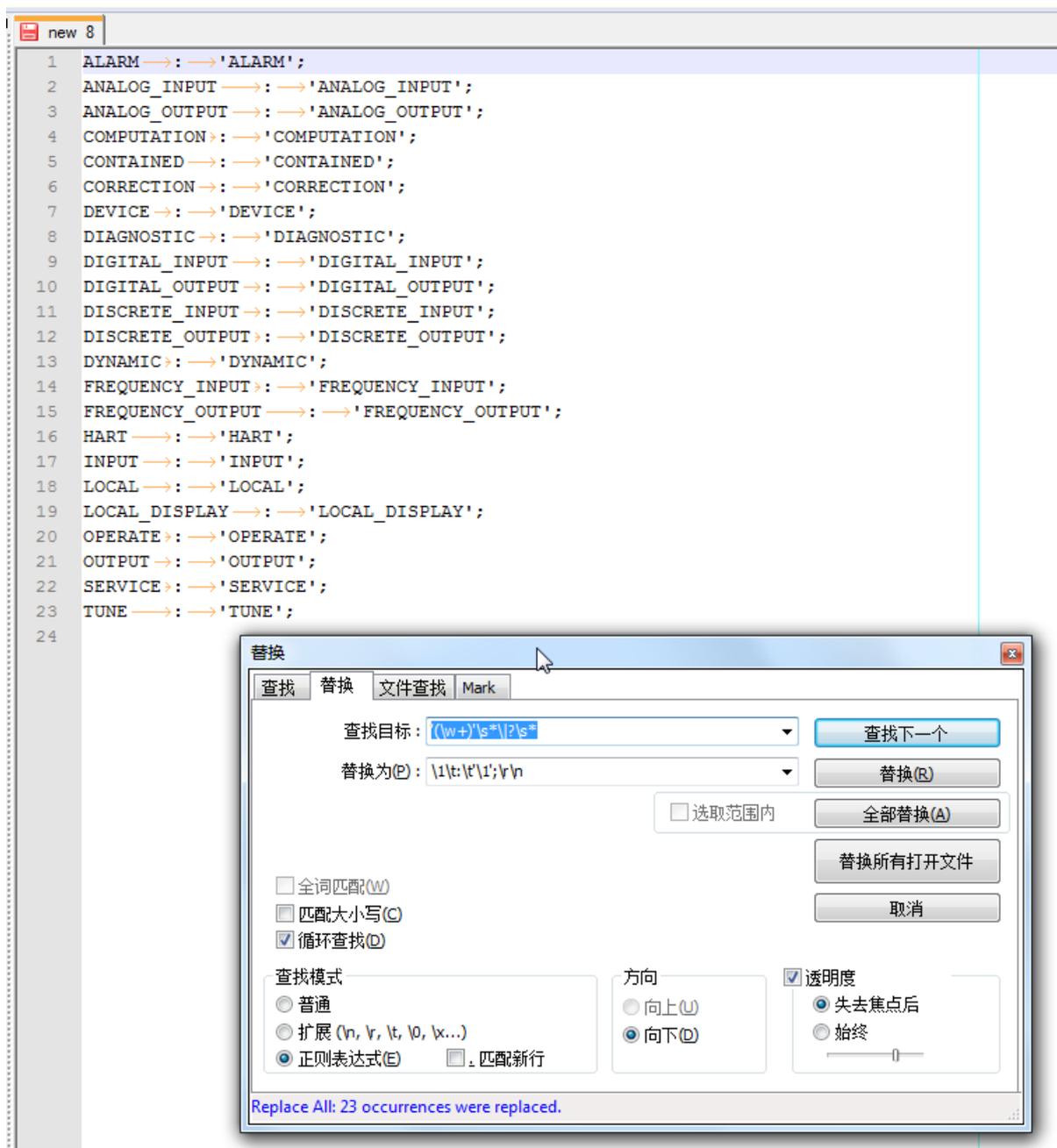
```
'(\w+)\s*|'?s*
```

```
\1\t:t'\1';\r\n
```

将：



替换为：



很明显，还是很帅的，一次性搞定去除引号，加上对应的token的定义，再加上对应的回车换行。
正则，就是高效率啊。

例 3.17. Notepad++ 正则表达式替换举例：dd宏定义中去除多国语言字符串

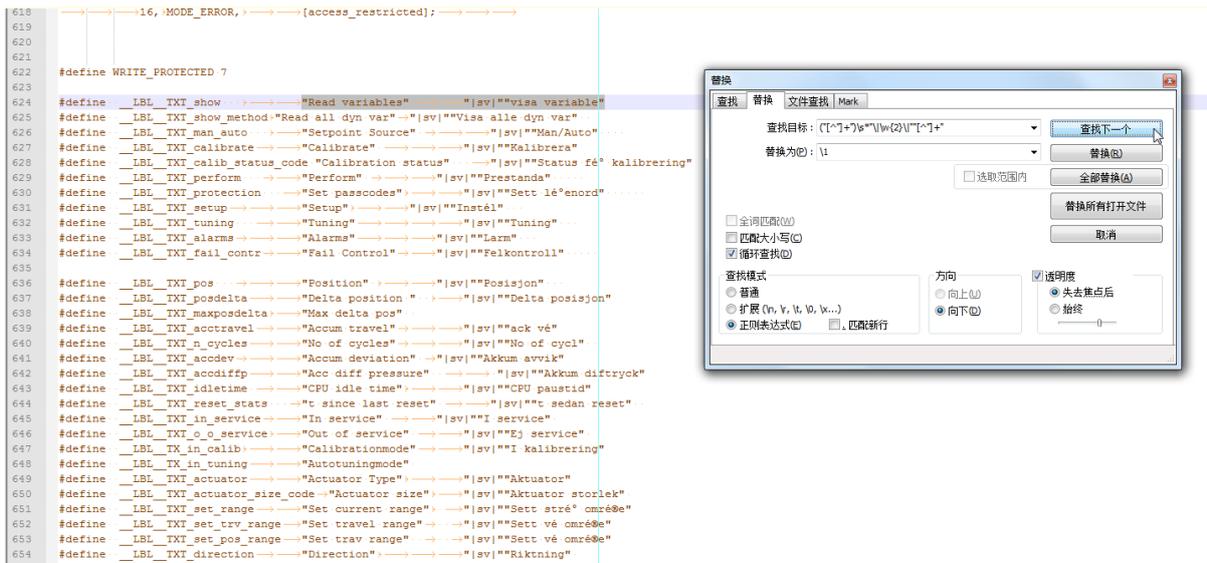
在折腾HART的EDDL文件时，将宏定义中多国语言字符串去除掉：

用正则：

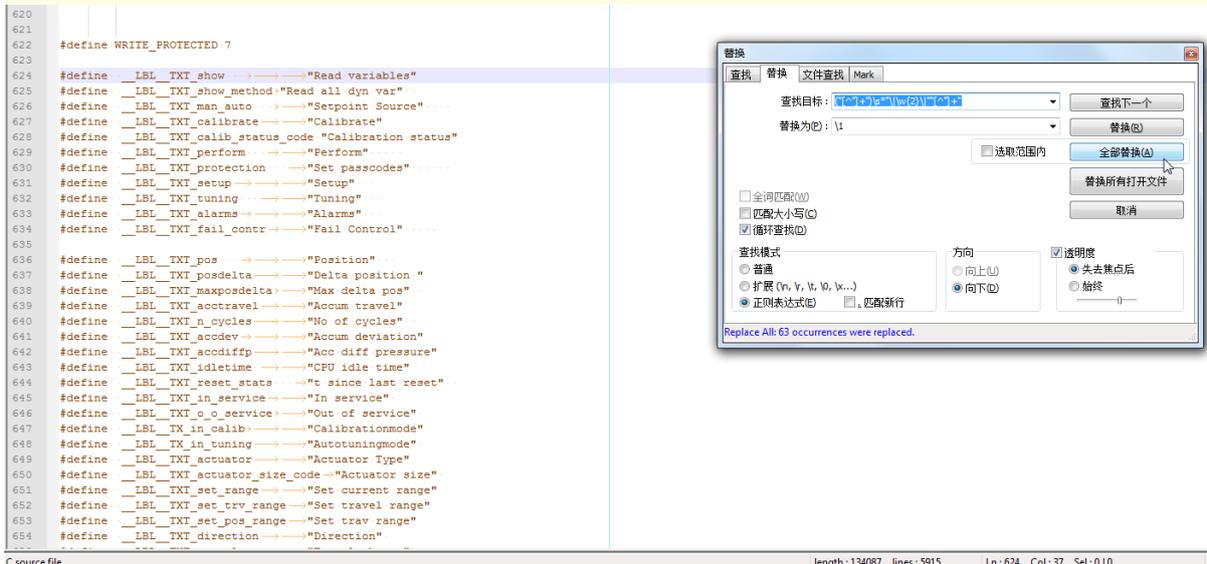
`("[^"]+")\s*"\\w{2}"["^"]+`

`\1`

将：



替换为：



如此，省去了，一点点选择和删除对应的内容了。

例 3.18. Notepad++ 正则表达式替换举例：C宏定义转java变量定义

在折腾

[【记录】给usb-serial-for-android中的Silicon Labs的CP2102中添加RTS和DTR的支持](#)⁷

时，将C语言中的宏定义，转换为对应的Java中的变量定义：

用正则：

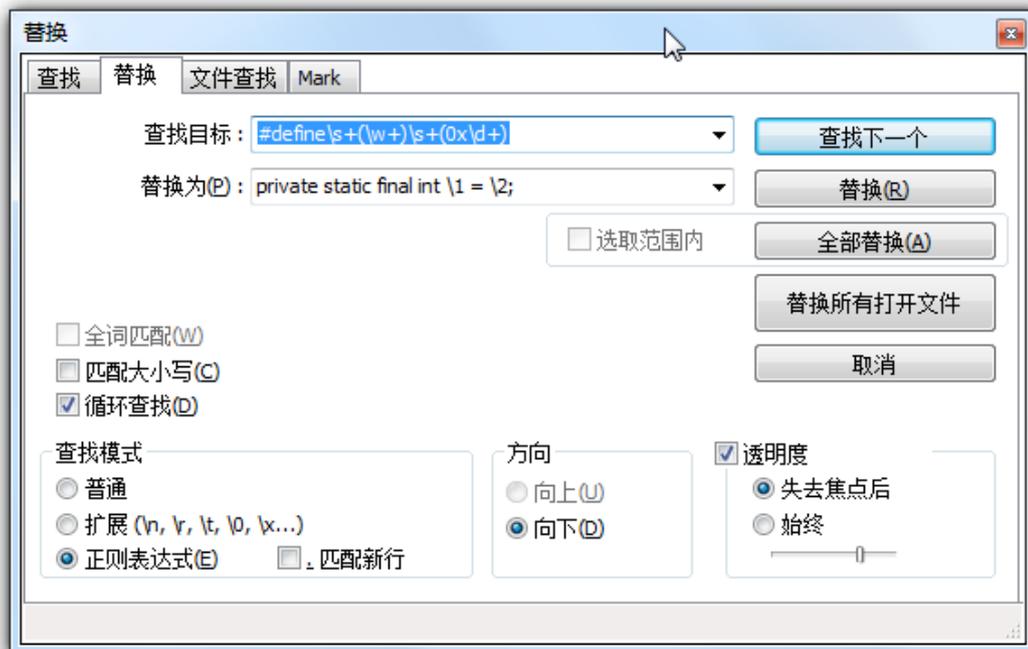
```
#define\s+(\w+)\s+(0x\d+)
```

```
private static final int \1 = \2;
```

将：

⁷ http://www.crifan.com/add_rts_and_dtr_support_for_silicon_labs_cp2102_in_usb_serial_for_android/

```
1 /* CP210X_SET_BREAK */
2 #define BREAK_ON → → 0x0001
3 #define BREAK_OFF → → 0x0000
4
5 /* CP210X_(SET_MHS|GET_MDMSTS) */
6 #define CONTROL_DTR → → 0x0001
7 #define CONTROL_RTS → → 0x0002
8 #define CONTROL_CTS → → 0x0010
9 #define CONTROL_DSR → → 0x0020
10 #define CONTROL_RING → → 0x0040
11 #define CONTROL_DCD → → 0x0080
12 #define CONTROL_WRITE_DTR → 0x0100
13 #define CONTROL_WRITE_RTS → 0x0200
```



替换为：

```

1  /* CP210X_SET_BREAK */
2  private static final int BREAK_ON = 0x0001;
3  private static final int BREAK_OFF = 0x0000;
4
5  /* CP210X_(SET_MHS|GET_MDMSTS) */
6  private static final int CONTROL_DTR = 0x0001;
7  private static final int CONTROL_RTS = 0x0002;
8  private static final int CONTROL_CTS = 0x0010;
9  private static final int CONTROL_DSR = 0x0020;
10 private static final int CONTROL_RING = 0x0040;
11 private static final int CONTROL_DCD = 0x0080;
12 private static final int CONTROL_WRITE_DTR = 0x0100;
13 private static final int CONTROL_WRITE_RTS = 0x0200;

```



很是方便好用，不用自己一点点改代码了。

例 3.19. Notepad++ 正则表达式替换举例：docbook的link变bibliomixed

在写

[ANTLR教程](#)⁸

时，将别的某个docbook的xml中已有的帖子链接，都是link形式的，去转换为bibliomixed的形式。

本来都是手动一点点复制粘贴，然后再修改ref的xml:id的值的，很是麻烦，现在去用正则，一次性处理：

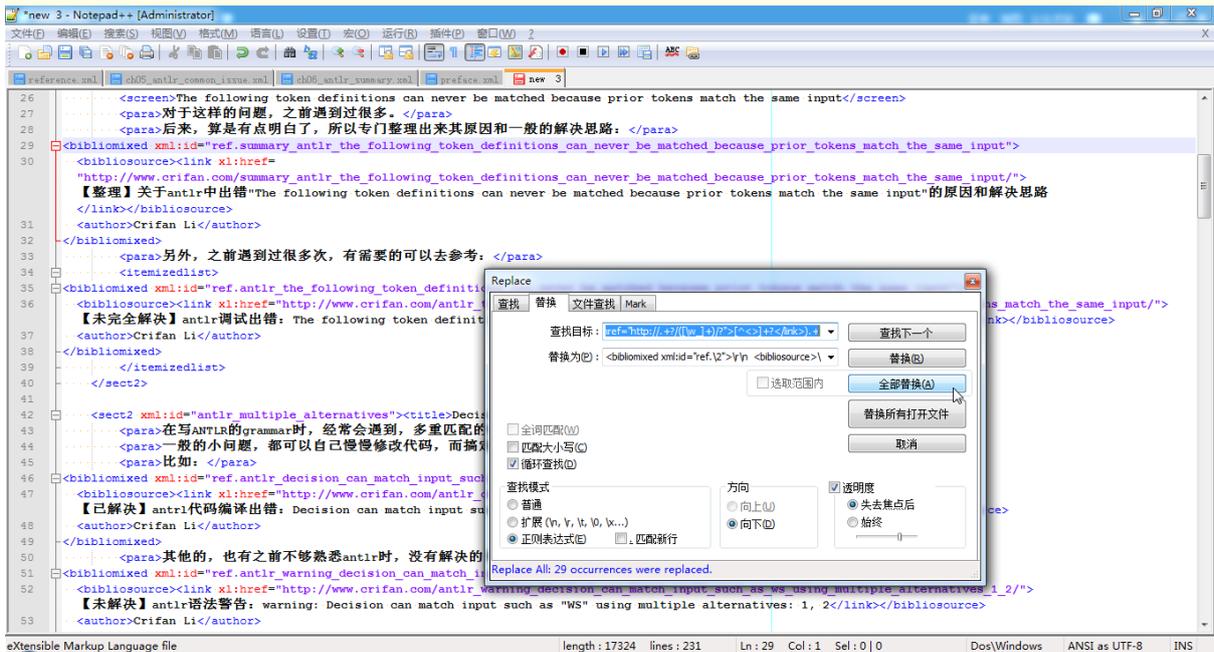
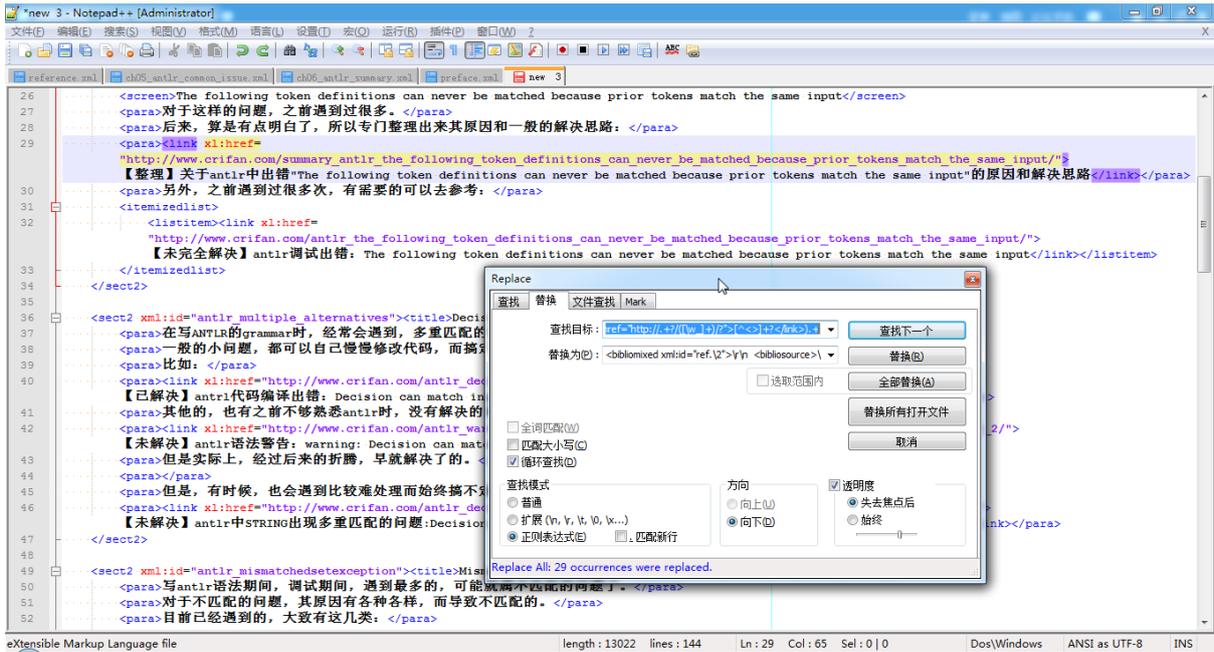
```
.(<link\s+xml:href="http://.+?/([\w_]+)/?">[^<>]+?</link>).+
```

```
<bibliomixed xml:id="ref.\2">\r\n <bibliosource>\1</bibliosource>\r\n <author>Crifan Li</author>\r\n</bibliomixed>
```

将：

⁸ http://www.crifan.com/files/doc/docbook/antlr_tutorial/release/htmls/

Notepad++的功能详解



就可以免去手动的麻烦了。

对应的，就是把之前的link内容：

```
<para> <link xl:href="http://www.crifan.com/summary_antlr_the_following_token_definitions_can_never_be_matched_because_prior_tokens_match_the_sam理】关于antlr中出错"The following token definitions can never be matched because prior tokens match the same input"的原因和解决思路</link> </para>
```

就变成了：

当你双击选择某个单词后，本身Notepad++就会自动帮你高亮对应单词，然后再点击“计数”，就可以弹出对话框显示统计的结果，显示该单词一共出现了多少次。然后不得不感叹，Notepad++真是很是方便使用啊。

3.6. Notepad++的列编辑功能

下面来解释Notepad++中的强大且好用的列编辑功能。

3.6.1. 什么是列编辑模式

普通编辑器，编辑文本的时候，选中一部分内容，都是在一行或多行的范围内操作，从左到右的，所以，可以看做是行模式。

与此相对应的，就是上下方向的列模式了。

进入列模式后，操作顺序和范围，是在所选范围内，从上到下的，比如全部插入某些字符等等，是在所选的列模式操作区域内，所涉及的每一行，都对应的插入这些字符。

当有些高级的操作，需要对不同的列，同时进行编辑的话，那么列编辑模式，就非常有用。

3.6.2. Notepad++的列编辑模式的基本操作

在Notepad++中，按住Alt键之后，就处于列（编辑）模式了。

比如，按住Alt键，此处从上到下，选择多列：

```

<sect4><title>Notepad++的列编辑模式的基本操作</title>
<para>在Notepad++中，按住<em>Alt键</em>之后，就处于列（编辑）模式了。</para>
<para>比如，按住Alt键，此处从上到下，选择多列：</para>
</para>
</para>
</para>
</para>
</para>
</para>
</para>
</para>
</para>
</sect4>

```

例 3.20. 列编辑：多行输入同样内容

然后多行同时输入同样的内容，效果如下：

```

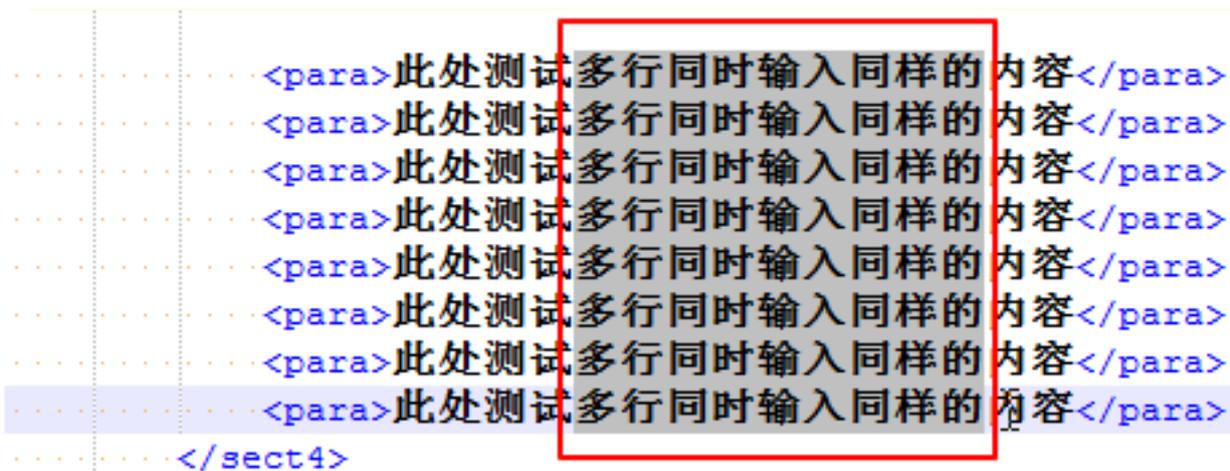
<sect4><title>Notepad++的列编辑模式的基本操作</title>
<para>在Notepad++中，按住<em>Alt键</em>之后，就处于列（编辑）模式了。</para>
<para>比如，按住Alt键，此处从上到下，选择多列：</para>
<para>此处测试多行同时输入同样的内容</para>
<para>此处测试多行同时输入同样的内容</para>
<para>此处测试多行同时输入同样的内容</para>
<para>此处测试多行同时输入同样的内容</para>
<para>此处测试多行同时输入同样的内容</para>
<para>此处测试多行同时输入同样的内容</para>
<para>此处测试多行同时输入同样的内容</para>
<para>此处测试多行同时输入同样的内容</para>
</sect4>

```

例 3.21. 列编辑：删除多行内容

然后也可以同时删除多行内容：

先按住Alt键，选后同时选取多列：



```

<para>此处测试多行同时输入同样的内容</para>
<para>此处测试多行同时输入同样的内容</para>
<para>此处测试多行同时输入同样的内容</para>
<para>此处测试多行同时输入同样的内容</para>
<para>此处测试多行同时输入同样的内容</para>
<para>此处测试多行同时输入同样的内容</para>
<para>此处测试多行同时输入同样的内容</para>
<para>此处测试多行同时输入同样的内容</para>
</sect4>

```

然后松掉Alt键，点击右键选择删除，或者直接按键盘上面的Delete键，都可以实现删除所选的多行中对应部分的内容：



```

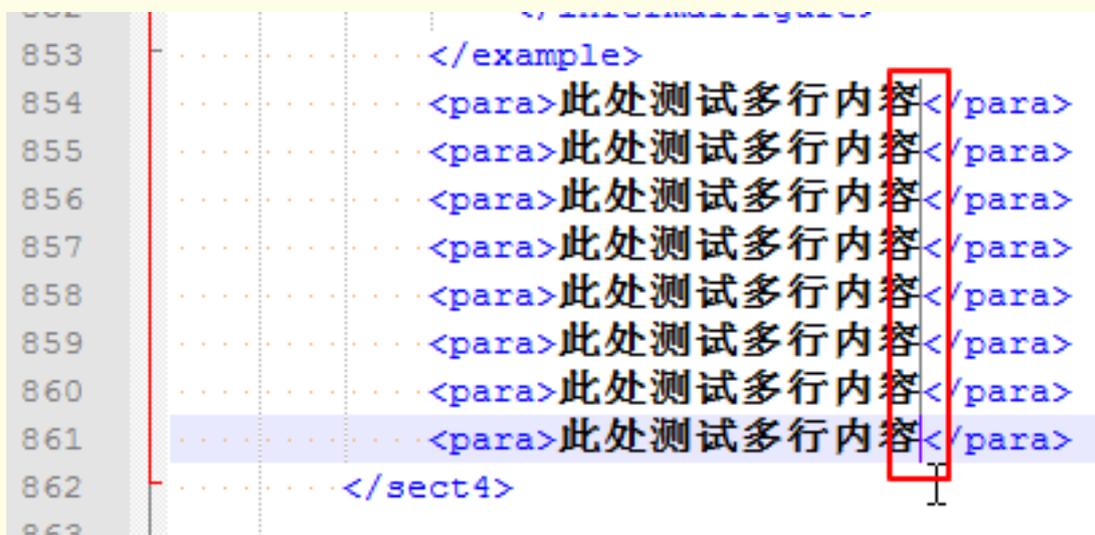
<para>此处测试多行内容</para>
<para>此处测试多行内容</para>
<para>此处测试多行内容</para>
<para>此处测试多行内容</para>
<para>此处测试多行内容</para>
<para>此处测试多行内容</para>
<para>此处测试多行内容</para>
<para>此处测试多行内容</para>
</sect4>

```

例 3.22. 列编辑：多行中插入相同或不同的内容

进入列编辑模式后，除了可以手动输入内容外，也可以通过插入，实现输入多行内容：

按住Alt键进入列编辑模式后：



```

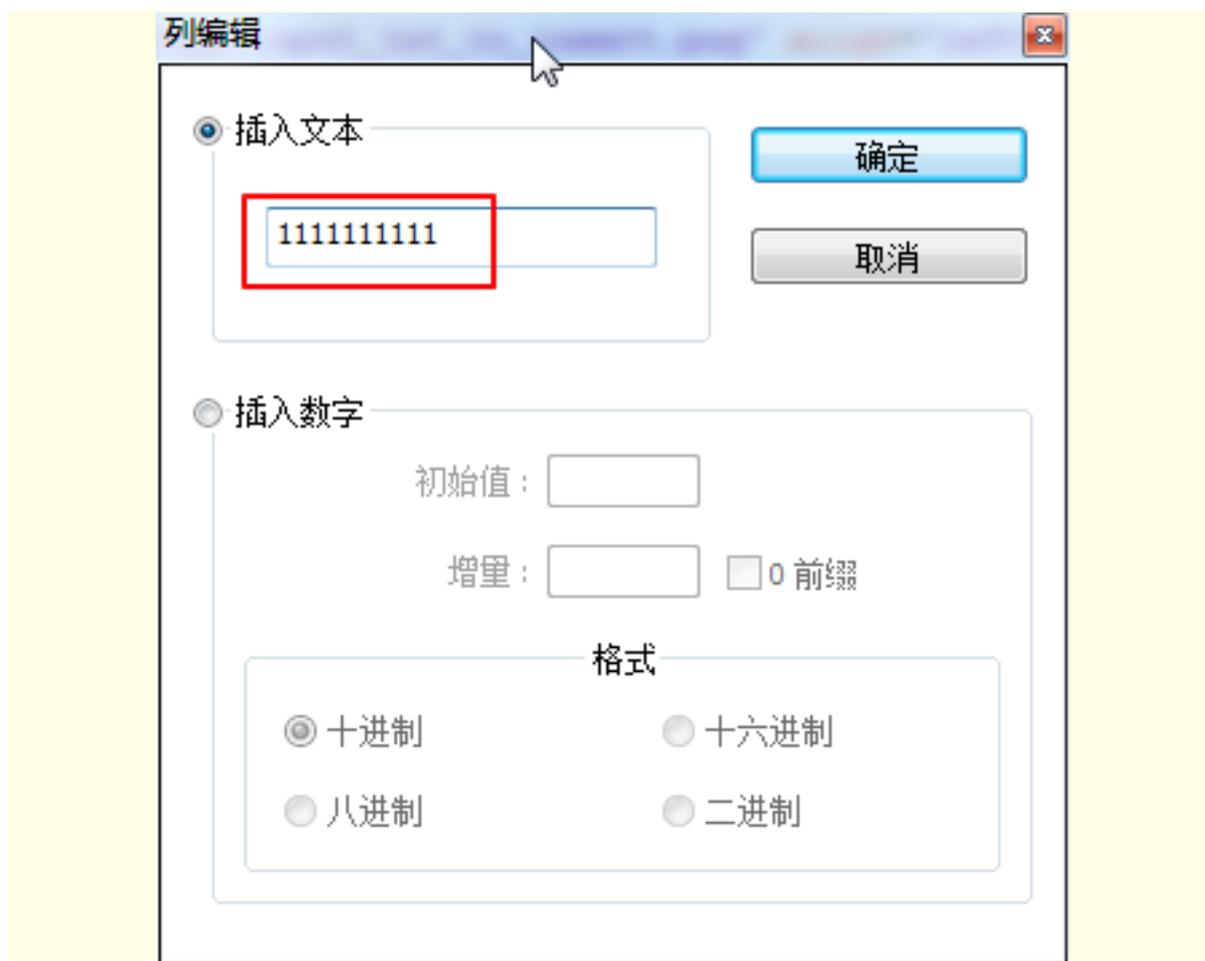
853 </example>
854 <para>此处测试多行内容</para>
855 <para>此处测试多行内容</para>
856 <para>此处测试多行内容</para>
857 <para>此处测试多行内容</para>
858 <para>此处测试多行内容</para>
859 <para>此处测试多行内容</para>
860 <para>此处测试多行内容</para>
861 <para>此处测试多行内容</para>
862 </sect4>
863

```

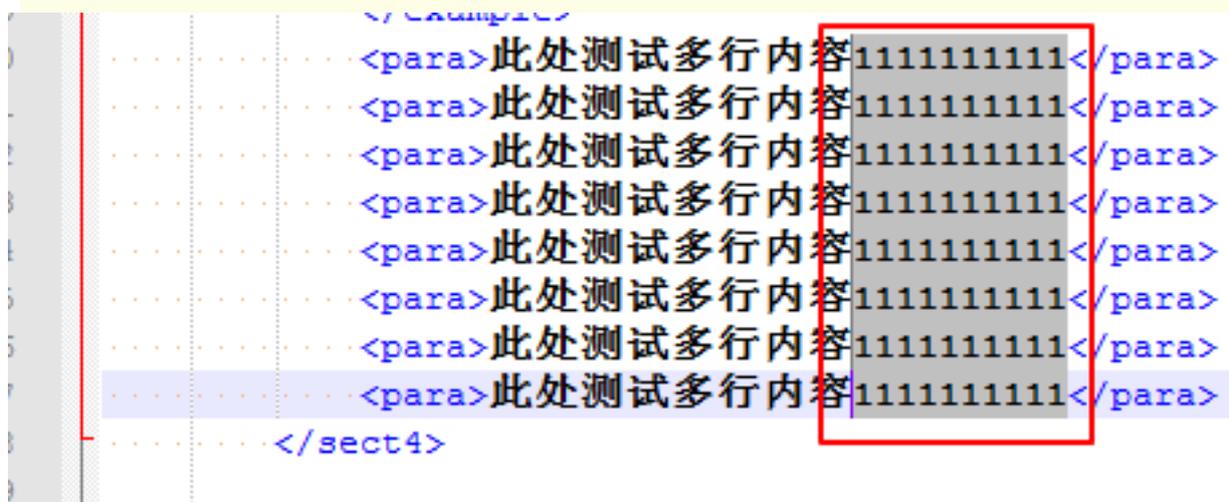
然后松掉Alt键，点击 编辑(E) ⇒ 列编辑 Alt+C：



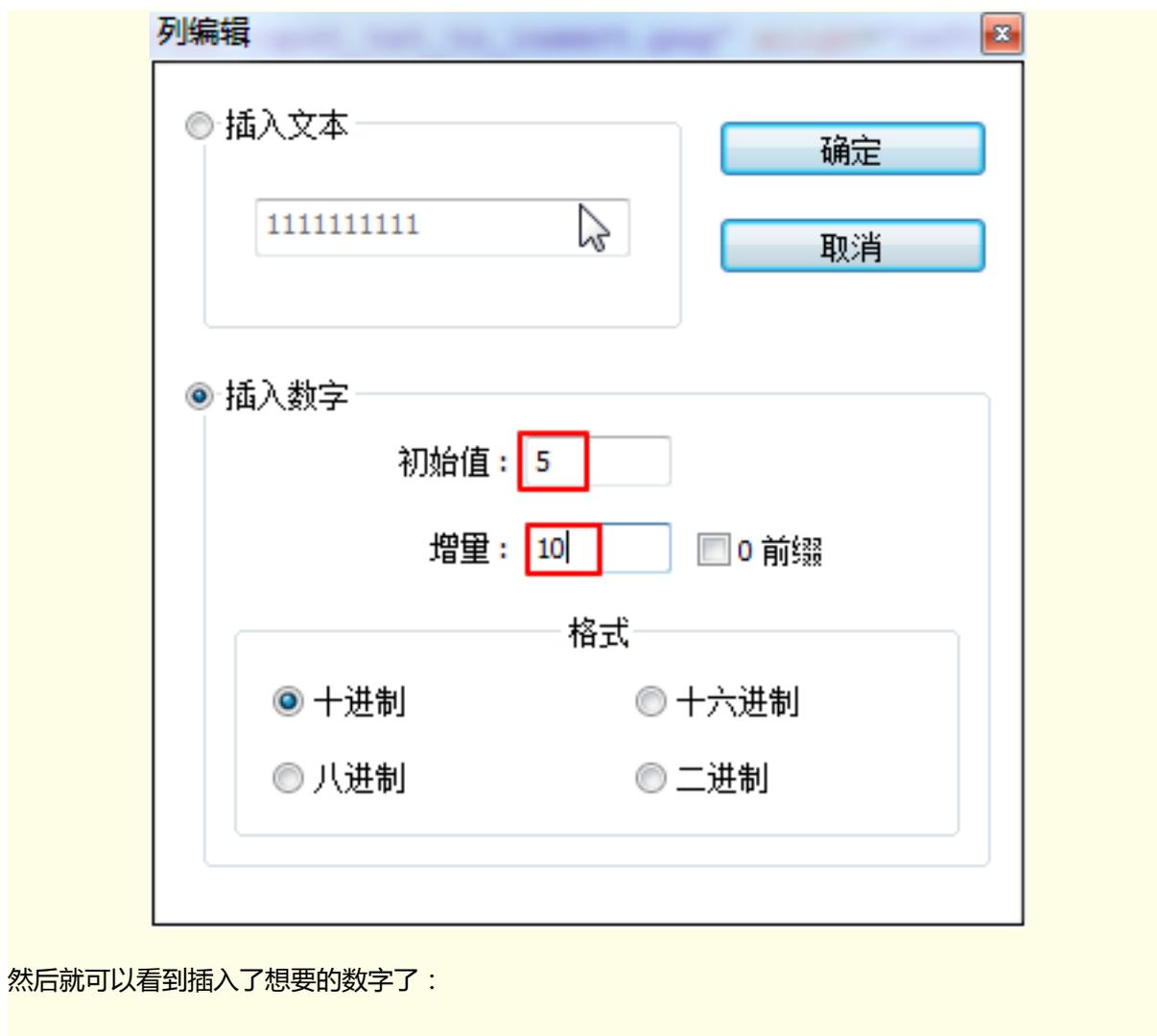
然后在“插入文本”处，填写你所要插入的内容：



此处填入想要插入的内容，即可同时插入多行内容：



同样，可以看到，上面还有个“插入数字”的功能，可以插入自己所想要的某种序列的数字：



然后就可以看到插入了想要的数字了：

```
874 ..... </example>
875 ..... <para>此处测试多行内容1111111111 5 </para>
876 ..... <para>此处测试多行内容1111111111 15 </para>
877 ..... <para>此处测试多行内容1111111111 25 </para>
878 ..... <para>此处测试多行内容1111111111 35 </para>
879 ..... <para>此处测试多行内容1111111111 45 </para>
880 ..... <para>此处测试多行内容1111111111 55 </para>
881 ..... <para>此处测试多行内容1111111111 65 </para>
882 ..... <para>此处测试多行内容1111111111 75 </para>
883 ..... </sect4>
```

虽然上面只是一些基本操作，但是对于很多人，如果需要用到此功能，也会觉得很方便了。

3.6.3. Notepad++的列编辑模式的高级用法

其实，列编辑的功能的好处，在于相对更加复杂一点的用法。

下面就介绍一下，我之前所遇到的情况，以及如何利用列编辑模式，来提高工作效率的：

例 3.23. 列编辑：同时复制和粘贴多列

比如，我之前，需要得到这样的xml内容：

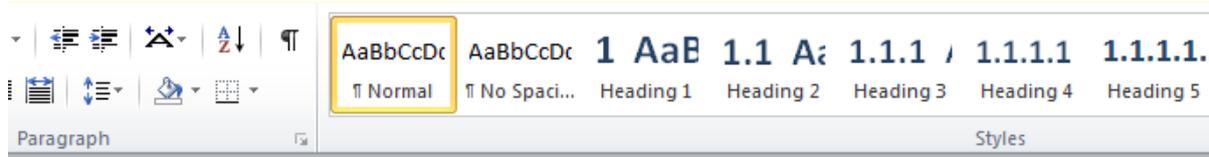
```

20 <sect1><title>ASCII 编码表(0-127)</title>
21 <table border="1"><thead><tr><th>ASCII 编码表(0-127)</th></tr></thead>
22 <tbody>
23 <tr><td></td></tr>
24 <tr><td></td></tr>
25 </tbody>
26 </table>
27
28 <tbody>
29 <tr><td></td></tr>
30 <tr><td></td></tr>
31 <tr><td></td></tr>
32 <tr><td></td></tr>
33 <tr><td></td></tr>
34 <tr><td></td></tr>
35 <tr><td></td></tr>
36 <tr><td></td></tr>
37 <tr><td></td></tr>
38 <tr><td></td></tr>
39 <tr><td></td></tr>
40 <tr><td></td></tr>
41 <tr><td></td></tr>
42 <tr><td></td></tr>
43 <tr><td></td></tr>
44 <tr><td></td></tr>
45 <tr><td></td></tr>

```

即，需要在多个entry中，一点点输入对应的数字和字母。

而数据来源，是来自另外已有的word文档中的表格：



图表 1 ASCII 编码表

十六进制	十进制	字符	十六进制	十进制	字符	十六进制	十进制	字符	十六进制	十进制	字符
0	0	NUL	20	32	[space]	40	64	@	60	96	'
1	1	SOH	21	33	!	41	65	A	61	97	a
2	2	STX	22	34	"	42	66	B	62	98	b
3	3	ETX	23	35	#	43	67	C	63	99	c
4	4	EOT	24	36	\$	44	68	D	64	100	d
5	5	ENQ	25	37	%	45	69	E	65	101	e
6	6	ACK	26	38	&	46	70	F	66	102	f
7	7	BEL	27	39	`	47	71	G	67	103	g
8	8	BS	28	40	(48	72	H	68	104	h
9	9	HT	29	41)	49	73	I	69	105	i
0a	10	LF	2a	42	*	4a	74	J	6a	106	j
0b	11	VT	2b	43	+	4b	75	K	6b	107	k
0c	12	FF	2c	44	,	4c	76	L	6c	108	l
0d	13	CR	2d	45	-	4d	77	M	6d	109	m
0e	14	SO	2e	46	.	4e	78	N	6e	110	n
0f	15	SI	2f	47	/	4f	79	O	6f	111	o
10	16	DLE	30	48	0	50	80	P	70	112	p
11	17	DC1	31	49	1	51	81	Q	71	113	q
12	18	DC2	32	50	2	52	82	R	72	114	r
13	19	DC3	33	51	3	53	83	S	73	115	s
14	20	DC4	34	52	4	54	84	T	74	116	t
15	21	NAK	35	53	5	55	85	U	75	117	u
16	22	SYN	36	54	6	56	86	V	76	118	v
17	23	ETB	37	55	7	57	87	W	77	119	w
18	24	CAN	38	56	8	58	88	X	78	120	x
19	25	EM	39	57	9	59	89	Y	79	121	y
1a	26	SUB	3a	58	:	5a	90	Z	7a	122	z
1b	27	ESC	3b	59	;	5b	91	[7b	123	{
1c	28	FS	3c	60	<	5c	92	\	7c	124	
1d	29	GS	3d	61	=	5d	93]	7d	125	}
1e	30	RS	3e	62	>	5e	94	^	7e	126	~
1f	31	US	3f	63	?	5f	95	_	7f	127	[delete]

对此，一般传统的办法，那无外乎，一个个数字，和字母的从word中拷贝，然后粘贴到xml文件中对应的两个entry的位置中去。

此种方法，效率及其低下不说，还很容易出错，所以肯定不是好办法。

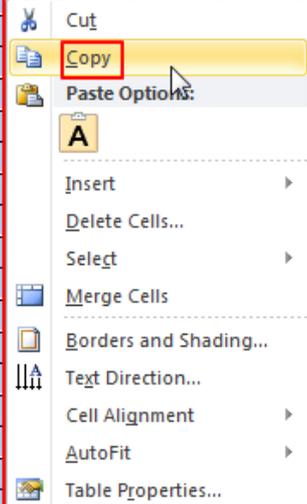
但是如果不会用列编辑的话，那么好像也只能用此很笨的办法，慢慢的去拷贝粘贴了。

下面就来看看，如果使用Notepad++的列编辑，是如何提高效率和准确度的：

首先当前是在目标xml文件中，已经有了对应的entry了：

图表 1 ASCII 编码表

十六进制	十进制	字符	十六进制	十进制	字符	十六进制	十进制	字符	十六进制	十进制	字符
0	0	NUL	20	32	[space]	40	64	@	60	96	'
1	1	SOH	21	33	!	41	65	A	61	97	a
2	2	STX	22	34	"	42	66	B	62	98	b
3	3	ETX	23	35	#	43	67	C	63	99	c
4	4	EOT	24	36	\$	44	68	D	64	100	d
5	5	ENQ	25	37	%	45	69	E	65	101	e
6	6	ACK	26	38	&	46	70	F	66	102	f
7	7	BEL	27	39	`	47	71	G	67	103	g
8	8	BS	28	40	(48	72	H	68	104	h
9	9	HT	29	41)	49	73	I	69	105	i
0a	10	LF	2a	42	*	4a	74	J	6a	106	j
0b	11	VT	2b	43	+	4b	75	K	6b	107	k
0c	12	FF	2c	44	,	4c	76	L	6c	108	l
0d	13	CR	2d	45	-	4d	77	M	6d	109	m
0e	14	SO	2e	46	.	4e	78	N	6e	110	n
0f	15	SI	2f	47	/	4f	79	O	6f	111	o
10	16	DL	30	48	0	50	80	P	70	112	p
11	17	DC1	31	49	1	51	81	Q	71	113	q
12	18	DC2	32	50	2	52	82	R	72	114	r
13	19	DC3	33	51	3	53	83	S	73	115	s
14	20	DC4	34	52	4	54	84	T	74	116	t
15	21	NAK	35	53	5	55	85	U	75	117	u
16	22	SYN	36	54	6	56	86	V	76	118	v
17	23	ETB	37	55	7	57	87	W	77	119	w
18	24	CAN	38	56	8	58	88	X	78	120	x
19	25	EM	39	57	9	59	89	Y	79	121	y
1a	26	END	3a	58	:	5a	90	Z	7a	122	z
1b	27	ESC	3b	59	;	5b	91	[7b	123	{
1c	28	FS	3c	60	<	5c	92	\	7c	124	
1d	29	GS	3d	61	=	5d	93]	7d	125	}
1e	30	RS	3e	62	>	5e	94	^	7e	126	~
1f	31	US	3f	63	?	5f	95	_	7f	127	[delete]



3. 字母的字符编码：EASCII 和 ISO

然后在Notepad++中先进入列编辑模式：

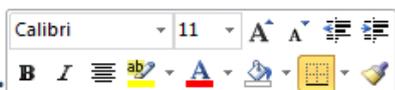

```
3 </thead>
4 <row><entry>十六进制</entry><entry>十进制</entry><entry>字符</ent
  </entry><entry>十六进制</entry><entry>十进制</entry><entry>字符</
5 </thead>
6
7 <tbody>
8 <row><entry>0
9 1
10 2
11 3
12 4
13 5
14 6
15 7
16 8
17 9
18 0a
19 0b
20 0c
21 0d
22 0e
23 0f
24 10
25 11
26 12
27 13
28 14
29 15
30 16
31 17
32 18
33 19
34 1a
35 1b
36 1c
37 1d
38 1e
39 1f
40 </entry><entry></entry><entry></entry><entry></entry><entry></entr
41 <row><entry></entry><entry></entry><entry></entry><entry></entry><
42 <row><entry></entry><entry></entry><entry></entry><entry></entry><
43 <row><entry></entry><entry></entry><entry></entry><entry></entry><
44 <row><entry></entry><entry></entry><entry></entry><entry></entry><
45 <row><entry></entry><entry></entry><entry></entry><entry></entry><
46 <row><entry></entry><entry></entry><entry></entry><entry></entry><
47 <row><entry></entry><entry></entry><entry></entry><entry></entry><
```

后来才知道，原来应该这么做：

同样先去word中拷贝对应的列的内容：

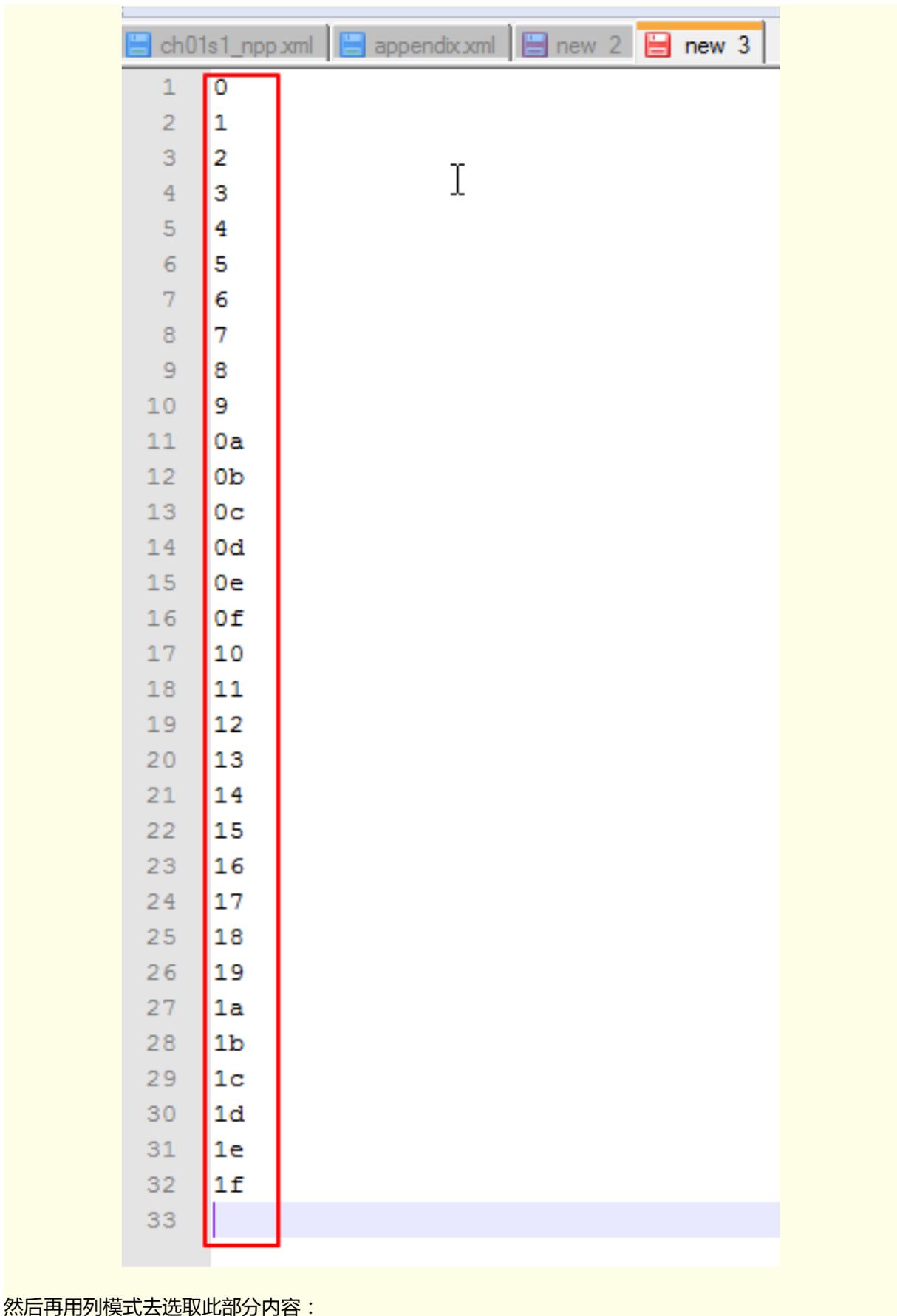
图表 1 ASCII 编码表

十六进制	十进制	字符	十六进制	十进制	字符	十六进制	十进制	字符	十六进制	十进制	字符
0	0	NUL	20	32	[space]	40	64	@	60	96	'
1	1	SOH	21	33	!	41	65	A	61	97	a
2	2	STX	22	34	"	42	66	B	62	98	b
3	3	ETX	23	35	#	43	67	C	63	99	c
4	4	EOT	24	36	\$	44	68	D	64	100	d
5	5	ENQ	25	37	%	45	69	E	65	101	e
6	6	ACK	26	38	&	46	70	F	66	102	f
7	7	BEL	27	39	`	47	71	G	67	103	g
8	8	BS	28	40	(48	72	H	68	104	h
9	9	HT	29	41)	49	73	I	69	105	i
0a	10	LF	2a	42	*	4a	74	J	6a	106	j
0b	11	VT	2b	43	+	4b	75	K	6b	107	k
0c	12	FF	2c	44	,	4c	76	L	6c	108	l
0d	13	CR	2d	45	-	4d	77	M	6d	109	m
0e	14	SO	2e	46	.	4e	78	N	6e	110	n
0f	15	SI	2f	47	/	4f	79	O	6f	111	o
10	16	DL	30	48	0	50	80	P	70	112	p
11	17	DC1	31	49	1	51	81	Q	71	113	q
12	18	DC2	32	50	2	52	82	R	72	114	r
13	19	DC3	33	51	3	53	83	S	73	115	s
14	20	DC4	34	52	4	54	84	T	74	116	t
15	21	NAK	35	53	5	55	85	U	75	117	u
16	22	SYN	36	54	6	56	86	V	76	118	v
17	23	ETB	37	55	7	57	87	W	77	119	w
18	24	CAN	38	56	8	58	88	X	78	120	x
19	25	EM	39	57	9	59	89	Y	79	121	y
1a	26	END	3a	58	:	5a	90	Z	7a	122	z
1b	27	ESC	3b	59	;	5b	91	[7b	123	{
1c	28	FS	3c	60	<	5c	92	\	7c	124	
1d	29	GS	3d	61	=	5d	93]	7d	125	}
1e	30	RS	3e	62	>	5e	94	^	7e	126	~
1f	31	US	3f	63	?	5f	95	_	7f	127	[delete]

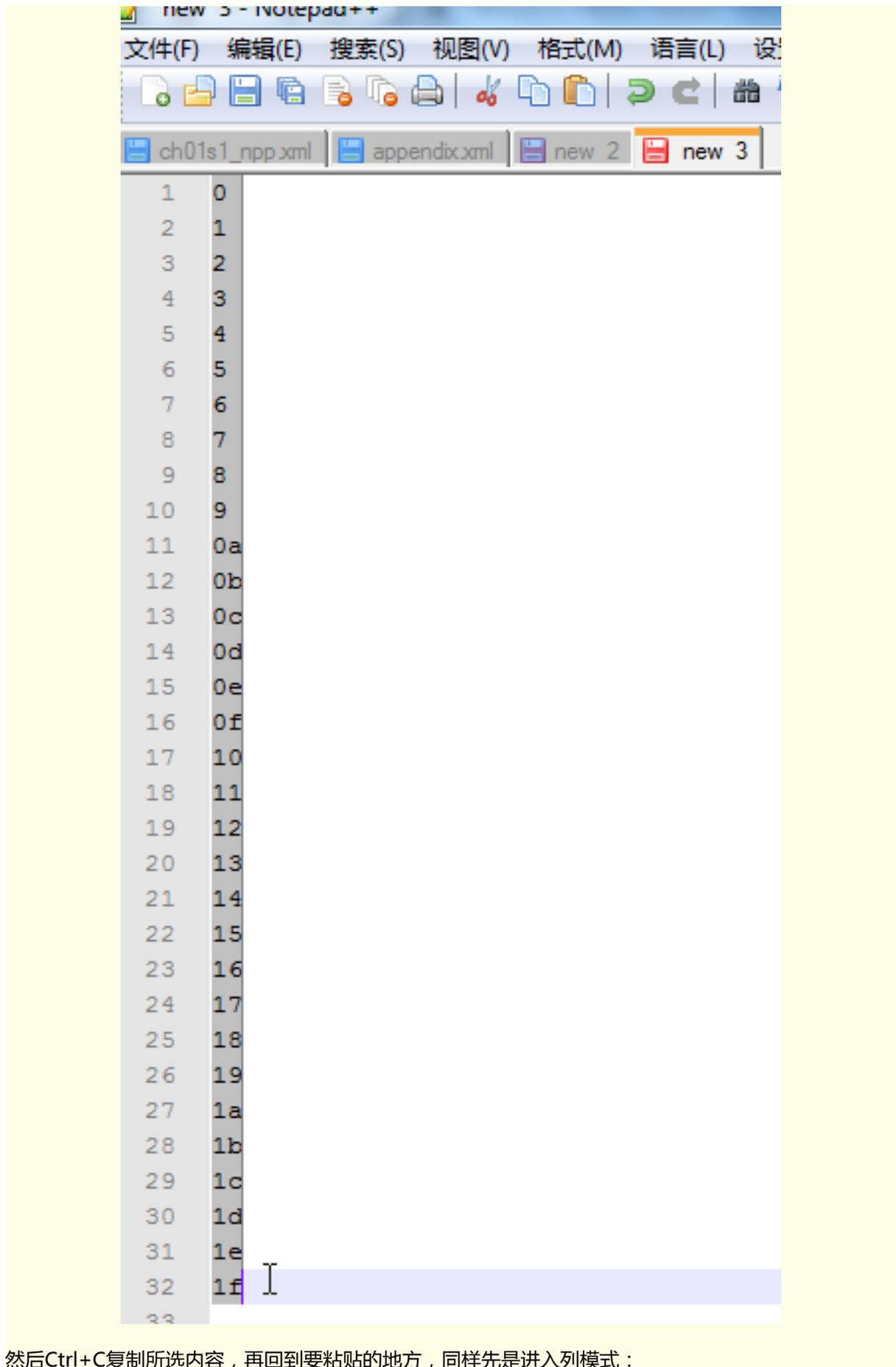


3. 字母的字符编码：EASCII 和 ISO

然后在Notepad++中，新建一个页面，将拷贝的内容，粘贴到新建页面中：



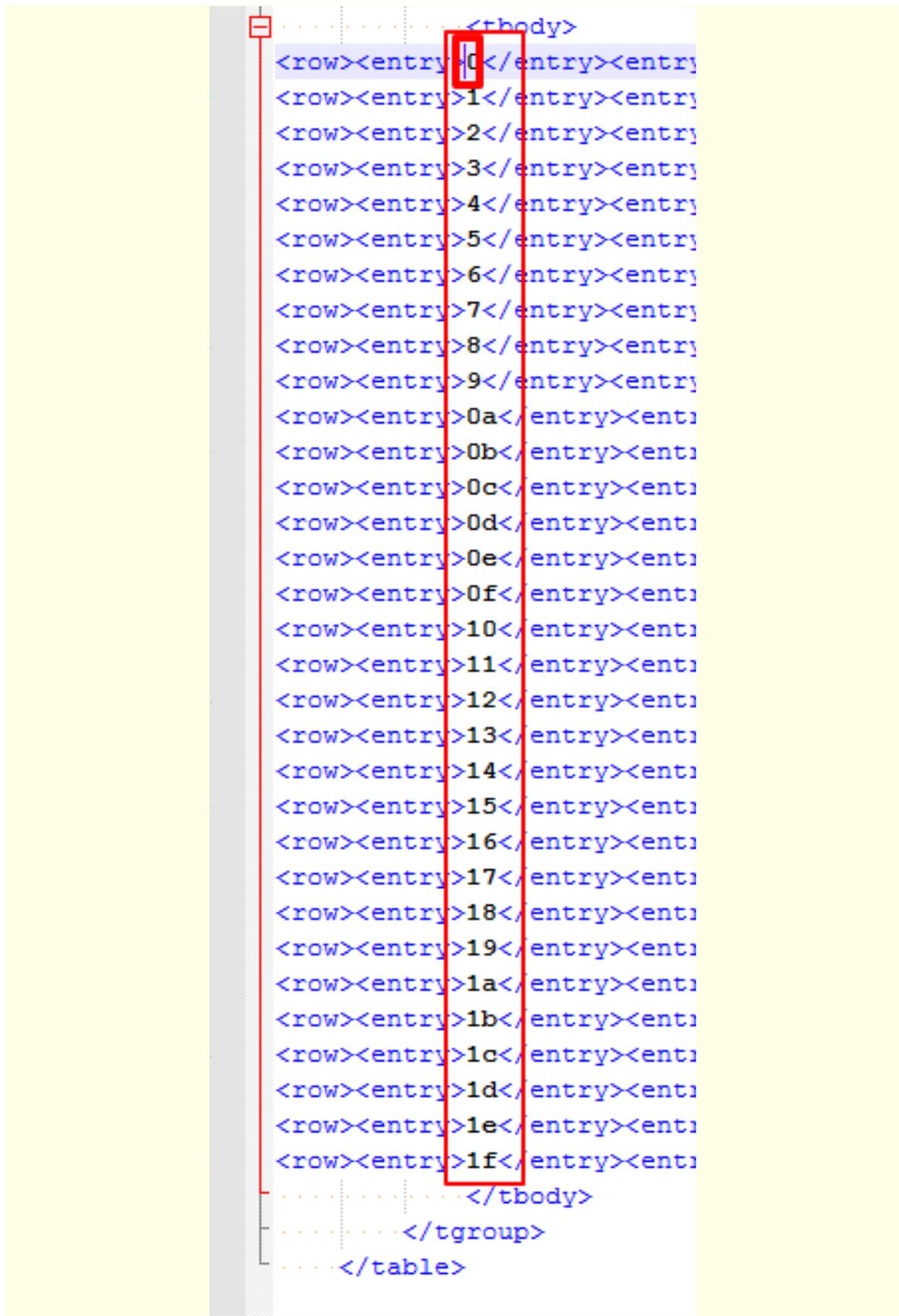
然后再用列模式去选取此部分内容：



然后Ctrl+C复制所选内容，再回到要粘贴的地方，同样先是进入列模式：

```
3 <thead>
4 <tr><td>十六进制</td><td>十进制</td><td>字符</td><td></td></tr>
5 <tr><td>十六进制</td><td>十进制</td><td>字符</td><td></td></tr>
6 </thead>
7 <tbody>
8 <tr><td></td><td></td><td></td><td></td></tr>
9 <tr><td></td><td></td><td></td><td></td></tr>
10 <tr><td></td><td></td><td></td><td></td></tr>
11 <tr><td></td><td></td><td></td><td></td></tr>
12 <tr><td></td><td></td><td></td><td></td></tr>
13 <tr><td></td><td></td><td></td><td></td></tr>
14 <tr><td></td><td></td><td></td><td></td></tr>
15 <tr><td></td><td></td><td></td><td></td></tr>
16 <tr><td></td><td></td><td></td><td></td></tr>
17 <tr><td></td><td></td><td></td><td></td></tr>
18 <tr><td></td><td></td><td></td><td></td></tr>
19 <tr><td></td><td></td><td></td><td></td></tr>
20 <tr><td></td><td></td><td></td><td></td></tr>
21 <tr><td></td><td></td><td></td><td></td></tr>
22 <tr><td></td><td></td><td></td><td></td></tr>
23 <tr><td></td><td></td><td></td><td></td></tr>
24 <tr><td></td><td></td><td></td><td></td></tr>
25 <tr><td></td><td></td><td></td><td></td></tr>
26 <tr><td></td><td></td><td></td><td></td></tr>
27 <tr><td></td><td></td><td></td><td></td></tr>
28 <tr><td></td><td></td><td></td><td></td></tr>
29 <tr><td></td><td></td><td></td><td></td></tr>
30 <tr><td></td><td></td><td></td><td></td></tr>
31 <tr><td></td><td></td><td></td><td></td></tr>
32 <tr><td></td><td></td><td></td><td></td></tr>
33 <tr><td></td><td></td><td></td><td></td></tr>
34 <tr><td></td><td></td><td></td><td></td></tr>
35 <tr><td></td><td></td><td></td><td></td></tr>
36 <tr><td></td><td></td><td></td><td></td></tr>
37 <tr><td></td><td></td><td></td><td></td></tr>
38 <tr><td></td><td></td><td></td><td></td></tr>
39 <tr><td></td><td></td><td></td><td></td></tr>
40 </tbody>
41 </tfoot>
42 </table>
```

然后再按Ctrl+V，这样才可以正确的将通过列模式选取的内容通过（Ctrl+V）粘贴到列模式所选取的范围内，即所选取的每一行的内容，粘贴到目标的每一行的位置：

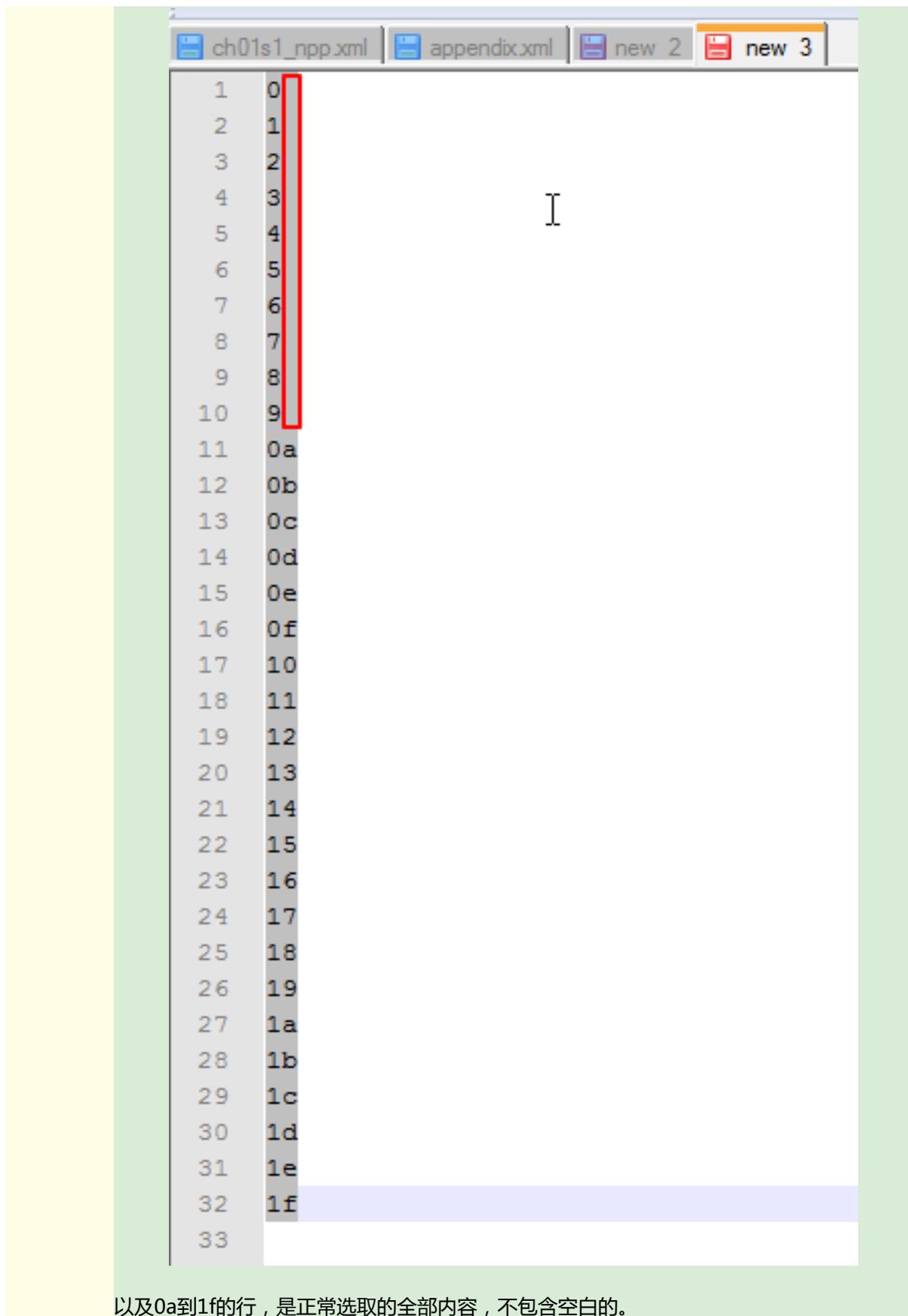


```
.....<tbody>
<row><entry>0</entry><entry>
<row><entry>1</entry><entry>
<row><entry>2</entry><entry>
<row><entry>3</entry><entry>
<row><entry>4</entry><entry>
<row><entry>5</entry><entry>
<row><entry>6</entry><entry>
<row><entry>7</entry><entry>
<row><entry>8</entry><entry>
<row><entry>9</entry><entry>
<row><entry>0a</entry><entry>
<row><entry>0b</entry><entry>
<row><entry>0c</entry><entry>
<row><entry>0d</entry><entry>
<row><entry>0e</entry><entry>
<row><entry>0f</entry><entry>
<row><entry>10</entry><entry>
<row><entry>11</entry><entry>
<row><entry>12</entry><entry>
<row><entry>13</entry><entry>
<row><entry>14</entry><entry>
<row><entry>15</entry><entry>
<row><entry>16</entry><entry>
<row><entry>17</entry><entry>
<row><entry>18</entry><entry>
<row><entry>19</entry><entry>
<row><entry>1a</entry><entry>
<row><entry>1b</entry><entry>
<row><entry>1c</entry><entry>
<row><entry>1d</entry><entry>
<row><entry>1e</entry><entry>
<row><entry>1f</entry><entry>
.....</tbody>
.....</tgroup>
.....</table>
```



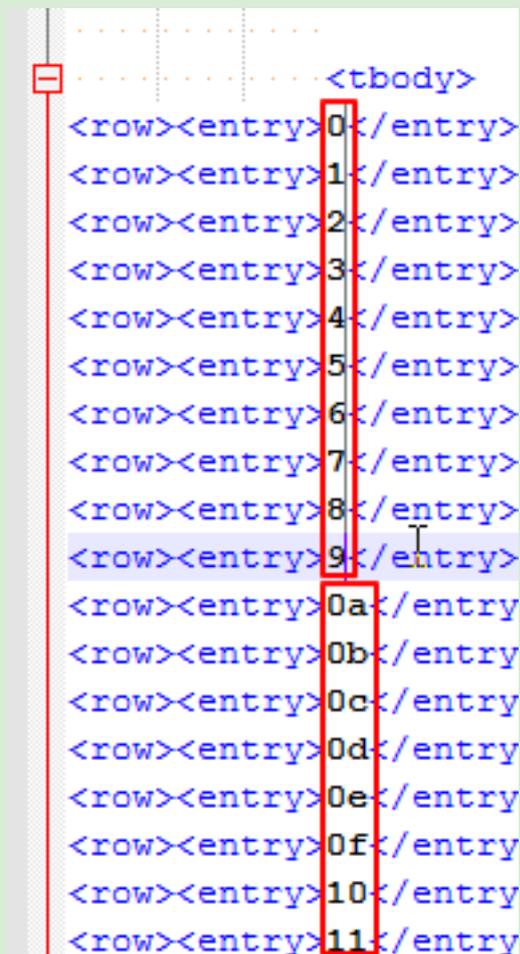
列模式粘贴的时候，会自动帮你去掉所选内容中的空白处，即可以调整被粘贴后的内容的宽度

此处，可以注意到，原先列模式去选取内容的时候，0到9的那些行，（至少显示出来的效果中）是包含了多余的空格的：



以及0a到1f的行，是正常选取的全部内容，不包含空白的。

而粘贴出来的效果，可以看到，对应的0到9的行，是没有多余的空格，是一个字符的宽度的，没有被变成2个字符的宽度：



```
.....<tbody>
<row><entry>0</entry>
<row><entry>1</entry>
<row><entry>2</entry>
<row><entry>3</entry>
<row><entry>4</entry>
<row><entry>5</entry>
<row><entry>6</entry>
<row><entry>7</entry>
<row><entry>8</entry>
<row><entry>9</entry>
<row><entry>0a</entry>
<row><entry>0b</entry>
<row><entry>0c</entry>
<row><entry>0d</entry>
<row><entry>0e</entry>
<row><entry>0f</entry>
<row><entry>10</entry>
<row><entry>11</entry>
```

即，列编辑的时候，所选择的内容中，显示出来的效果中，好像是包含了空格，实际没有，所以在粘贴出来后，不会多余出空格。

话句话说，列编辑模式下，选取内容，和粘贴内容，会自动帮你计算好对应的内容的，不会（像我以为的）多余的插入（那些在选取时显示出来的多余的）空格的

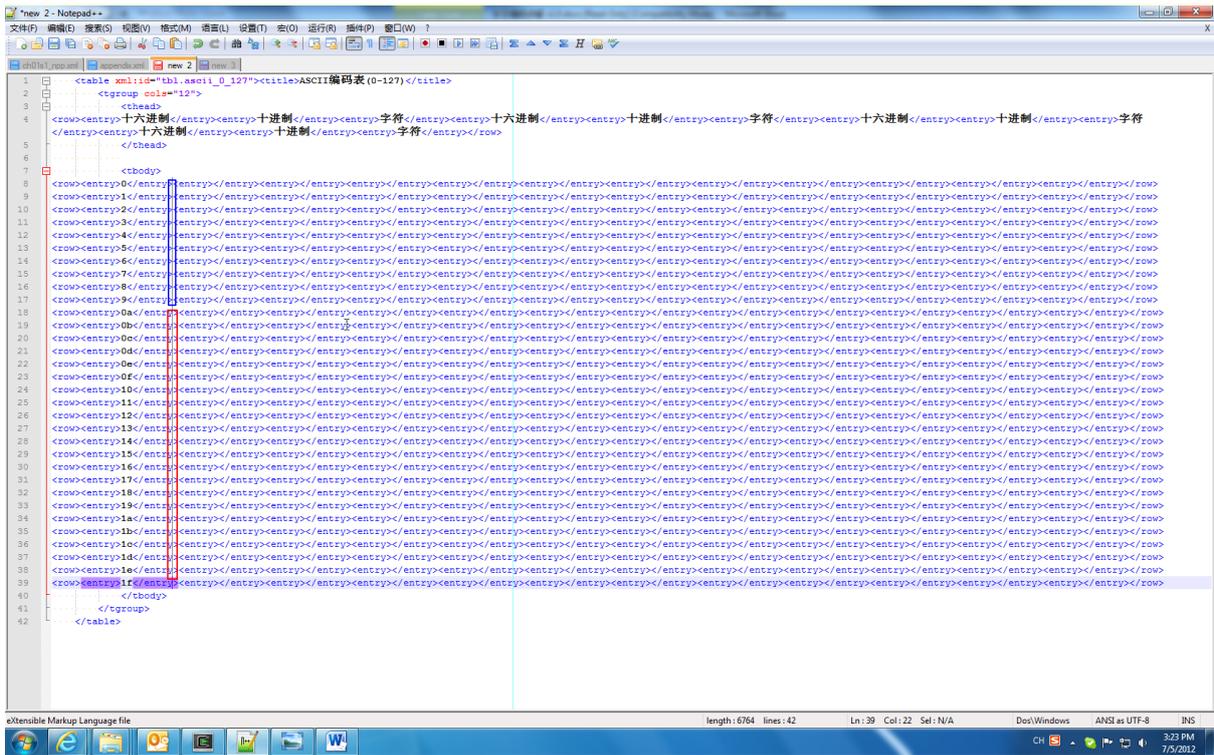
所以，这方面，个人觉得，其做得还是很人性化，或者说很符合用户需求的。

按照上述方法，你可以去接着一列的，去粘贴word中别的列的内容。

但是，如果你接着这么做的话，你却发现，有些问题了。

因为经过上面第一列的粘贴，上面的0到9的那些行是一个字符宽度，而0a到1f却是2个字符宽度，这导致接下来的想要选取第二列的所有的entry，以进入列编辑模式的话，就无法正确全部选择对应的位置了：

Notepad++ 的功能详解



即，0到9列，是可以正确的选择了两个"<entry>"中间的，但是0a到1f的列，却都选择了y和>之间了，位置错了。

如此，想要接着实现列拷贝粘贴的话，一般人所能想到的，那就只是，先拷贝粘贴0到9列的，然后再拷贝粘贴0a到1f列的。

如果这么做的话，其对于此处只是被分为2个不同的列的位置去操作，也还算能接受。

但是如果被粘贴的内容，像后面的同一列中，既有1个字符宽度的，又有2个和3个字符宽度的，甚至其他更多字符的，那么一个完全的列，就被分为多个不连续的，可供列编辑操作的列了。

那结果就又几乎回到了之前的手工慢慢的复制粘贴的效果了。

所以，还是要找到更好的解决办法，尽量实现尽可能多的列，都实现一次性的列操作，这样才能真正提高效率。

后来，偶发现了一个办法，那就是，可以先操作右边的列，然后依次向左处理每一列，这样就可以避免此问题。

之所以想到如此去做，是因为，右边的列，即使每一列的内容宽度不同，但是不会影响到左边的列的对齐，所以，就可以规避此问题了。

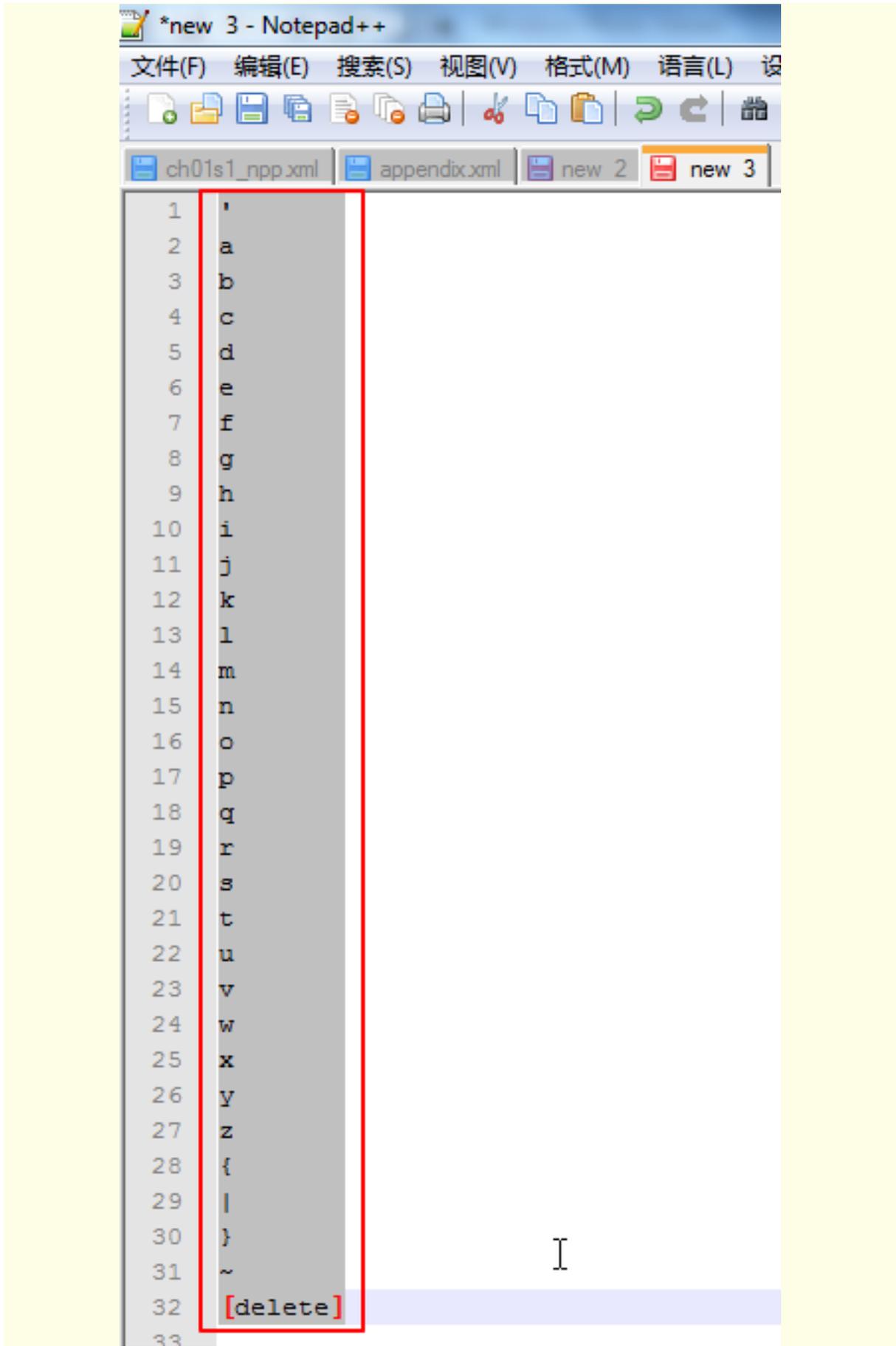
其具体做法很简单，只是换个顺序处理而已：

在word中，先复制最后一列的内容：

图表 1 ASCII 编码表

十六进制	十进制	字符	十六进制	十进制	字符	十六进制	十进制	字符	十六进制	十进制	字符
0	0	NUL	20	32	[space]	40	64	@	60	96	'
1	1	SOH	21	33	!	41	65	A	61	97	a
2	2	STX	22	34	"	42	66	B	62	98	b
3	3	ETX	23	35	#	43	67	C	63	99	c
4	4	EOT	24	36	\$	44	68	D	64	100	d
5	5	ENQ	25	37	%	45	69	E	65	101	e
6	6	ACK	26	38	&	46	70	F	66	102	f
7	7	BEL	27	39	`	47	71	G	67	103	g
8	8	BS	28	40	(48	72	H	68	104	h
9	9	HT	29	41)	49	73	I	69	105	i
0a	10	LF	2a	42	*	4a	74	J	6a	106	j
0b	11	VT	2b	43	+	4b	75	K	6b	107	k
0c	12	FF	2c	44	,	4c	76	L	6c	108	l
0d	13	CR	2d	45	-	4d	77	M	6d	109	m
0e	14	SO	2e	46	.	4e	78	N	6e	110	n
0f	15	SI	2f	47	/	4f	79	O	6f	111	o
10	16	DLE	30	48	0	50	80	P	70	112	p
11	17	DC1	31	49	1	51	81	Q	71	113	q
12	18	DC2	32	50	2	52	82	R	72	114	r
13	19	DC3	33	51	3	53	83	S	73	115	s
14	20	DC4	34	52	4	54	84	T	74	116	t
15	21	NAK	35	53	5	55	85	U	75	117	u
16	22	SYN	36	54	6	56	86	V	76	118	v
17	23	ETB	37	55	7	57	87	W	77	119	w
18	24	CAN	38	56	8	58	88	X	78	120	x
19	25	EM	39	57	9	59	89	Y	79	121	y
1a	26	SUB	3a	58	:	5a	90	Z	7a	122	z
1b	27	ESC	3b	59	;	5b	91	[7b	123	{
1c	28	FS	3c	60	<	5c	92	\	7c	124	
1d	29	GS	3d	61	=	5d	93]	7d	125	}
1e	30	RS	3e	62	>	5e	94	^	7e	126	~
1f	31	US	3f	63	?	5f	95	_	7f	127	[delete]

同样的，粘贴到新建的Notepad++的页面中，并以列模式去选取该内容：




```
<entry></entry><entry>40</entry><entry>64<
<entry></entry><entry>41</entry><entry>65<
<entry></entry><entry>42</entry><entry>66<
<entry></entry><entry>43</entry><entry>67<
<entry></entry><entry>44</entry><entry>68<
<entry></entry><entry>45</entry><entry>69<
<entry></entry><entry>46</entry><entry>70<
<entry></entry><entry>47</entry><entry>71<
<entry></entry><entry>48</entry><entry>72<
<entry></entry><entry>49</entry><entry>73<
<entry></entry><entry>4a</entry><entry>74<
<entry></entry><entry>4b</entry><entry>75<
<entry></entry><entry>4c</entry><entry>76<
<entry></entry><entry>4d</entry><entry>77<
<entry></entry><entry>4e</entry><entry>78<
<entry></entry><entry>4f</entry><entry>79<
<entry></entry><entry>50</entry><entry>80<
<entry></entry><entry>51</entry><entry>81<
<entry></entry><entry>52</entry><entry>82<
<entry></entry><entry>53</entry><entry>83<
<entry></entry><entry>54</entry><entry>84<
```

此处，看起来是隔行的列选择，实际仍是每个行的列选择，即，其实是不影响继续列操作的。

和上述的隔行选择的效果类似的，是处理到最后一列的时候：



```
<row><entry>十六进制</entry><entry>十进制</en
</entry><entry>十六进制</entry><entry>十进制<
..... </thead>
.....
..... <tbody>
<row><entry></entry><entry>0</entry><entry>NUL
</entry><entry>'</entry></row>
<row><entry></entry><entry>1</entry><entry>SOH
</entry></row>
<row><entry></entry><entry>2</entry><entry>STX
</entry></row>
<row><entry></entry><entry>3</entry><entry>ETX
</entry></row>
<row><entry></entry><entry>4</entry><entry>EOT
</entry></row>
<row><entry></entry><entry>5</entry><entry>ENQ
</entry></row>
<row><entry></entry><entry>6</entry><entry>ACK
</entry></row>
<row><entry></entry><entry>7</entry><entry>BEL
</entry></row>
<row><entry></entry><entry>8</entry><entry>BS<
</entry></row>
<row><entry></entry><entry>9</entry><entry>HT<
</entry></row>
<row><entry></entry><entry>10</entry><entry>LF
</entry></row>
<row><entry></entry><entry>11</entry><entry>VT
</entry></row>
<row><entry></entry><entry>12</entry><entry>FF
</entry></row>
<row><entry></entry><entry>13</entry><entry>CR
</entry></row>
<row><entry></entry><entry>14</entry><entry>SO
```

实际上，看起来“穿”过了字符串“entry”，而实际上，是不影响的entry字符串内容的。

如此继续操作，就处理完毕所有的列了：

```

<tbody>
<tr><td>十六进制</td><td>十进制</td><td>字符</td><td>十六进制</td><td>十六进制</td><td>十进制</td><td>字符</td></tr>
</thead>
<tbody>
<tr><td>0</td><td>0</td><td>NUL</td><td>20</td><td>32</td><td>er</td><td>'</td></tr>
<tr><td>1</td><td>1</td><td>SOH</td><td>21</td><td>33</td><td>er</td></tr>
<tr><td>2</td><td>2</td><td>STX</td><td>22</td><td>34</td><td>er</td></tr>
<tr><td>3</td><td>3</td><td>ETX</td><td>23</td><td>35</td><td>er</td></tr>
<tr><td>4</td><td>4</td><td>EOT</td><td>24</td><td>36</td><td>er</td></tr>
<tr><td>5</td><td>5</td><td>ENQ</td><td>25</td><td>37</td><td>er</td></tr>
<tr><td>6</td><td>6</td><td>ACK</td><td>26</td><td>38</td><td>er</td></tr>
<tr><td>7</td><td>7</td><td>BEL</td><td>27</td><td>39</td><td>er</td></tr>
<tr><td>8</td><td>8</td><td>BS</td><td>28</td><td>40</td><td>er</td></tr>
<tr><td>9</td><td>9</td><td>HT</td><td>29</td><td>41</td><td>er</td></tr>
<tr><td>0a</td><td>10</td><td>LF</td><td>2a</td><td>42</td><td>er</td><td>j</td></tr>
<tr><td>0b</td><td>11</td><td>VT</td><td>2b</td><td>43</td><td>er</td><td>k</td></tr>
<tr><td>0c</td><td>12</td><td>FF</td><td>2c</td><td>44</td><td>er</td><td>l</td></tr>
<tr><td>0d</td><td>13</td><td>CR</td><td>2d</td><td>45</td><td>er</td><td>m</td></tr>
<tr><td>0e</td><td>14</td><td>SO</td><td>2e</td><td>46</td><td>er</td><td>n</td></tr>
<tr><td>0f</td><td>15</td><td>SI</td><td>2f</td><td>47</td><td>er</td><td>o</td></tr>
<tr><td>10</td><td>16</td><td>DLE</td><td>30</td><td>48</td><td>er</td><td>p</td></tr>
<tr><td>11</td><td>17</td><td>DC1</td><td>31</td><td>49</td><td>er</td><td>q</td></tr>
<tr><td>12</td><td>18</td><td>DC2</td><td>32</td><td>50</td><td>er</td><td>r</td></tr>
<tr><td>13</td><td>19</td><td>DC3</td><td>33</td><td>51</td><td>er</td><td>s</td></tr>
<tr><td>14</td><td>20</td><td>DC4</td><td>34</td><td>52</td><td>er</td><td>t</td></tr>

```

这也就是我们最开始所看的效果。

而如此的列操作，其实只是对于每一列，去复制粘贴一下，就可以实现，整个列的内容录入了。

通过此列操作实现的多列同时录入，不仅效率很高，而且还不容易出错。

总之，有效利用列编辑模式，可以大幅度的提高做事情效率和准确度。

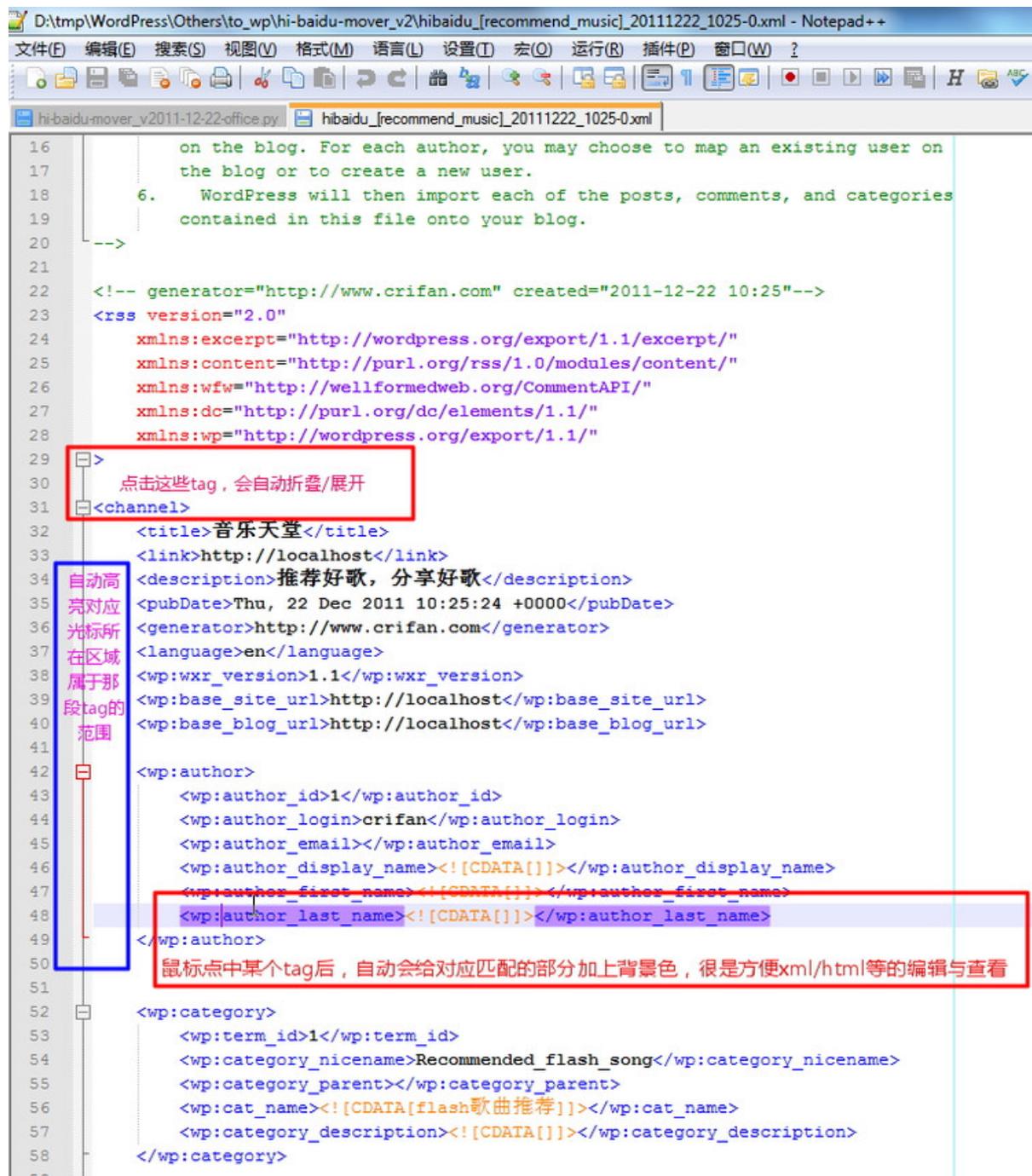
3.7. Notepad++的代码折叠

代码折叠，即，Notepad++对于不同语言，在语法高亮基础上，还能检测出对应的C，Python等语言的单个的函数，xml等语言的标签，

然后对这些独立的代码部分，最开端，有个加减号的可点击的小图标，对应的可以展开或折叠其对应的代码。

截图说明：

图 3.1. XML文件中的Notepad++的代码折腾功能



估计很多人可能和我最开始的想法一样，觉得此代码折叠功能，好像没啥实际用途。

直到有一天，我发现，代码折叠，至少对于更加清晰明确xml文件的架构和层次，有不小的帮助。

比如，我当前的xml文件，里面有很多个章节，很多行代码，而如果没有代码折叠，则每次定位到某个章节的时候，都需要找半天。

而有了代码折腾后，把其他暂时不关心的章节的内容，先折叠起来：

```

69 <sect2><title>使用Notepad++前要了解的知识</title>
99
100 <sect2 xml:id="npp.plugins"><title>Notepad++的插件</title>
206
207 <sect2 xml:id="npp.installation"><title>Notepad++的安装过程</title>
430
431 <sect2 xml:id="npp.functions"><title>Notepad++的功能详解</title>
432 <para>下面就详细介绍Notepad++中的N多好用易用的功能:</para>
433 <para>下列功能的排名先后顺序,由个人所认为的该功能的重要性高低而决定</para>
434
435 <sect3 xml:id="npp.code_highlight"><title>Notepad++的语法高亮</title>
529
530 <sect3 xml:id="npp.code_collapse"><title>Notepad++的代码折叠</title>
531 <para>代码折叠,即,Notepad++对于不同语言,在语法高亮基础上,还能检测出对应的C,Python等语言
532 <para>然后对这些独立的代码部分,最开始端,有个加减号的可点击的小图标,对应的可以展开或折叠:
533 <para>截图说明:</para>
534 <figure xml:id="fg.npp_code_collapse"><title>XML文件中的Notepad++的代码折腾功能</title>
535 <mediaobject>
536 <imageobject role="html"><imagedata fileref="http://www.crifan.com/files/pic/work_and_jc
537 <imageobject role="fo"><imagedata fileref="http://www.crifan.com/files/pic/work_and_jc
538 </mediaobject>
539 </figure>
540 <para>估计很多人可能和我最开始的想法一样,觉得此代码折叠功能,好像没啥实际用途.</para>
541 <para>直到有一天,我发现,代码折叠,至少对于更加清晰明确xml文件的架构和层次,有不小的帮助。
542 <para>比如,我当前的xml文件,里面有很多个章节,很多行代码,而如果没有代码折叠,则每次定位到
543 <para>而有了代码折腾后,把其他暂时不关心的章节的内容,先折叠起来:</para>
544 <para></para>
545 <para></para>

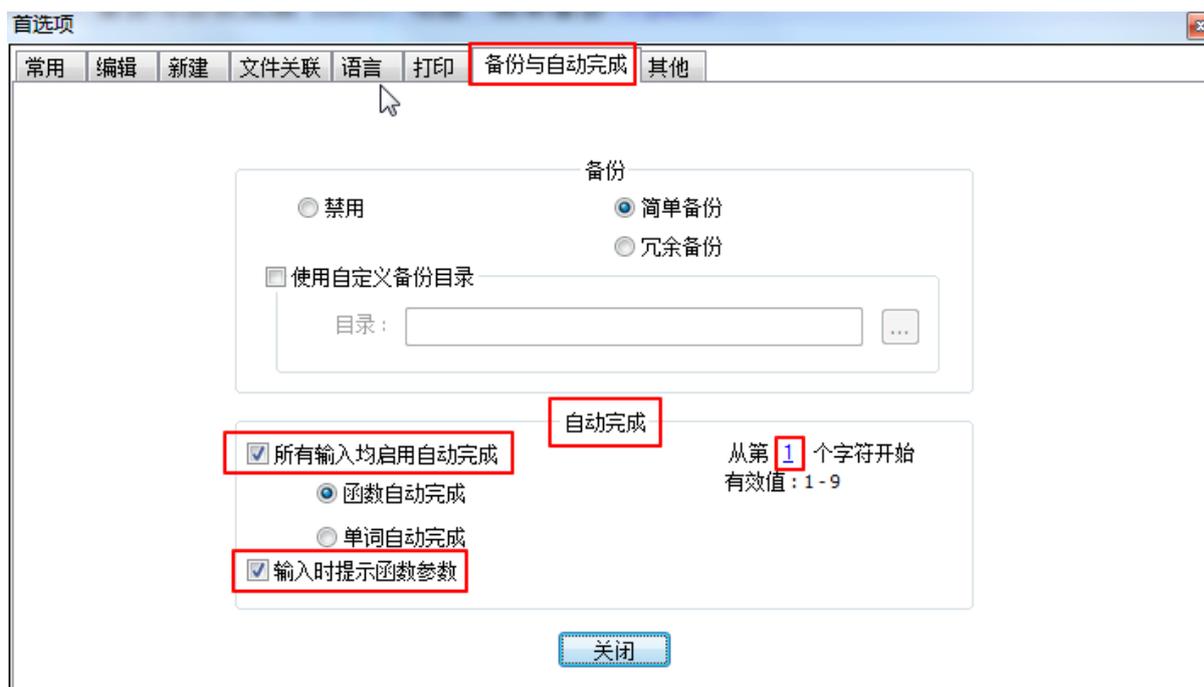
```

这样,立刻就可以把几百行的内容,暂时隐藏起来,使得整个xml的架构层次,很清晰明了,方便后续的查看编辑相关的内容。

如此,越加发现,其实很多功能,真的得到你需要用到,才会发现如此的有用。

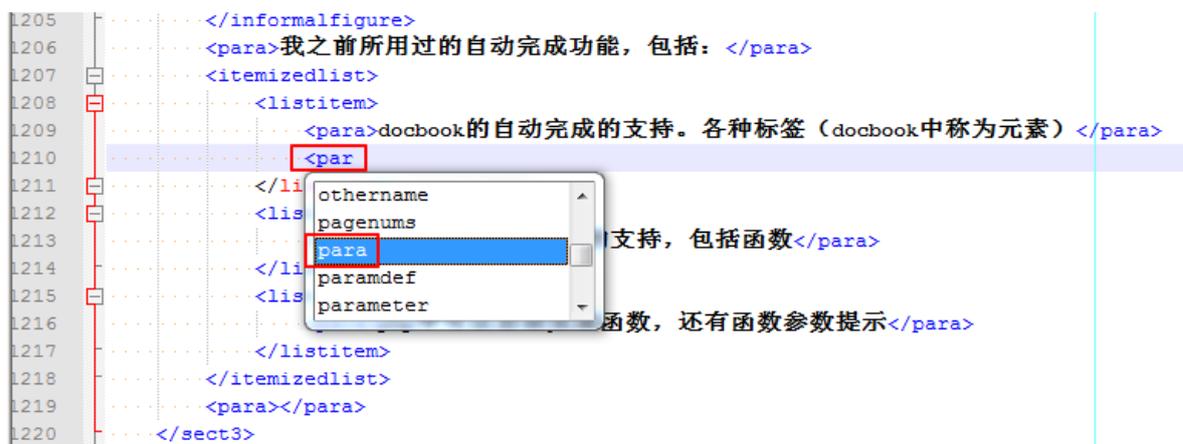
3.8. Notepad++的自动完成

设置(T) ⇒ 首选项... ⇒ 备份 ⇒ 备份与自动完成 ⇒ 自动完成 ⇒ 勾选 "所有输入均启用自动完成", "函数自动完成", "输入时提示函数参数"

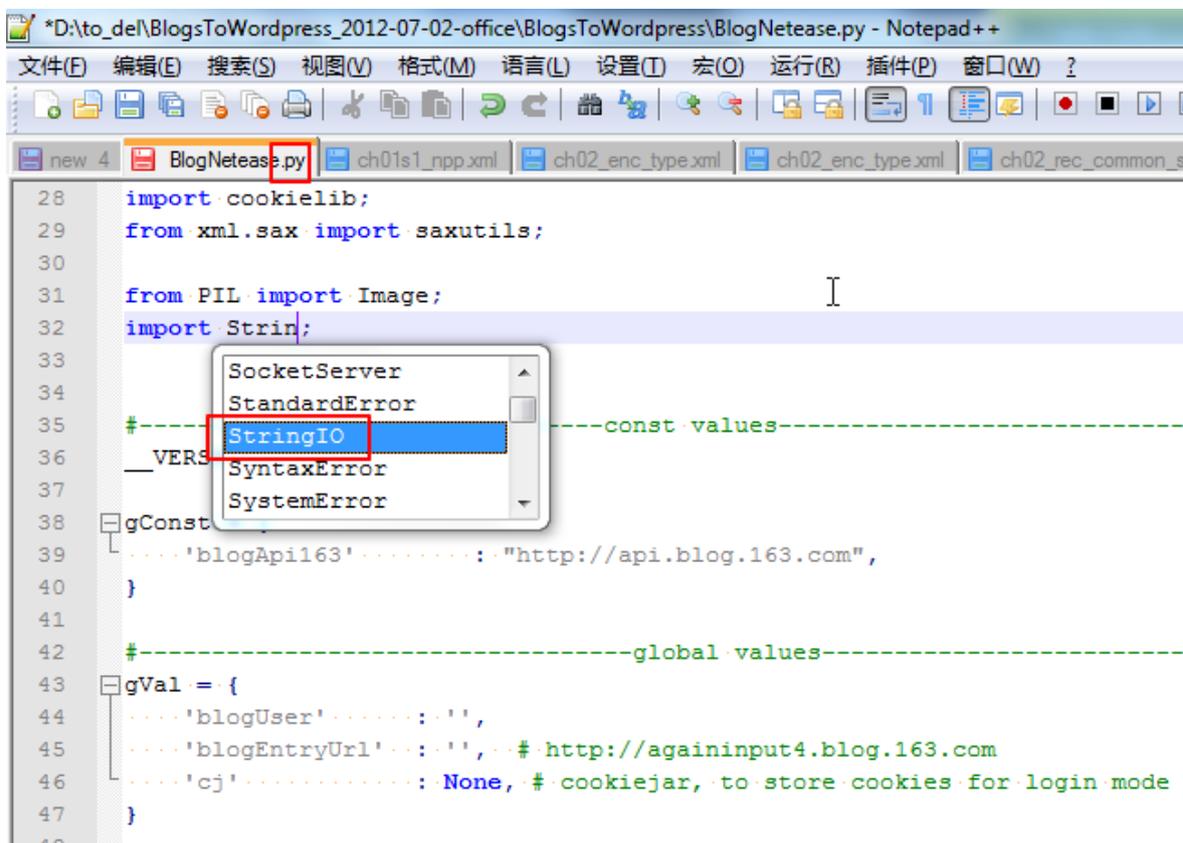


我之前所用过的自动完成功能,至少包括:

- docbook的自动完成的支持。各种标签（docbook中称为元素）的自动完成：



- python的自动完成的支持，包括函数和库：



- php中可以自动完成函数：

```

2727 .....}
2728 .....}
2729 .....subs
2730 .....
2731 .....str_split
2732 .....str_word_count
2733 .....substr
2734 .....substr_compare
2735 .....substr_count
2736 .....if(substr($curPath, -1) != "/")
2737 .....{
2738 .....    $curPath = $curPath."/";
2739 .....}
2740 .....echo "curPath=" . $curPath."<br />";
2741 .....
2742 .....$isDir = is_dir($curPath);
2743 .....echo "isDir=" . $isDir."<br />";
2744 .....
2745 .....$fullPath = $rootPath.$curPath;
2746 .....//$fullPath = $curPath;
2747 .....echo "fullPath=" . $fullPath."<br />";
2748 ...../* .....echo "----test dir() ----<br />";

```

还有函数参数提示：

```

2728 .....
2729 .....substr
2730 .....string|false substr (string str, int start, [int length])
2731 .....//$curPath = $p->GetUrl();
2732 .....//$curPath = "../files/doc/docbook";
2733 .....//$curPath = "../files/doc/docbook/";
2734 .....$curPath = "files/doc/docbook";
2735 .....echo "curPath=" . $curPath."<br />";
2736 .....if(substr($curPath, -1) != "/")
2737 .....{
2738 .....    $curPath = $curPath."/";
2739 .....}
2740 .....echo "curPath=" . $curPath."<br />";
2741 .....
2742 .....$isDir = is_dir($curPath);
2743 .....echo "isDir=" . $isDir."<br />";
2744 .....
2745 .....$fullPath = $rootPath.$curPath;
2746 .....//$fullPath = $curPath;
2747 .....echo "fullPath=" . $fullPath."<br />";
2748 ...../* .....echo "----test dir() ----<br />";

```

此自动完成的好处，还是很明显的：

至少不用让你完全记住每种语言的所有函数（关键字）的全部单词，只需要写出部分字母

剩下的字母就可以通过Notepad++帮你自动匹配，就可以通过上下键选择所需项，然后按Tab，实现自动补全了。

就不用全部一个个字母的输入了，还是很方便使用的。



Notepad++ 自动完成功能内在实现的逻辑

之前没太注意，所以也不太清楚，对于此自动完成的功能，到底是如何实现的，或者说，想要搞懂，Notepad++实现自动完成的大概逻辑是什么样的。

后来，才大概想明白：

因为在Notepad++在安装过程中，有个选项：[Notepad++安装选项：Auto Complete Files](#)

就是指的是自动完成的功能，需要支持哪些类型的文件。

此处，应该就是会去安装对应的数据库，然后对于打开的文件，通过文件名后缀（或者文件头的声明等），识别出当前文件的语言，比如c，Python，Docbook等，

之后，再根据当前所输入的字符，动态匹配当前语言的自动完成所涉及的数据库，显示出所匹配的函数，以及函数提示等。

至此，才清楚，其大概实现逻辑。

3.8.1. Notepad++的单词自动完成

之前，是知道了有函数自动完成。不过后来，在折腾Sublime

[【crifan推荐】一款相对不错的文本编辑器：Sublime Text 2](#)⁹

的过程中，发现其中有个很好用的功能：文件内的，函数，变量等内容的，自动完成。

之前不知道这个就个东东叫做，单词自动完成，现在才知道。

然后后来也发现了，Notepad++中，有个自动完成：

设置(T) ⇒ 首选项... ⇒ 备份 ⇒ 备份与自动完成 ⇒ 自动完成 ⇒ 勾选 "所有输入均启用自动完成", "单词自动完成"

但是，由于当前所用的6.2.3中的版本的单次自动完成有bug，无法正常显示列表：

[#4056 Word auto-completion is not working in NP++ 6.1.6 or late](#)¹⁰

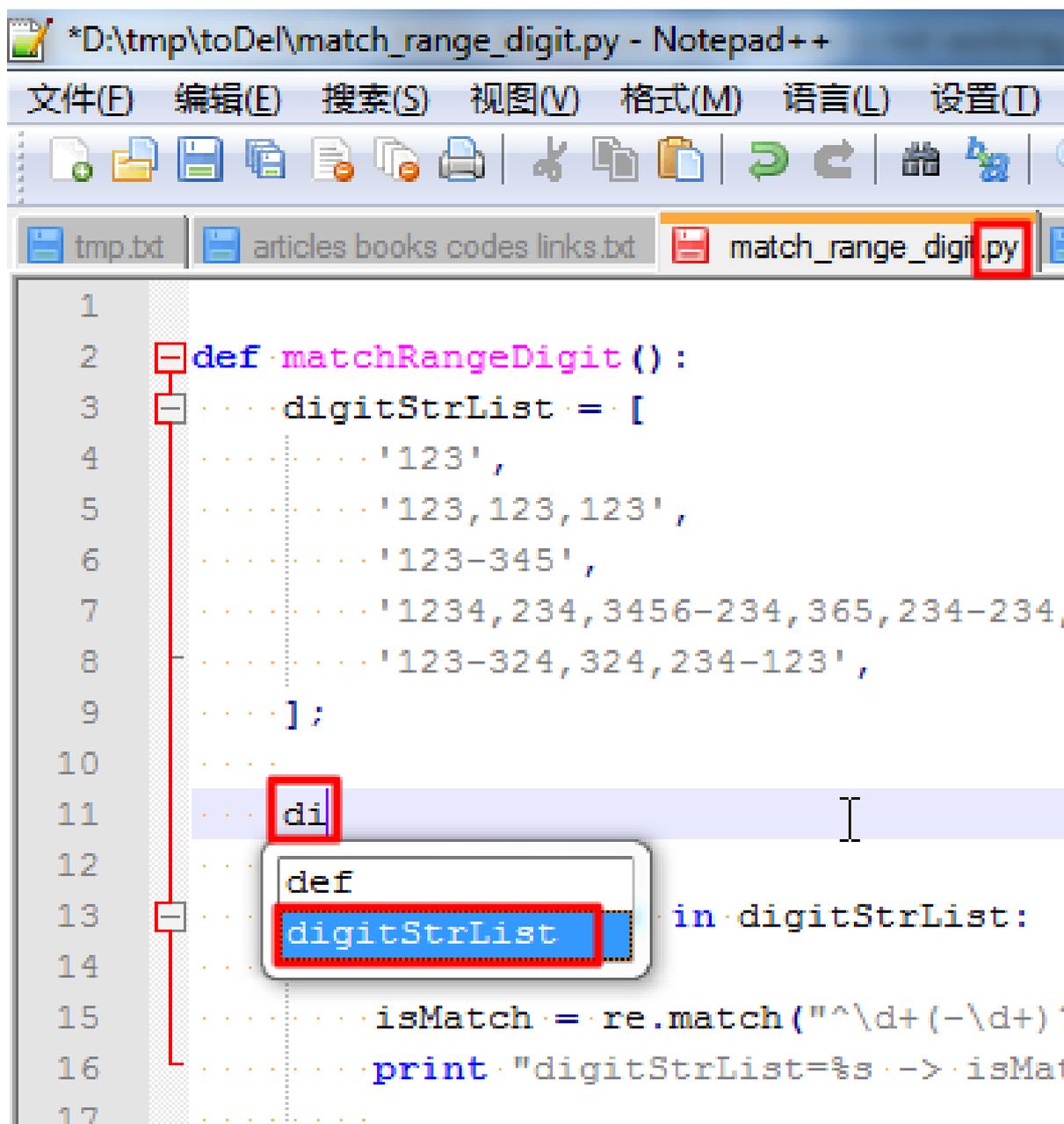
使得无法使用。

但是巧的是，Notepad++最近又发布了6.2.3的版本，修复了此bug，所以，刚去更新了最新6.2.3，然后就可以实现单词的自动完成的效果了：

比如某个Python文件内的变量的自动完成：

⁹ http://www.crifan.com/recommend_a_relative_good_text_editor_sublime_2

¹⁰ <http://sourceforge.net/p/notepad-plus/bugs/4056/>



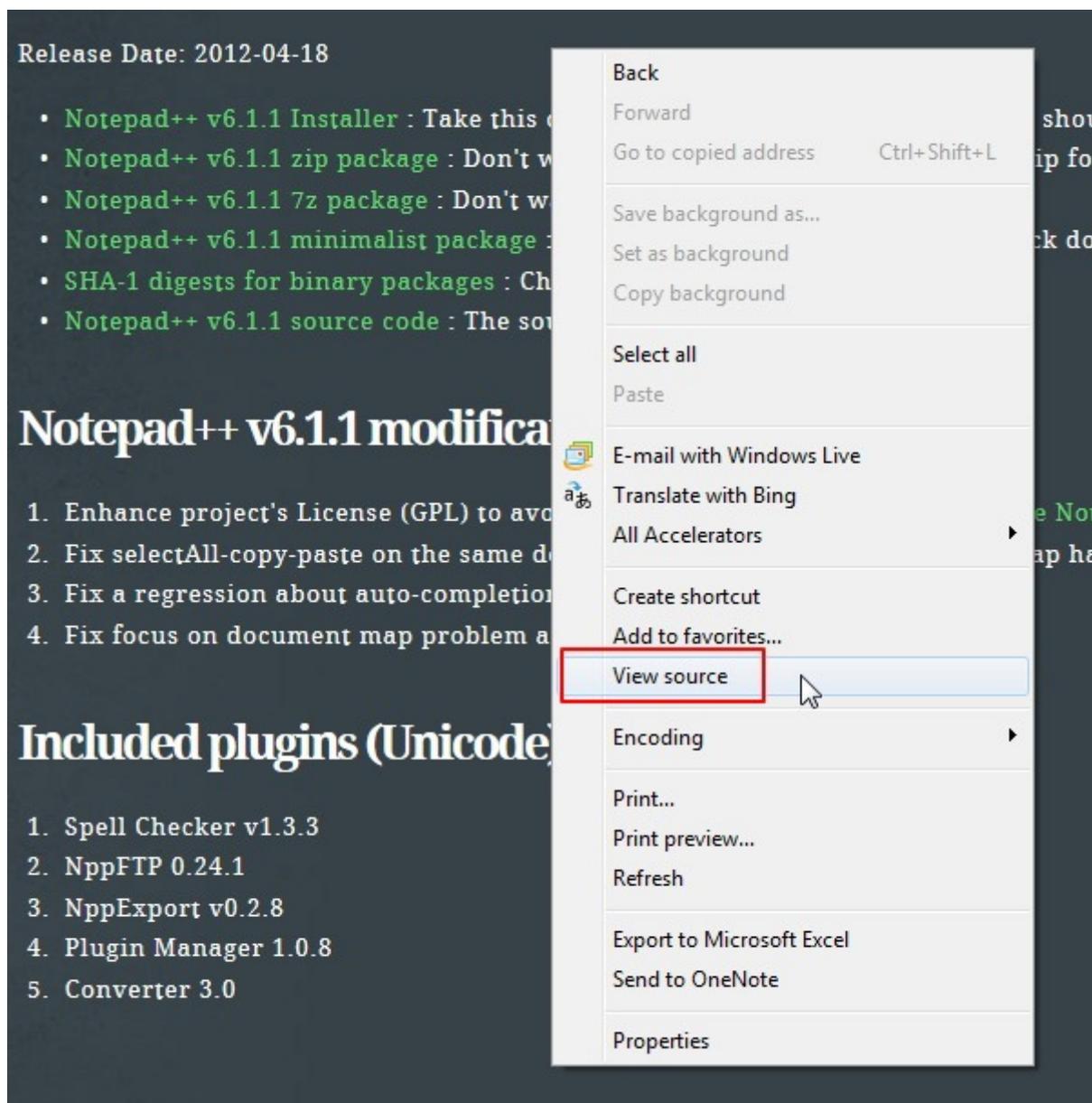
比如当前文件内部的，汉字都可以自动完成的：



3.9. Notepad++的默认HTML查看器

前面已经解释过了，在第 2.5 节 “Notepad++安装：选择组件” 的时候，就有个参数选择，可以把 Notepad++ 设置为默认HTML的查看器的：[Notepad++安装选项：As default html viewer](#)

这样，以后用浏览器查看某网页，想要查看其html源码的话，右键点击，选择 View Source：



就可以用Notepad++来查看HTML源码了：

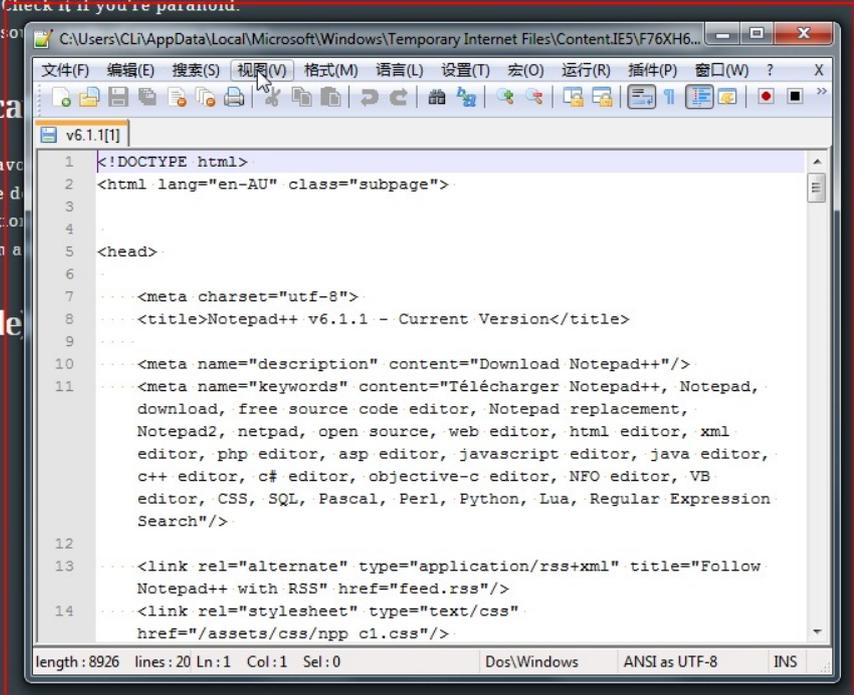
- Notepad++ v6.1.1 Installer : Take this one if you have no idea which one you should take.
- Notepad++ v6.1.1 zip package : Don't want to use installer? Check this one (zip format).
- Notepad++ v6.1.1 7z package : Don't want to use installer? 7z format.
- Notepad++ v6.1.1 minimalist package : No theme, no plugin, no updater, quick download and play directly. 7z format.
- SHA-1 digests for binary packages : Check it if you're paranoid.
- Notepad++ v6.1.1 source code : The source code is available on GitHub.

Notepad++ v6.1.1 modifications

1. Enhance project's License (GPL) to avoid legal issues.
2. Fix selectAll-copy-paste on the same document.
3. Fix a regression about auto-completion.
4. Fix focus on document map problem.

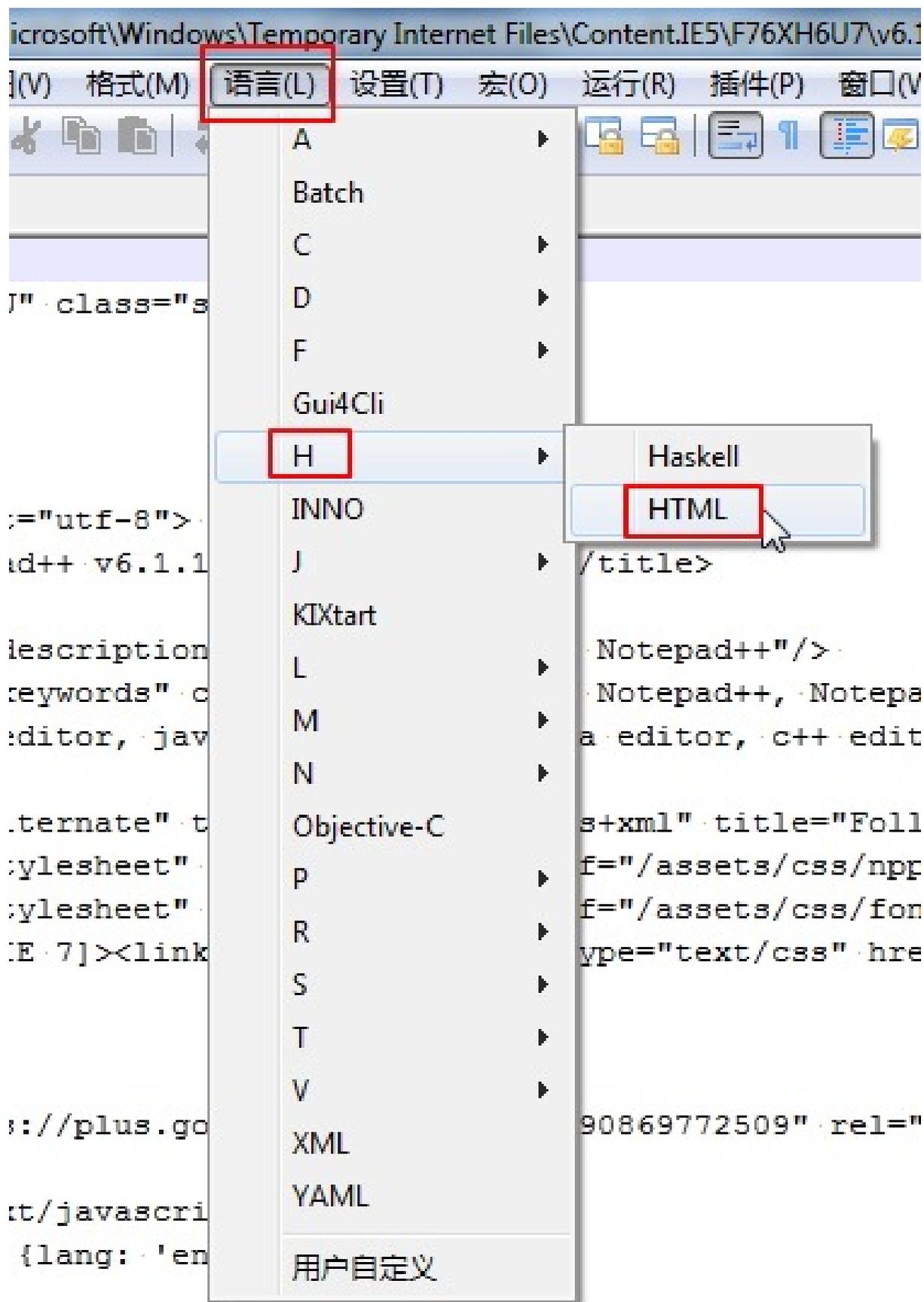
Included plugins (Unicode aware)

1. Spell Checker v1.3.3
2. NppFTP 0.24.1
3. NppExport v0.2.8
4. Plugin Manager 1.0.8
5. Converter 3.0

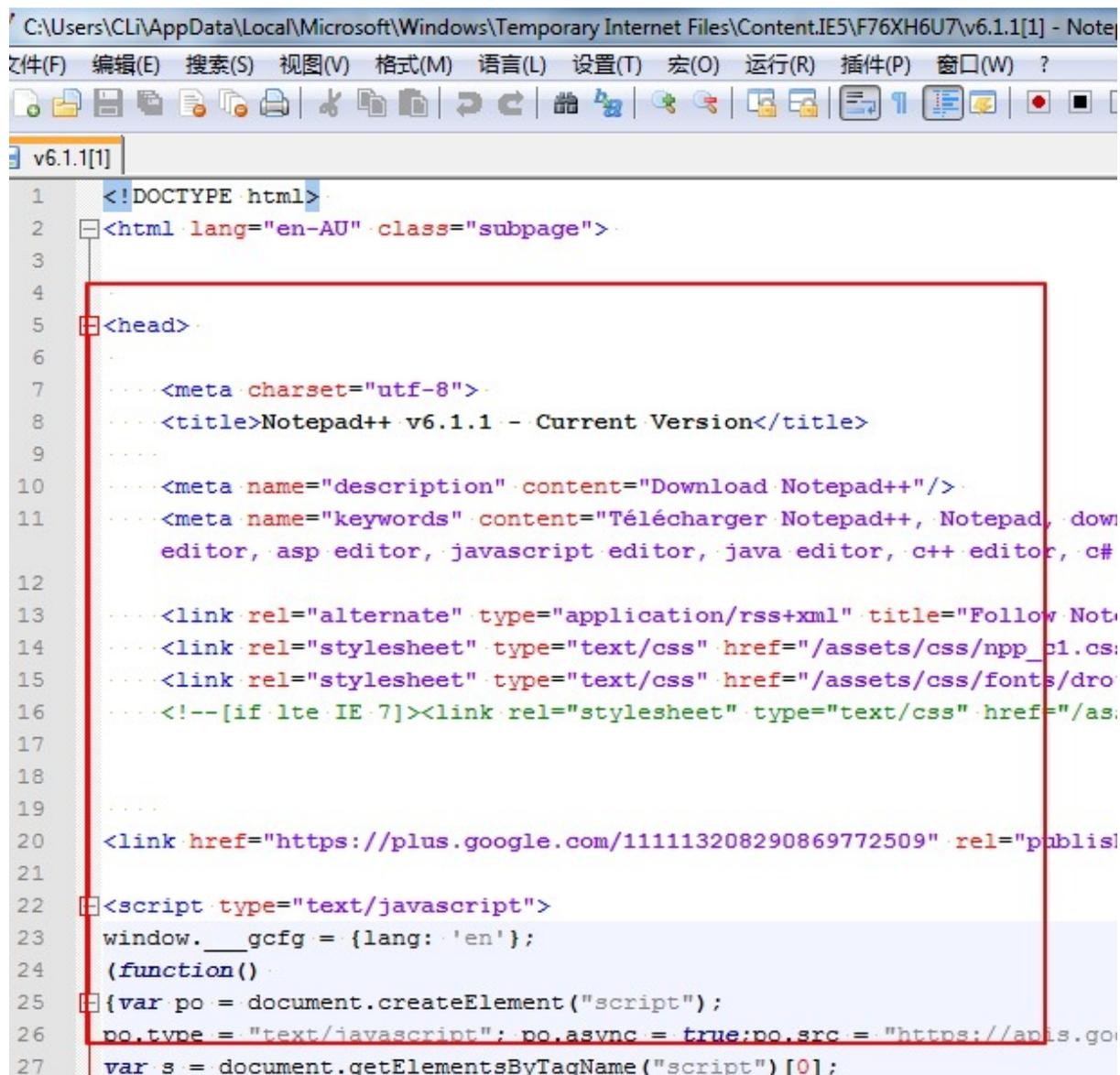


可以看到，其显示出来的效果，没有HTML高亮，可以自己手动设置一下：

语言(L) ⇒ H ⇒ HTML



就可以高亮的显示HTML文件了：



```
1 <!DOCTYPE html>
2 <html lang="en-AU" class="subpage">
3
4
5 <head>
6
7 <meta charset="utf-8">
8 <title>Notepad++ v6.1.1 - Current Version</title>
9
10 <meta name="description" content="Download Notepad++"/>
11 <meta name="keywords" content="Télécharger Notepad++, Notepad, down
12 editor, asp editor, javascript editor, java editor, c++ editor, c#
13
14 <link rel="alternate" type="application/rss+xml" title="Follow Not
15 <link rel="stylesheet" type="text/css" href="/assets/css/npp_cp1.cs
16 <!--[if lte IE 7]><link rel="stylesheet" type="text/css" href="/as
17
18
19
20 <link href="https://plus.google.com/111113208290869772509" rel="publis
21
22 <script type="text/javascript">
23 window.__gcf = {lang: 'en'};
24 (function()
25 {var po = document.createElement("script");
26 po.type = "text/javascript"; po.async = true; po.src = "https://apis.go
27 var s = document.getElementsByTagName("script")[0];
```

将Notepad++用作为HTML默认编辑器，主要适合于网络相关开发人员。

3.10. Notepad++支持显示回车符，换行符，TAB键，行首，行尾等特殊字符

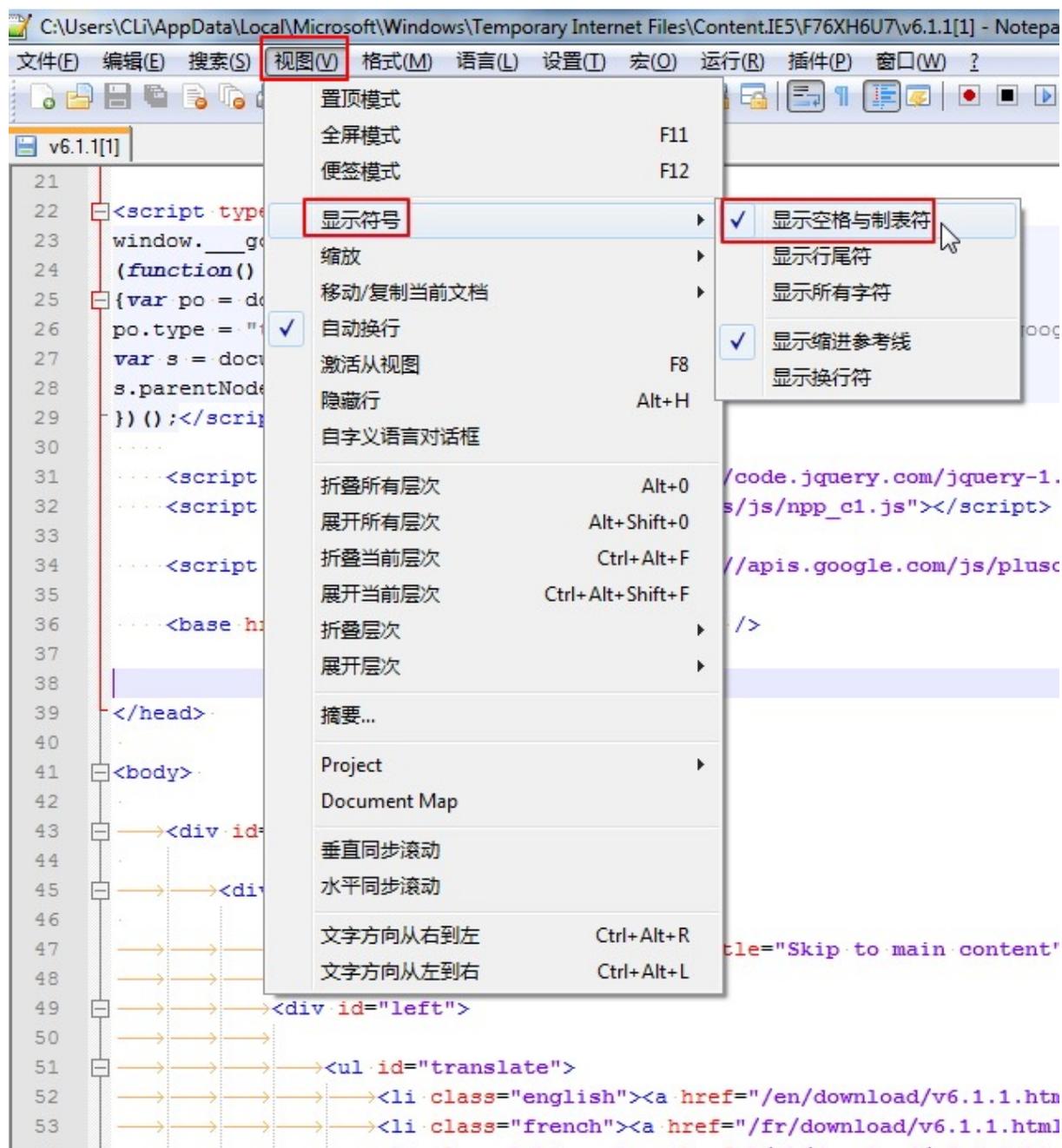
Notepad++默认，也和其他编辑器一样，是不显示空格和TAB键的

，不过我最近在用Notepad++写Python脚本过程中，常需要显示空格键和TAB键

以避免两者混合缩进，所以去设置了，将空格键和TAB键也显示出来。

下面以某HTML源码为例：

视图(V) ⇒ 显示符号 ⇒ 显示空格与制表符

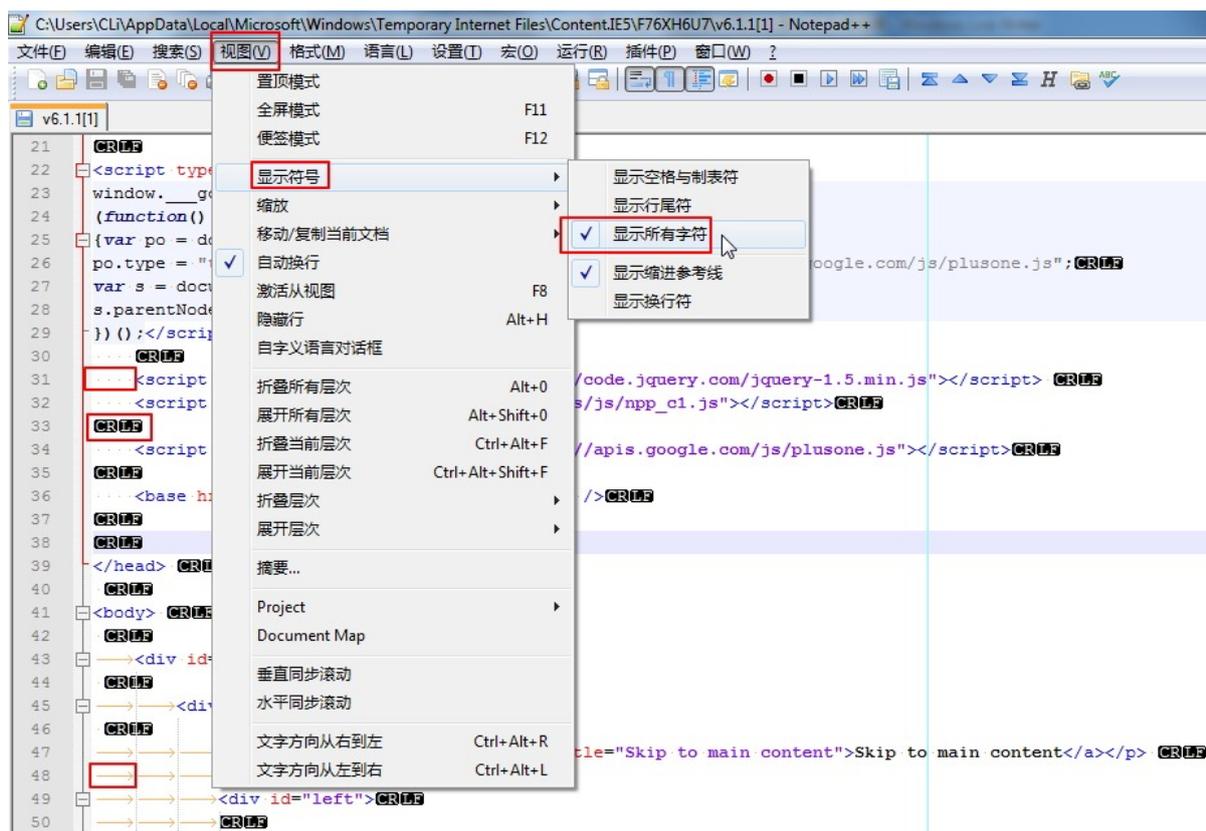


这样，就可以将空格键显示出来了，效果是四个橘黄色的点点，TAB键是右向的橘黄色箭头。

就很方便的，很清楚的看到，哪些是空格，哪些是TAB了。

另外，可以根据需要，比如想要知道那些行，有回车换行，还是单个的回车，单个的换行等，可以设置将所有的字符都显示出来：

视图(V) ⇒ 显示符号 ⇒ 显示所有字符



这样全部的回车换行符，CR和LF等，就都可以显示出来了。

例 3.24. Notepad++ 可以查看特殊字符的用途举例

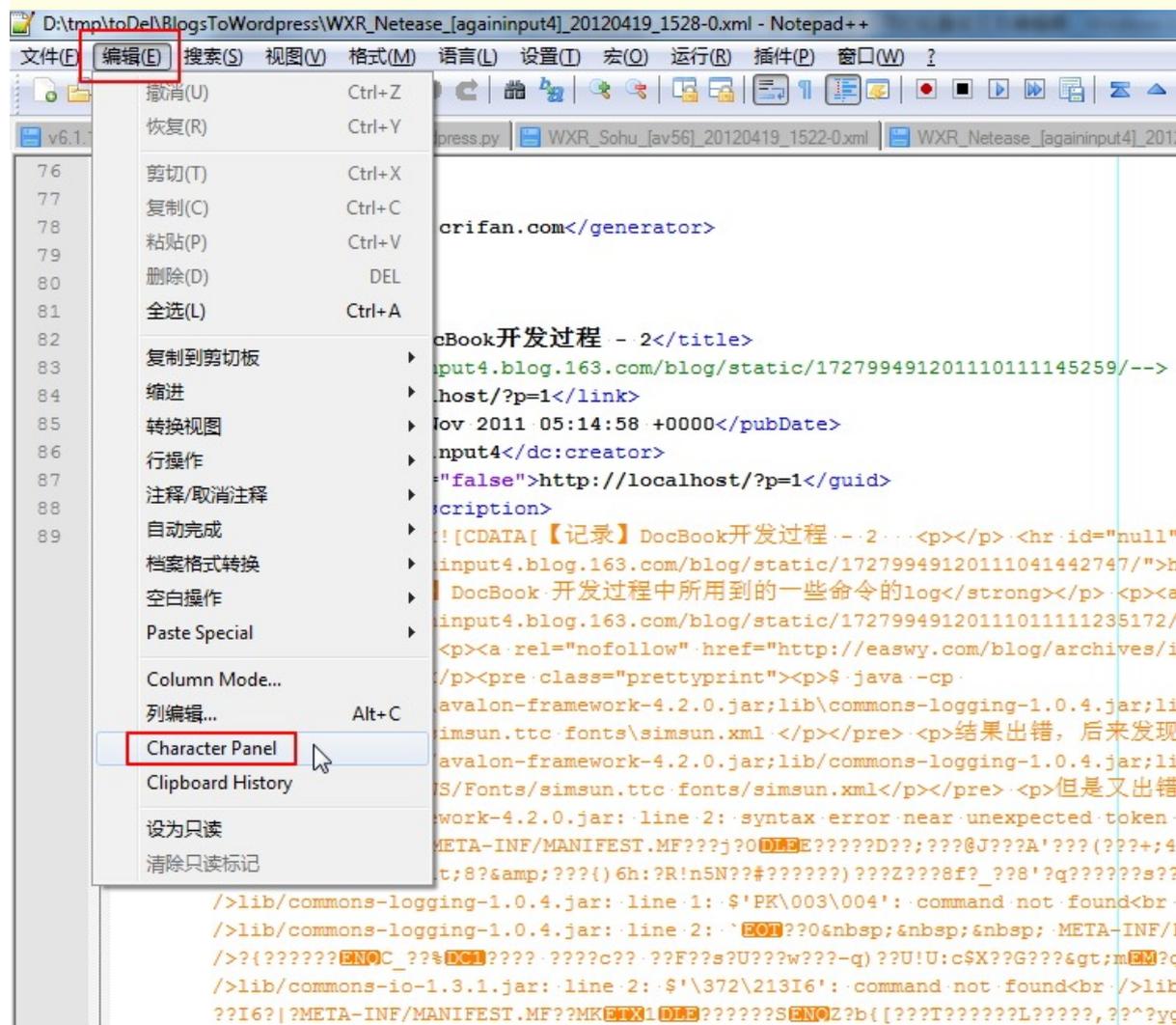
有人也许会问，就算Notepad++能显示所有特殊字符，好像也没啥用啊。

那此处就给你举个实际我遇到的例子：

之前遇到的一个问题是，用python脚本导出某163博客的帖子为WXR（XML文件）

但是结果导入WXR到wordpress失败，最后找到原因是，WXR文件中存在一些控制字符，该WXR文件效果如下：

编辑(E) ⇒ Character Panel



然后就调出了ASCII Insertion Panel，显示了ASCII字符表，包括了控制字符：



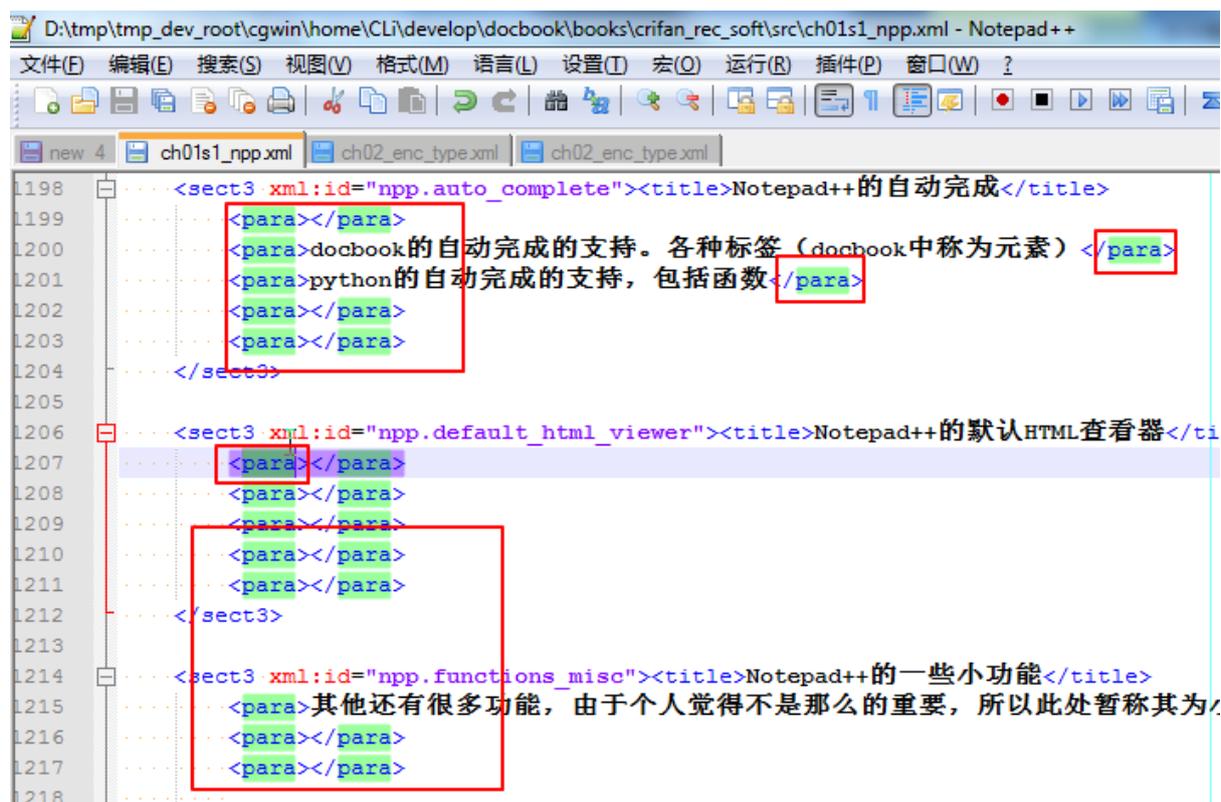
这样，双击ASCII Insertion Panel中任一（普通或控制）字符，比如DLE，DC1这两个控制字符，就可以插入对应的控制字符了。

而其他编辑器，好像没看到有这么好用的功能。

因此，不得不夸一句，Notepad++，真的是非常适合用来做文本编辑器，尤其是适合开发人员。

3.12. Notepad++ 高亮所选关键字

选中某单词即关键字后，notepad++会自动高亮对应选中的关键字：



方便查看和编辑文件。

此功能虽然简单，但是使用频度蛮高的，也是很实用的功能之一了。

3.13. Notepad++中Windows,Unix,Mac三种格式之间的转换

由于历史原因，导致Windows，Unix/Linux，Mac三者之间，对于文件中所用回车换行符，表示的方法，都不一样。

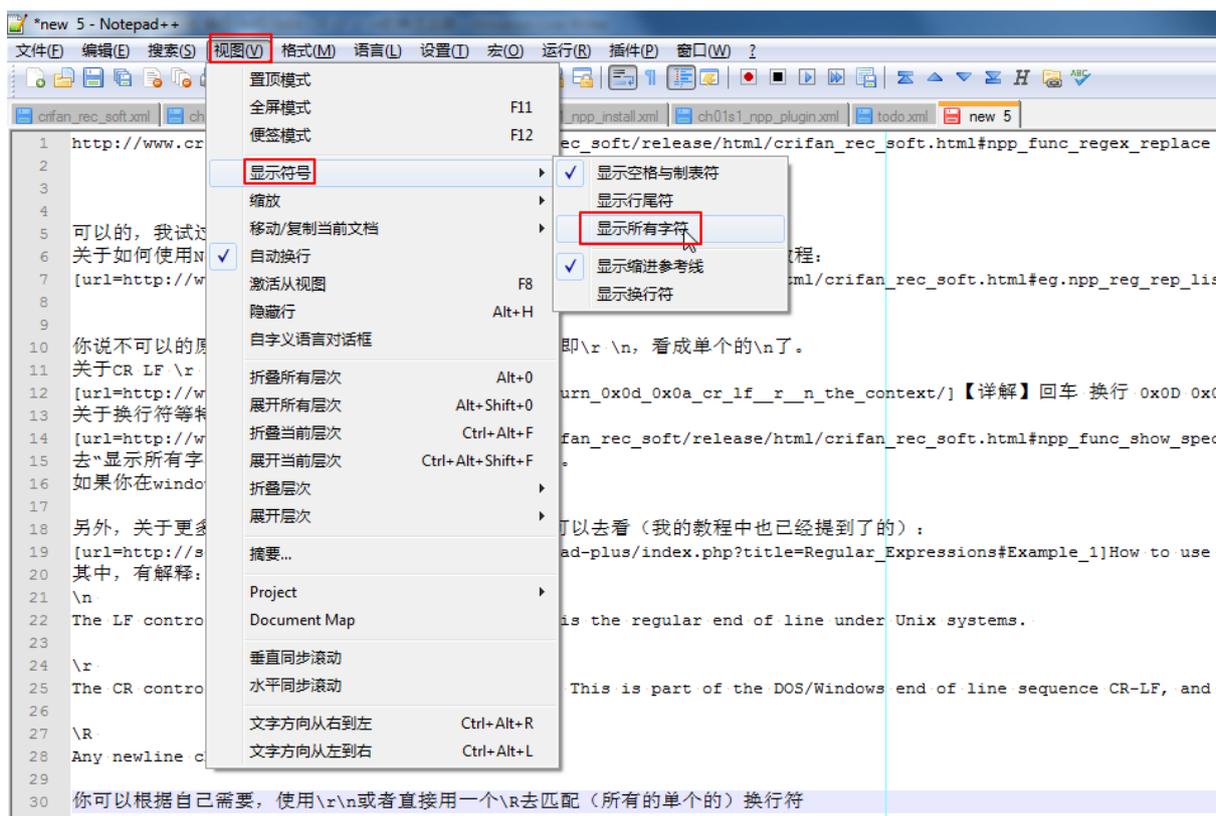
这就导致了很多人都会遇到回车换行符的困惑，和需要在不同格式间进行转换。

其中，关于回车换行符的详细解释，去看这里[\[13\]](#)

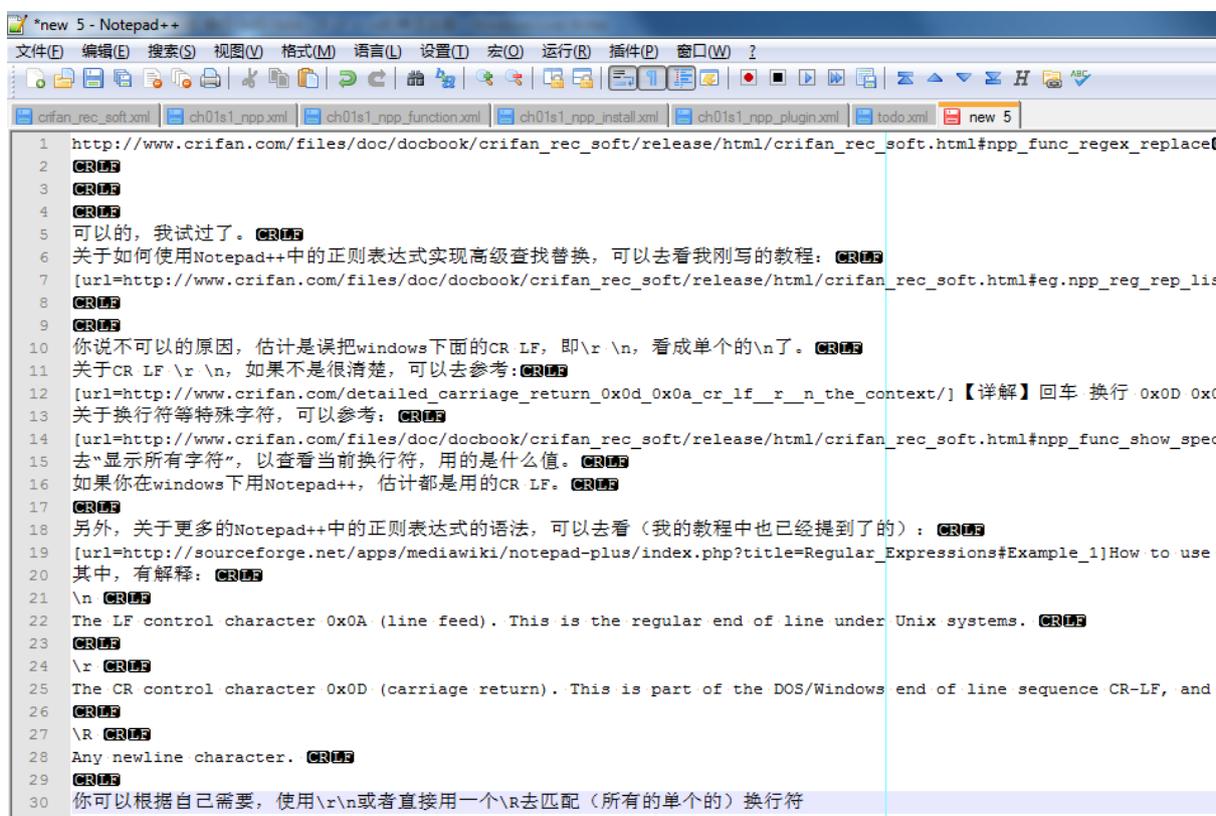
此处，介绍一下，如何通过Notepad++实现在这三者之间进行转换。

3.13.1. 查看当前文件格式（换行符用的是什么字符）

当前是Windows的平台，所以通过“显示所有字符”：

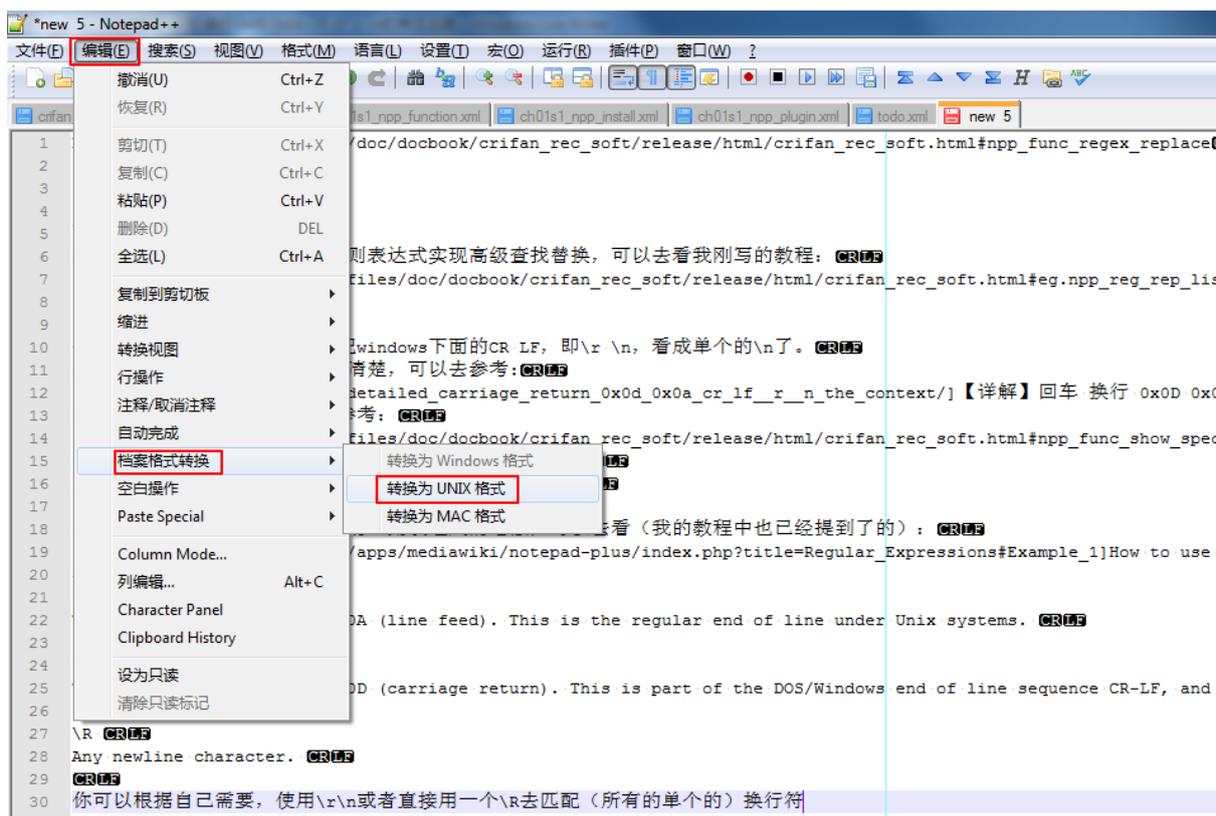


默认所查看到的是Windows的CR LF:

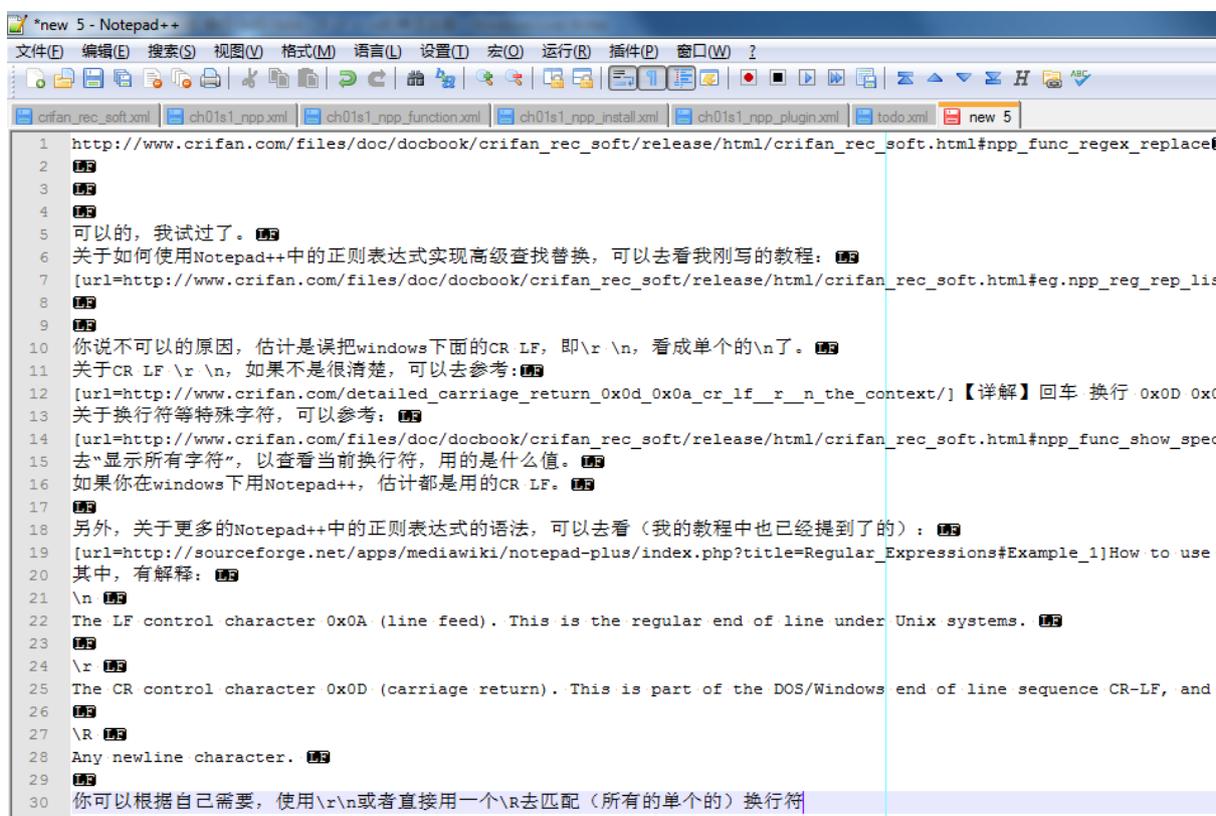


3.13.2. 从Windows转换为Unix/Linux格式

然后去从当前的Windows的格式，转换为Unix格式：

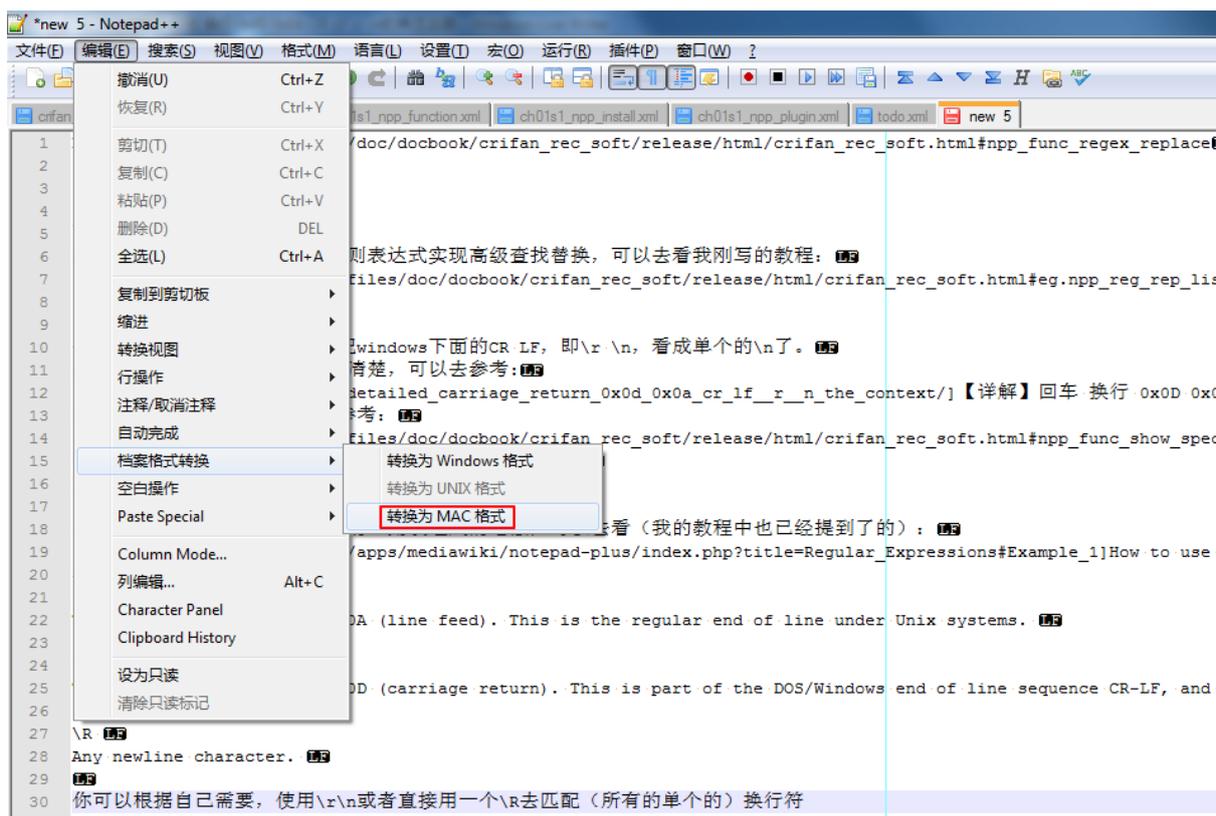


然后Windows的CR LF就都变成Unix/Linux的LF了:

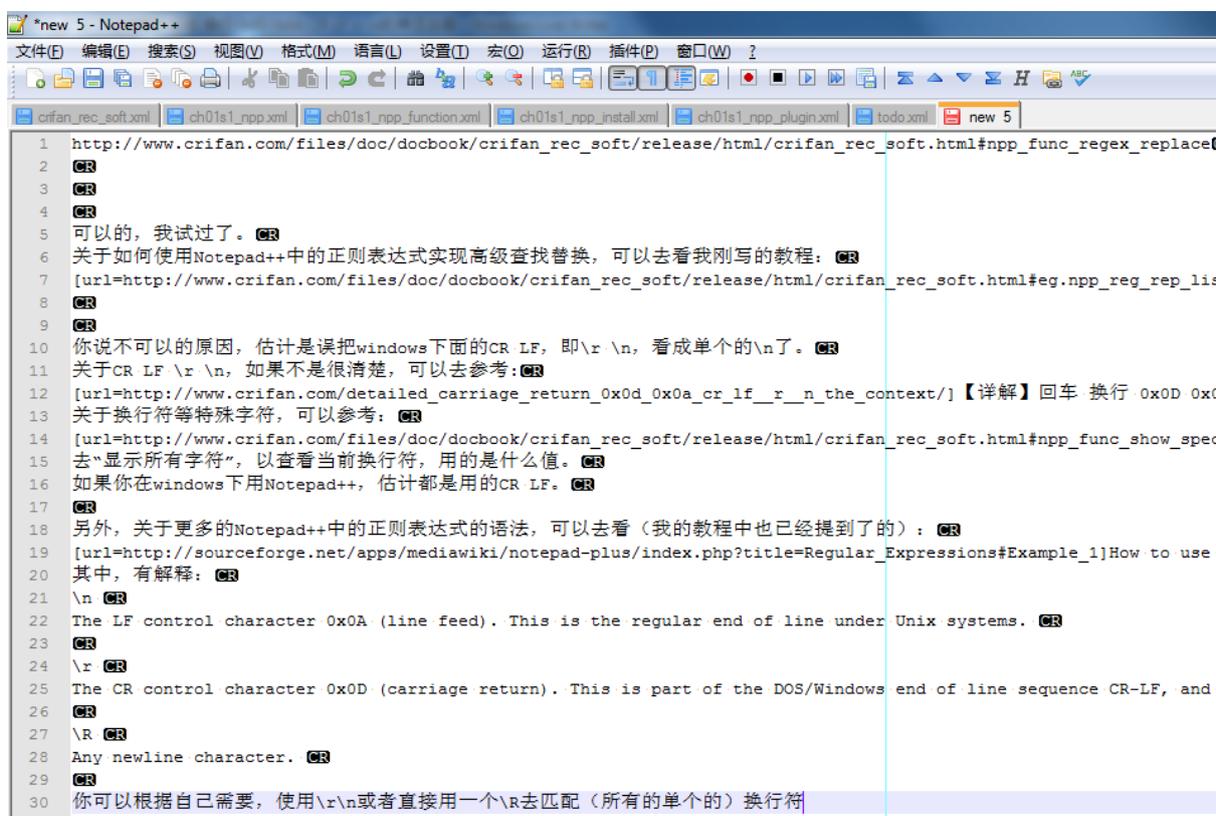


3.13.3. 从Unix/Linux转换为Mac格式

然后再去从当前的Unix/Linux格式，转换为Mac格式：

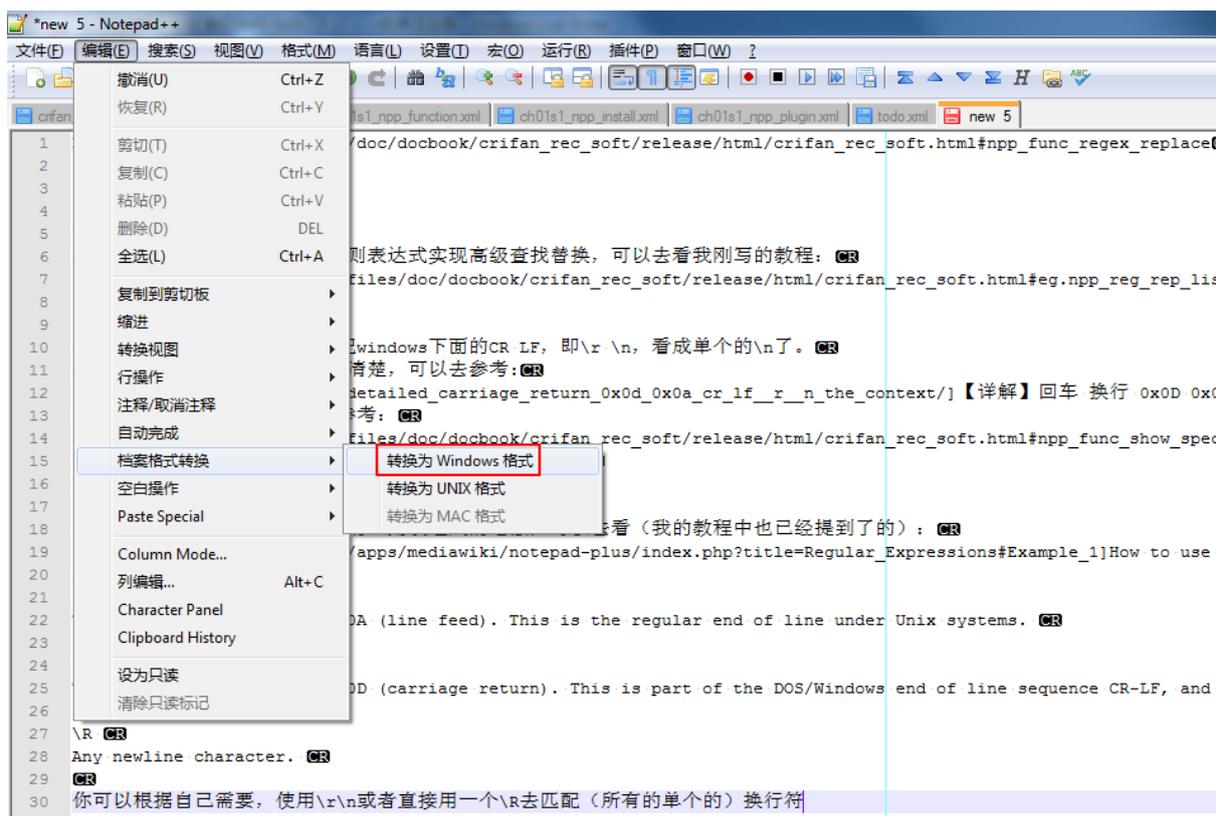


然后Unix/Linux的LF就都变成了Mac的CR了：

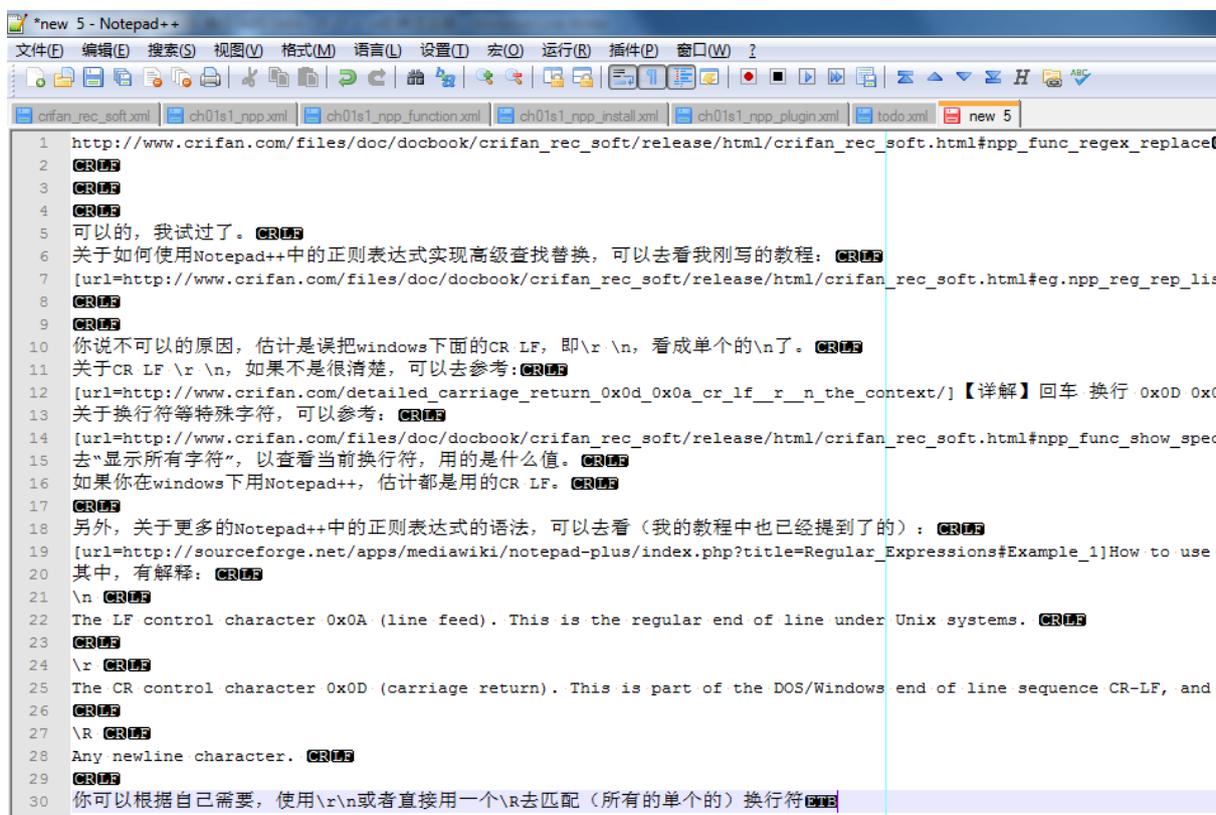


3.13.4. 从Mac格式转换为Windows格式

然后再去从当前的Mac格式，转换为Windows格式：

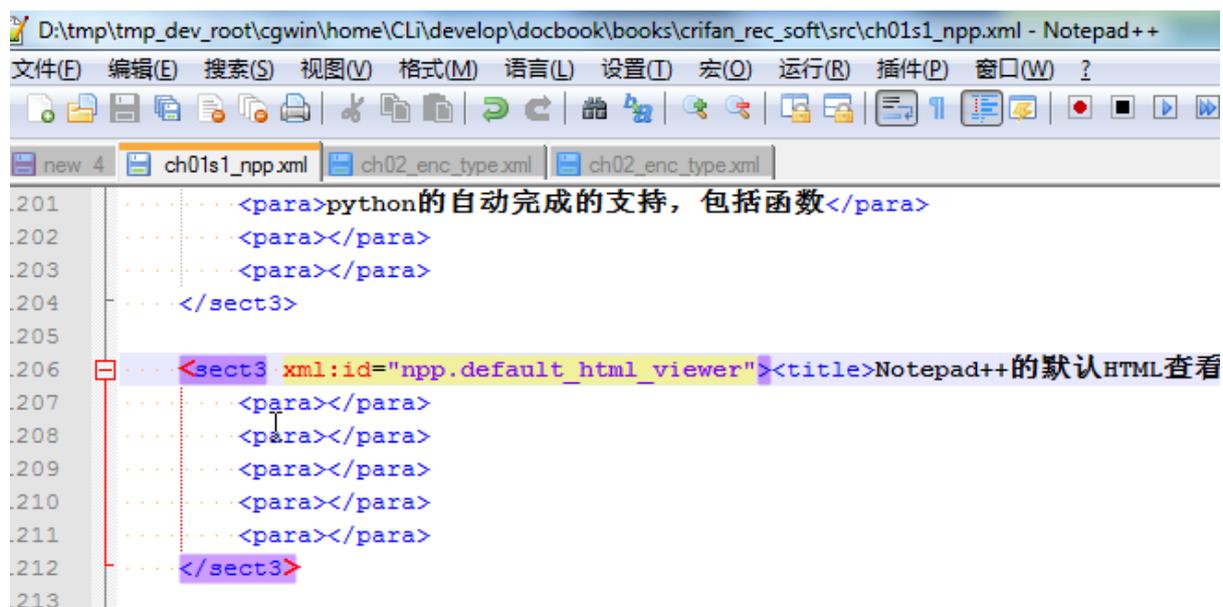


然后Mac的CR就都变成Windows的CR LF了：



3.14. Notepad++ 高亮匹配的键对

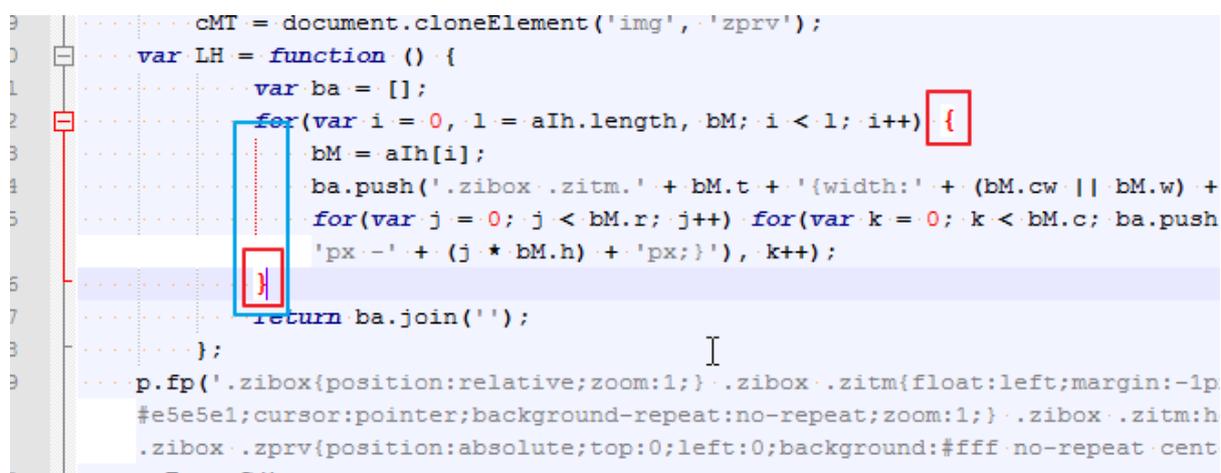
点击某个关键词后，会自动高亮显示其所匹配的键对：



方便查看和编辑文件。

此功能很适合对于编辑xml, html等类型的文件, 这样容易看清, 哪个标签, 关键字, 所对应的关系。

另外, 也适用于查看代码中的, 当前的代中括号, 方括号, 大括号所对应的那个括号在哪里:



3.15. Notepad++ 文件自动更新

之前不了解, 关于文件自动更新, 原来也是可以根据需要设置不同更新模式的。

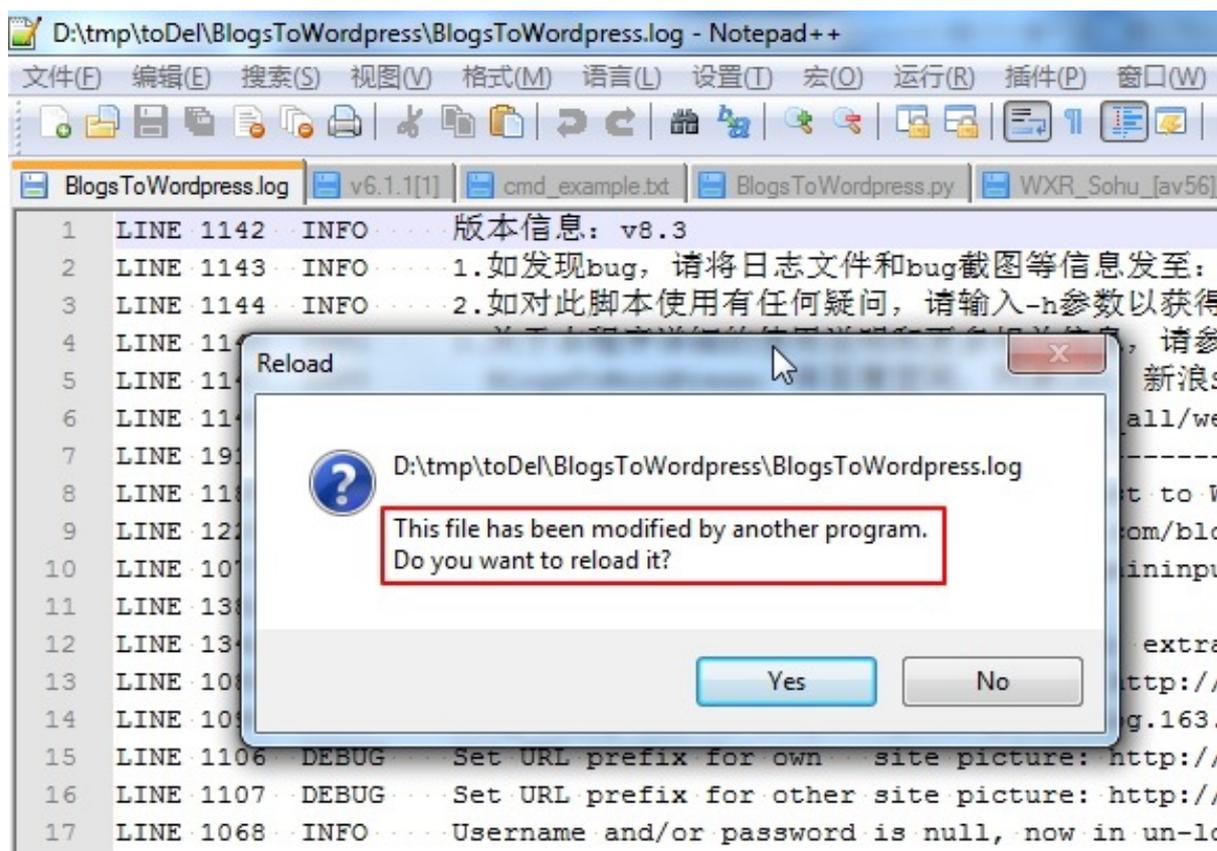
说个最实际的例子:

之前用写python脚本, 在windows的cmd中输入脚本去执行, 执行过程中, 会生成一个log文件: xxx.log,

另外每次执行不同的命令, 对应的log文件内容也会变化,

而该log文件以及对应的python脚本xxx.py文件, 我都是用Notepad++打开的,

所以, 执行完cmd中的python脚本后, 想要回来再次去编辑xxx.py文件, 就会先遇到, 关于xxx.log被更新的提示:



然后点击个yes,才能去编辑我想要编辑的xxx.py文件。

所以,每次都被这个提示框所打扰,觉得挺烦的。

后来,无意间,发现原来是可以取消此提示的:

设置(T) ⇒ 首选项... ⇒ 其他 ⇒ 自动检测文件状态,勾选"启用", "自动更新文件"



这样，每次xxx.log文件再被修改，就会自动更新内容，而没有了之前的跳出的那个对话框的提示了。

类似的，也可以通过勾选或取消“启用”，实现开启或关闭自动检测文件状态这个功能。

或者每次更新完文件，想要光标自动跳到文件末尾，那么就去选上“更新后定位到末位”。

3.16. Notepad++设置用空格取代TAB键

设置(T) ⇒ 首选项... ⇒ 语言 ⇒ 标签设置，勾选“以空格取代”



这样，以后输入Tab键的时候就会自动以所设置的4个空格代替。

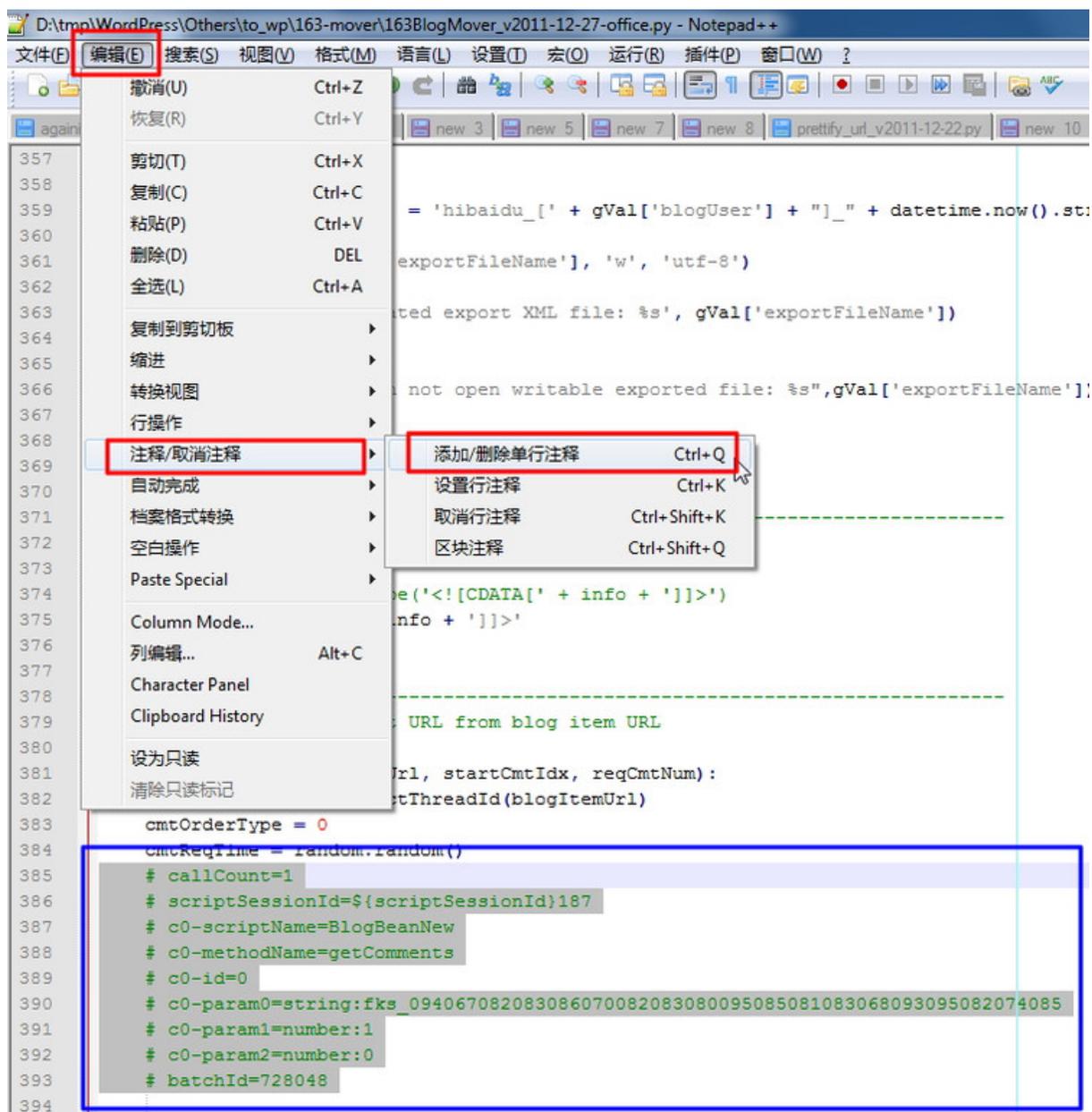
此功能的用途：

1. 部分程序的需要
主要是有些情况下，比如写Python脚本的时候用到，以避免TAB键和空格键的混合缩进，其会导致Python 3.0等报错，而让输入的TAB键，自动变成空格键，则自动规避了此问题。
2. 兼容不同平台
另外，此功能，也是出于兼容不同平台，TAB键的宽度不同的考虑

这样就可以避免不同平台下，不会因为TAB键是4个空格，还是2个空格等，而导致代码（文字）的对齐不匹配的现象了。

3.17. Notepad++给代码单行/批量，添加/取消注释

编辑(E) ⇒ 注释/取消注释 ⇒ 添加删除/单行注释 Ctrl+Q



就可以实现对选中的多行Python代码，每行都添加对应的符号"#", 实现对应的注释代码的功能了。

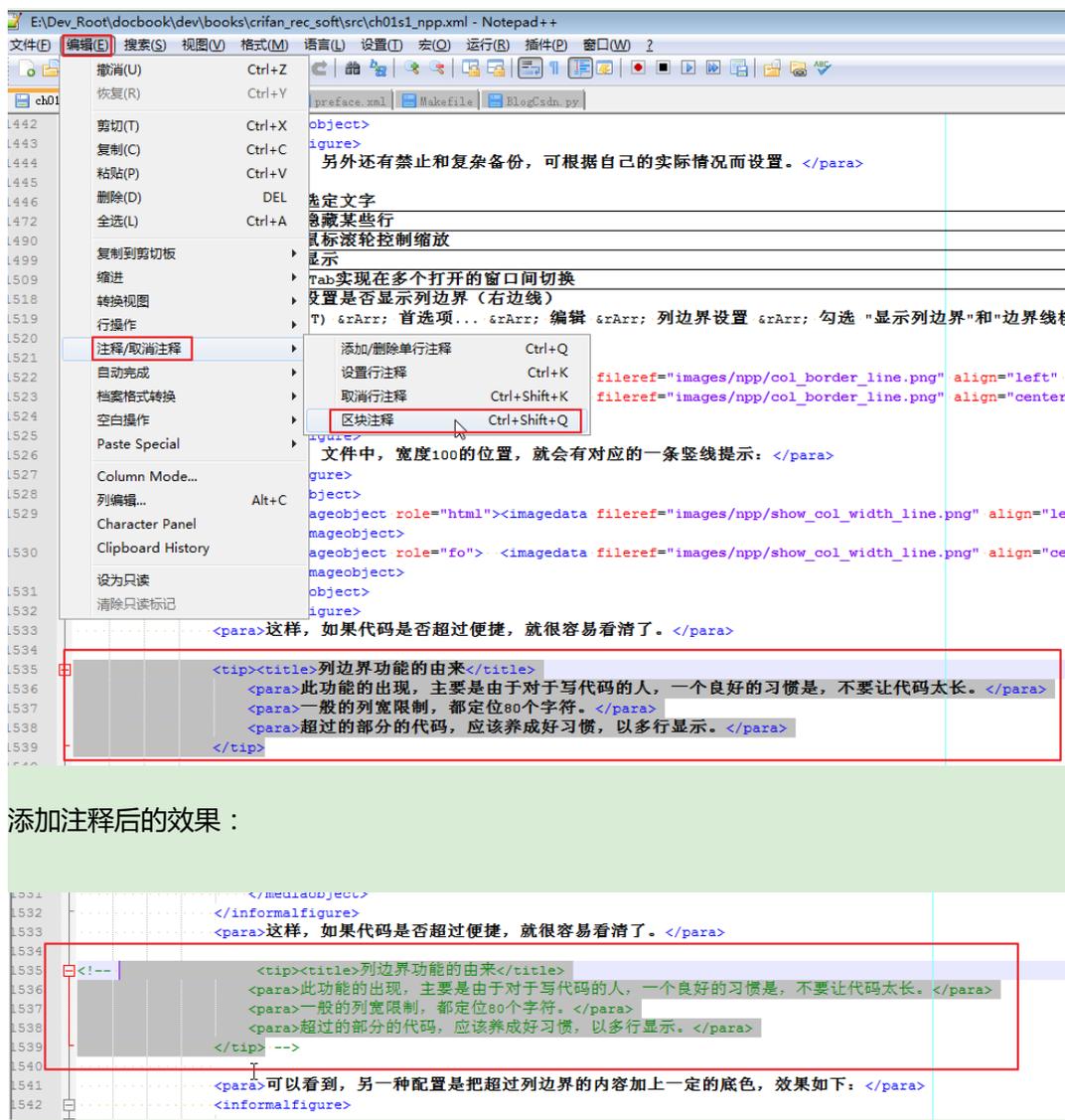


提示

此功能，不仅方便一次性可以注释多行代码。更加强大的是，会根据不同的语言的代码，添加对应的注释前缀。

比如，Python语言，以井号"#"为前缀，表示注释，而c等语言，是以斜杠"/"为前缀，表示注释，而xml语言中，是以<!-- xxx -->表示对xxx进行注释，其都可以根据所识别出来的源码的类型，去添加相应的注释。

下面就是通过区块注释，实现对xml代码实现给一段代码，添加区块注释功能的：

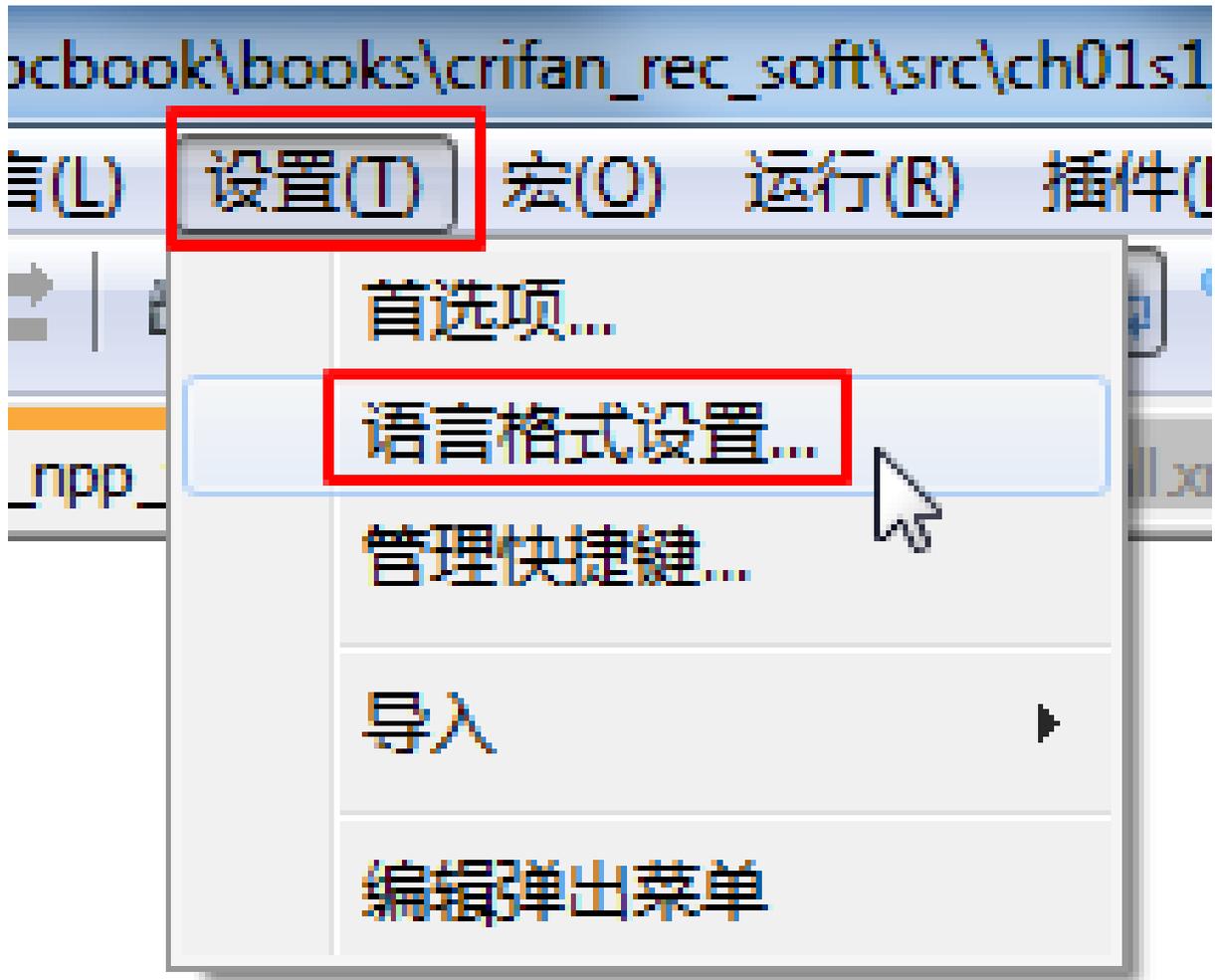


添加注释后的效果：

3.18. Notepad++的多主题

在前面的安装过程中：[Notepad++安装选项：Themes](#)，已经介绍了其会安装一些主题。

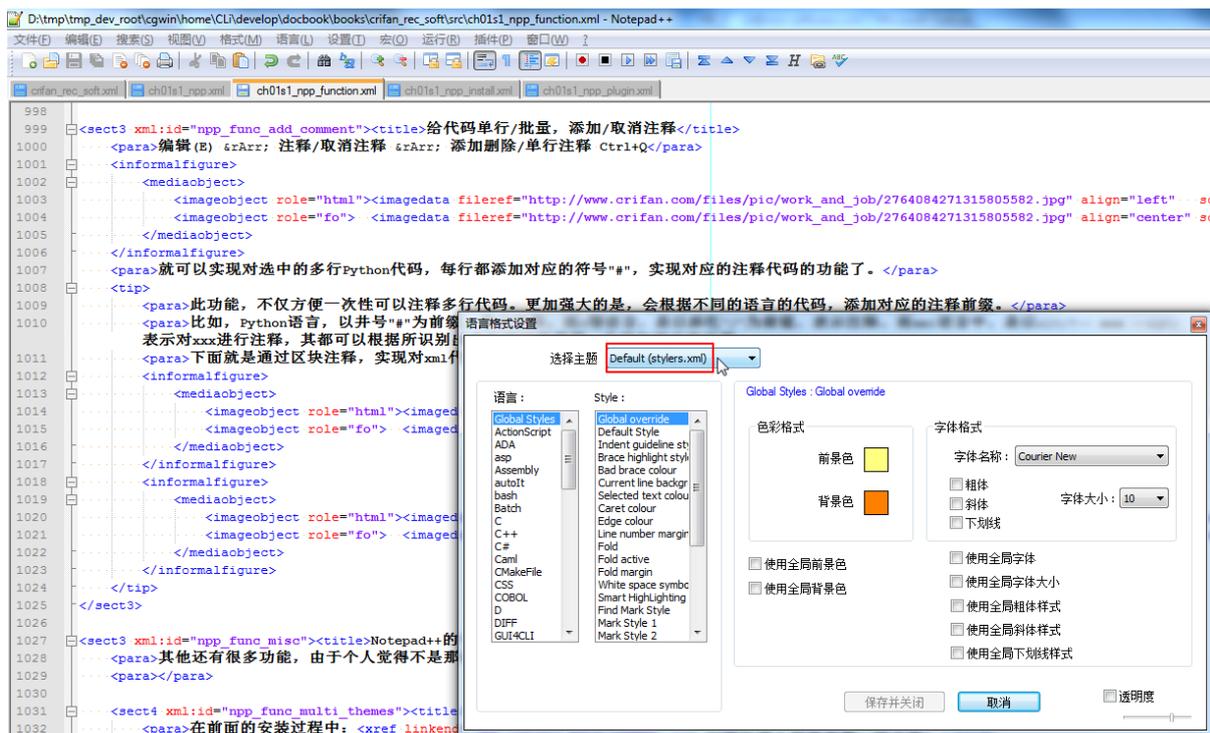
此处，可以去：设置(T) ⇒ 语言格式设置...



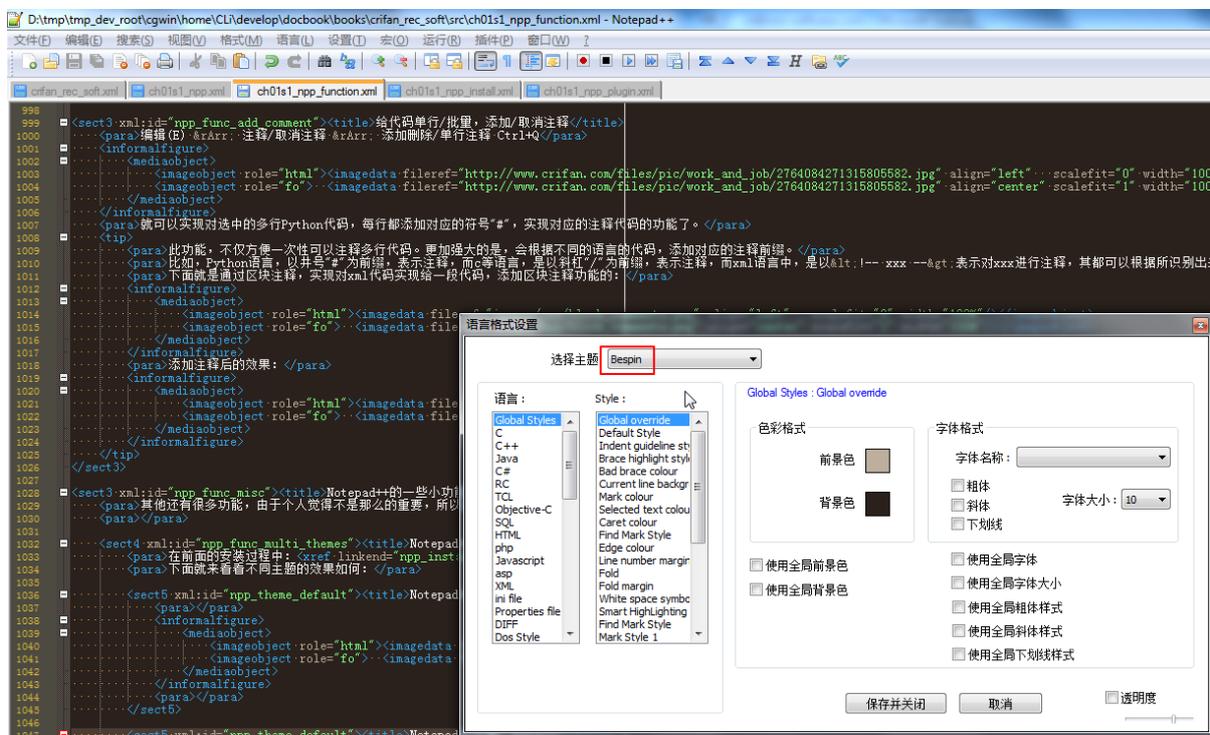
中，在“选择主题”部分，去设置不同的主题。

下面就来看看不同主题的效果如何：

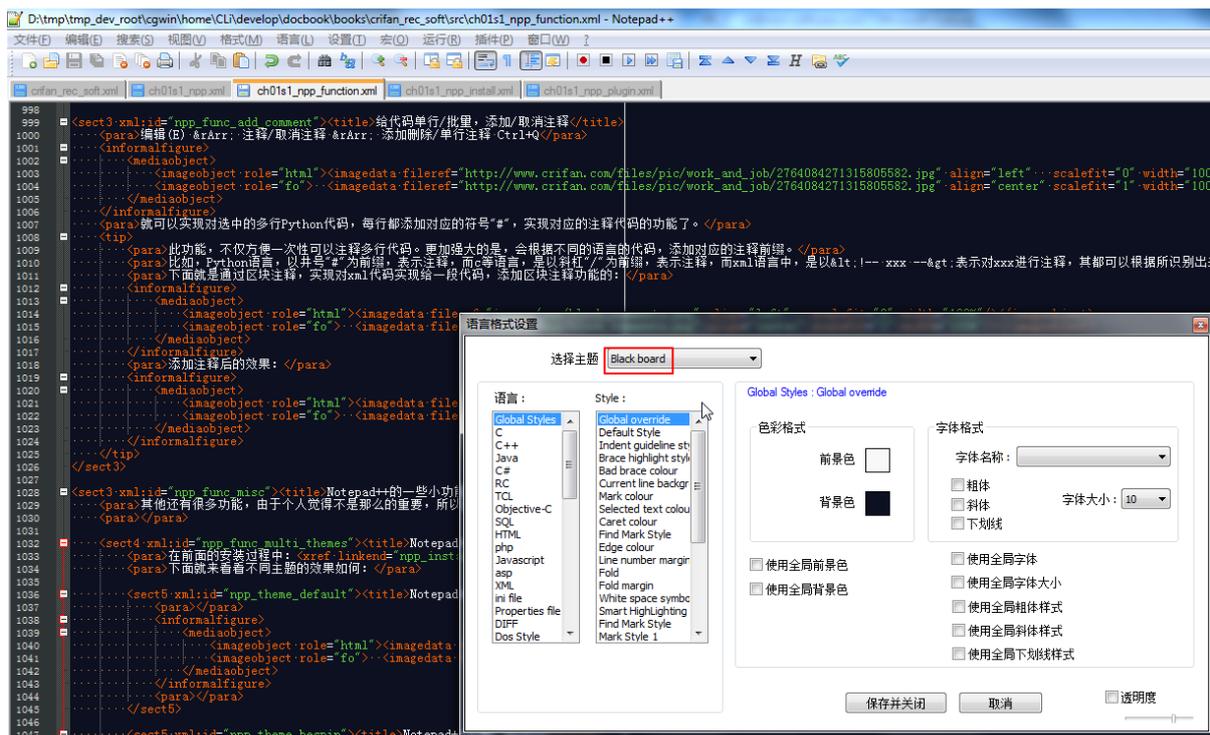
3.18.1. Notepad++主题: Default



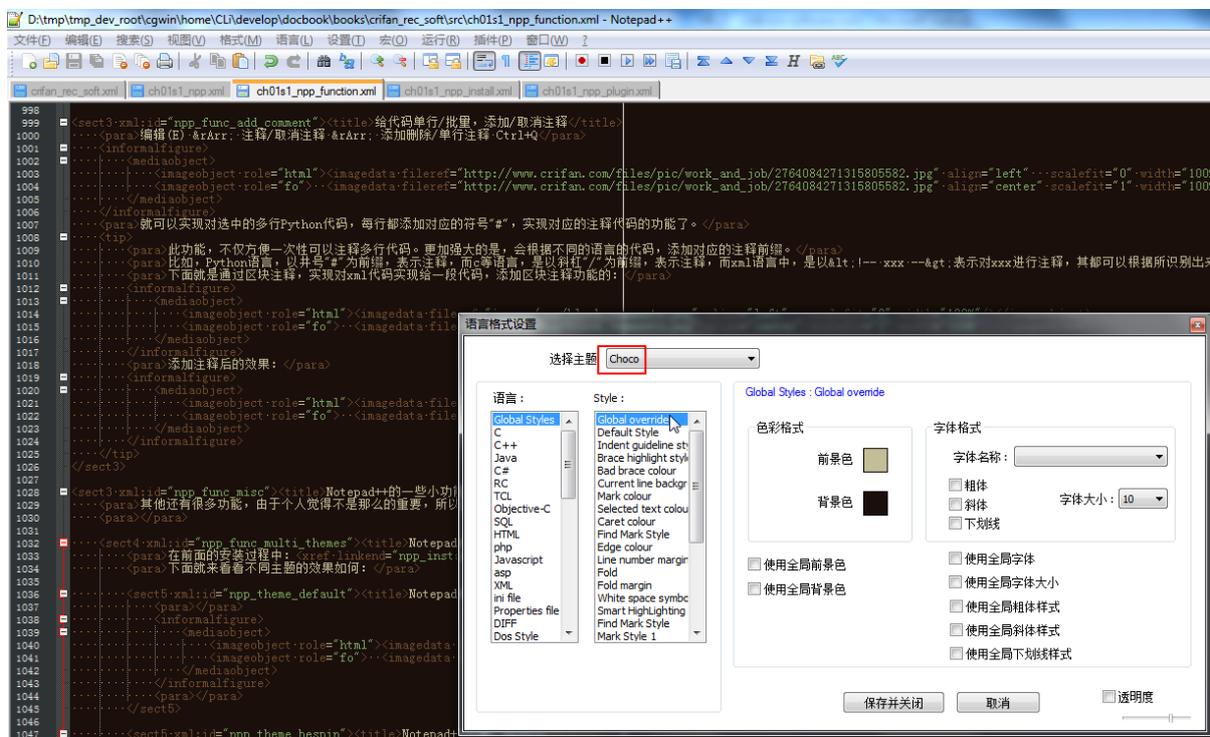
3.18.2. Notepad++主题: Bespin



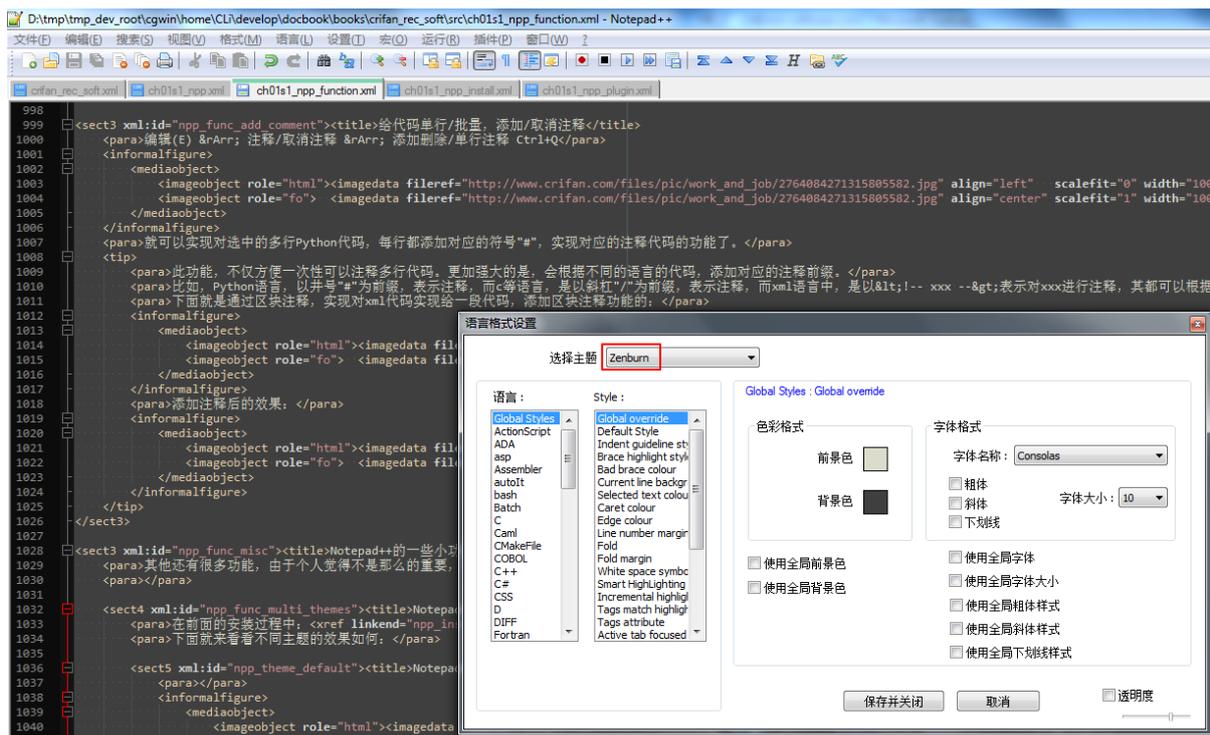
3.18.3. Notepad++主题: Black Board



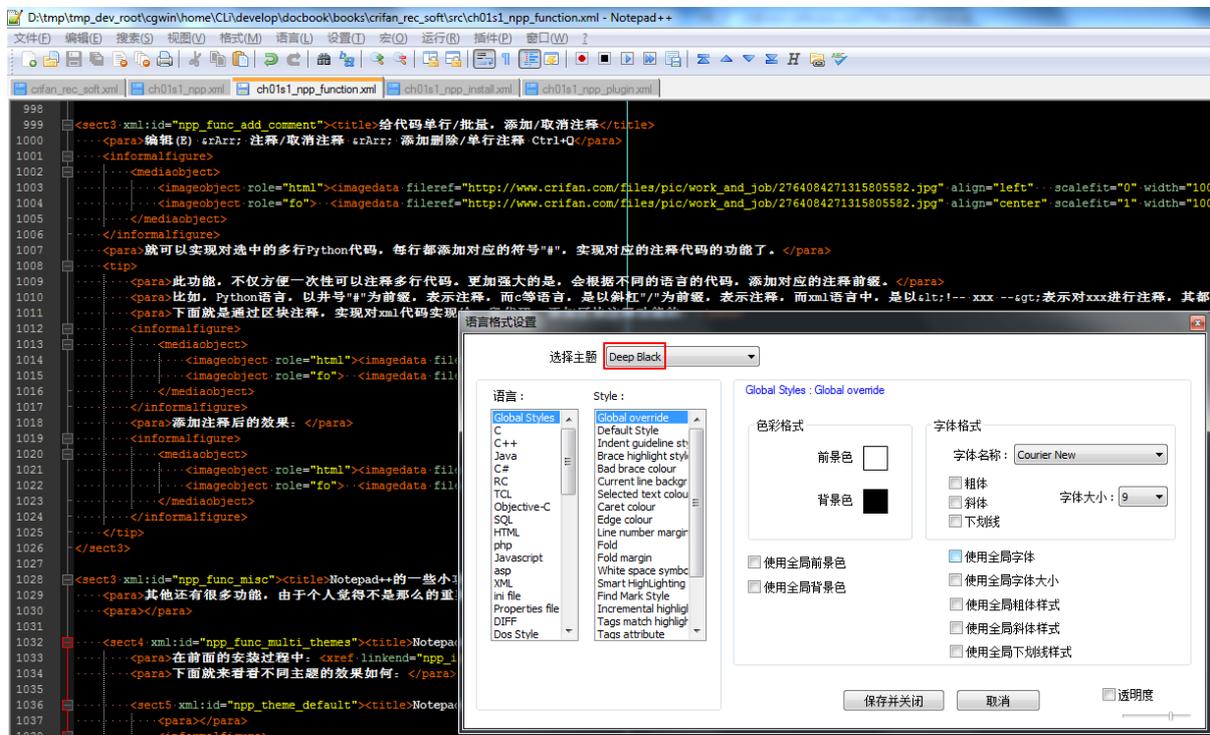
3.18.4. Notepad++ 主题: Choco



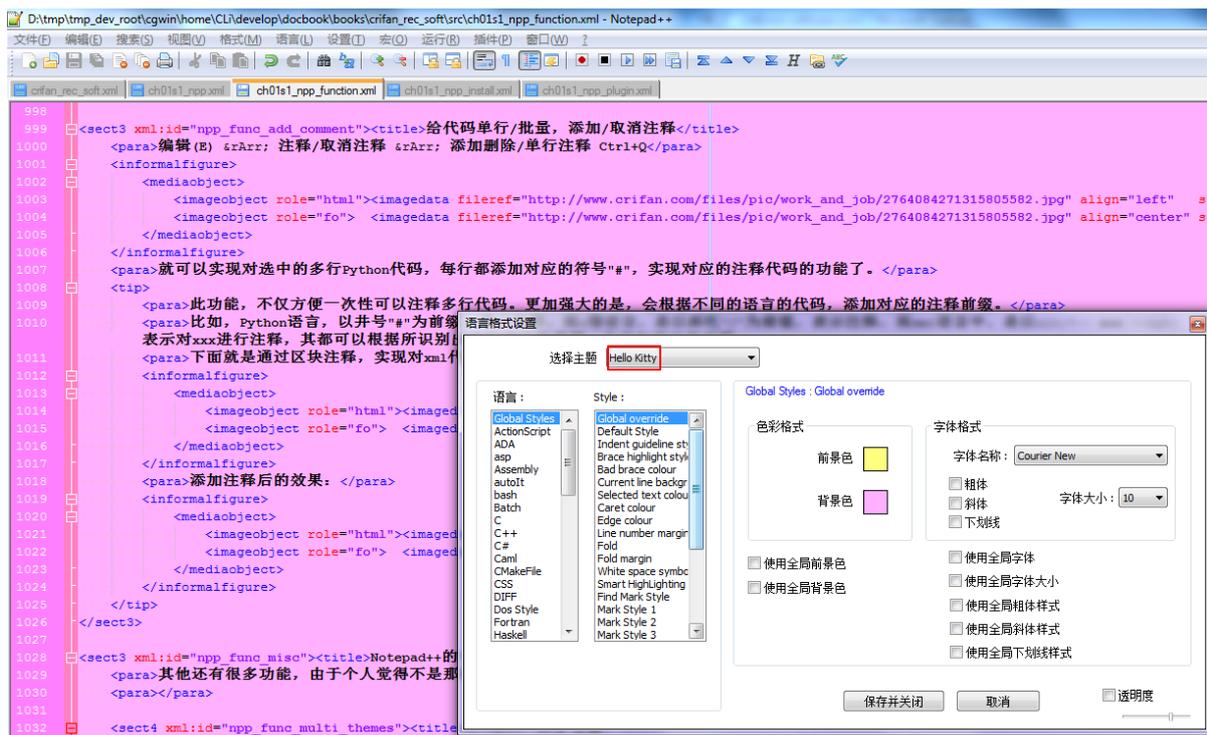
3.18.5. Notepad++ 主题: Zenburn



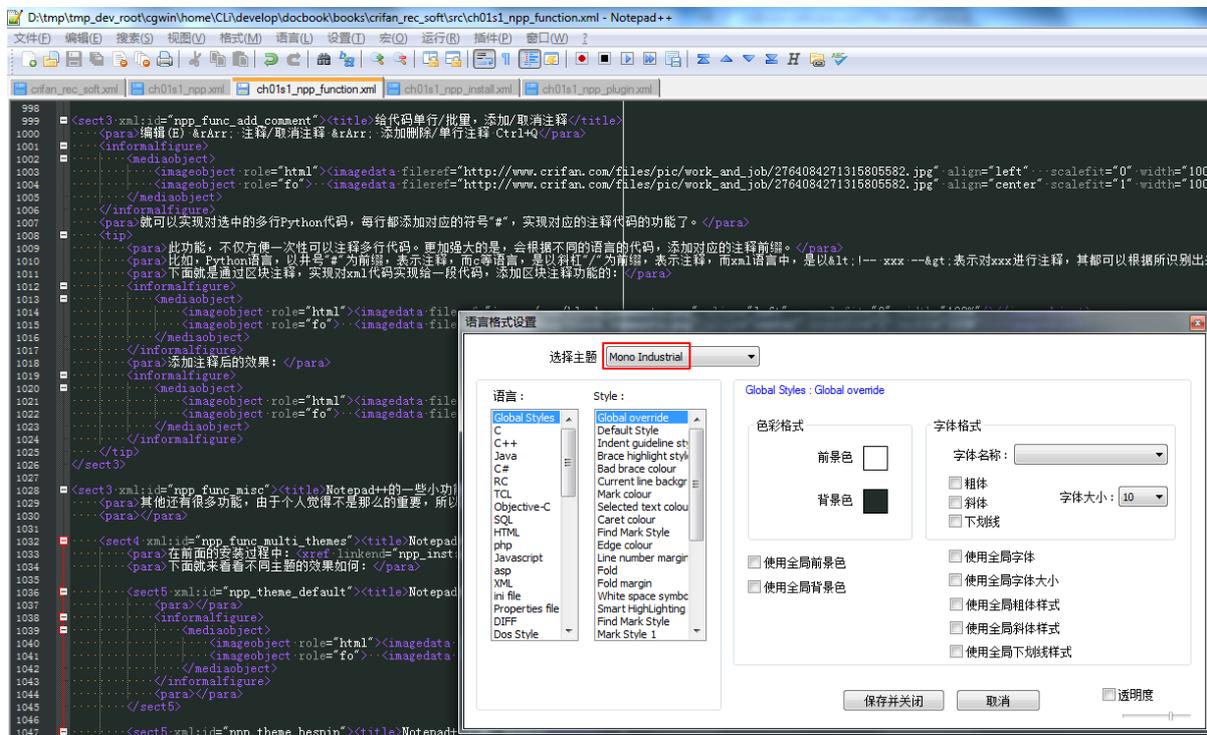
3.18.6. Notepad++主题: Deep Black



3.18.7. Notepad++主题: Hello Kitty

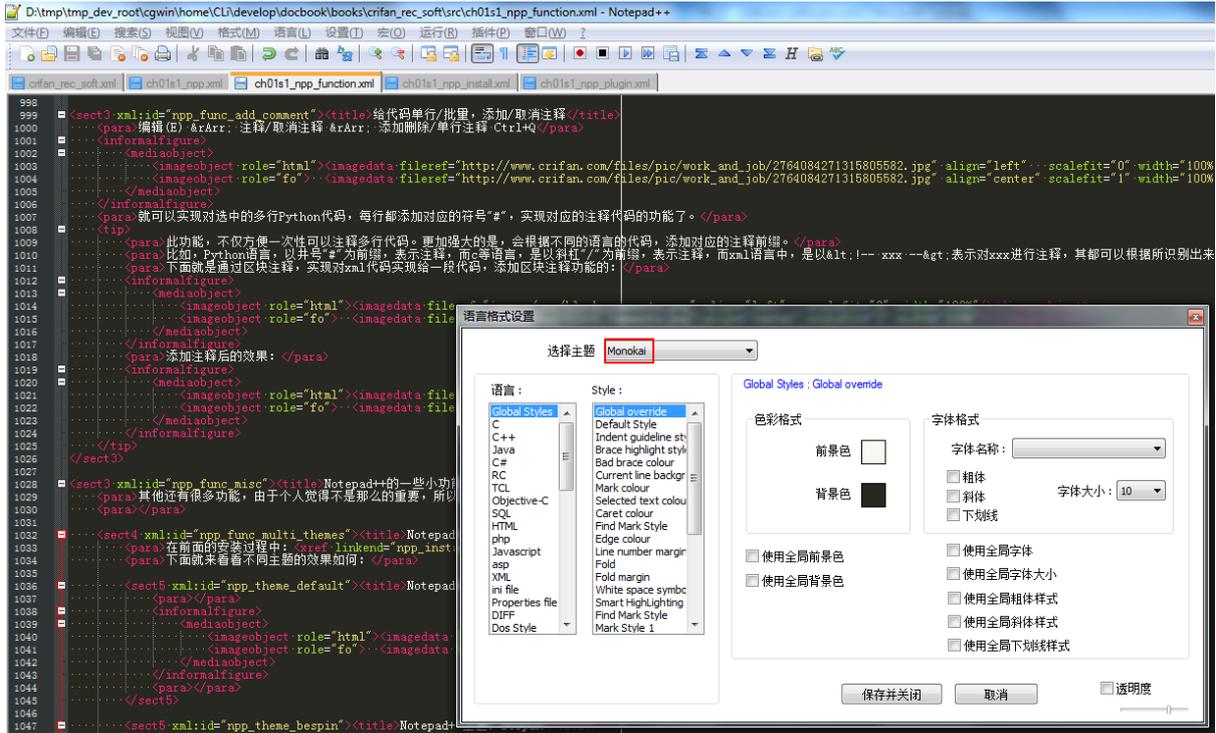


3.18.8. Notepad++主题: Mono Industrial

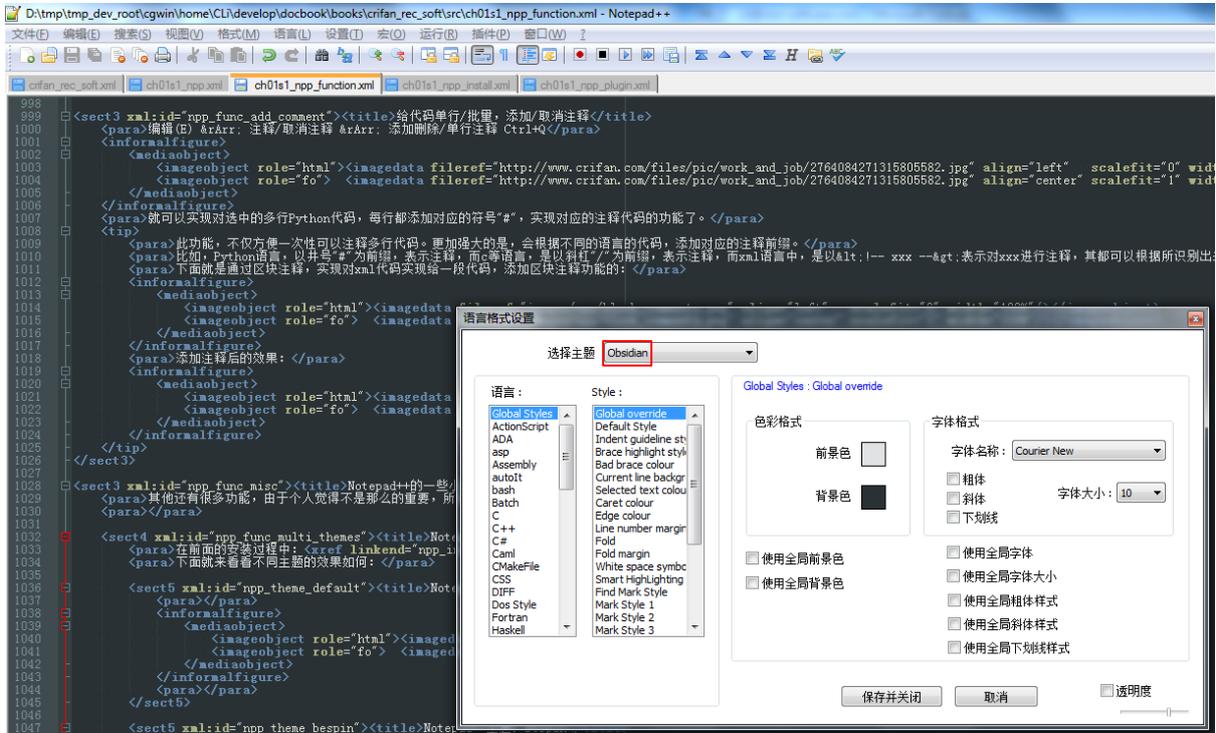


3.18.9. Notepad++主题: Monokai

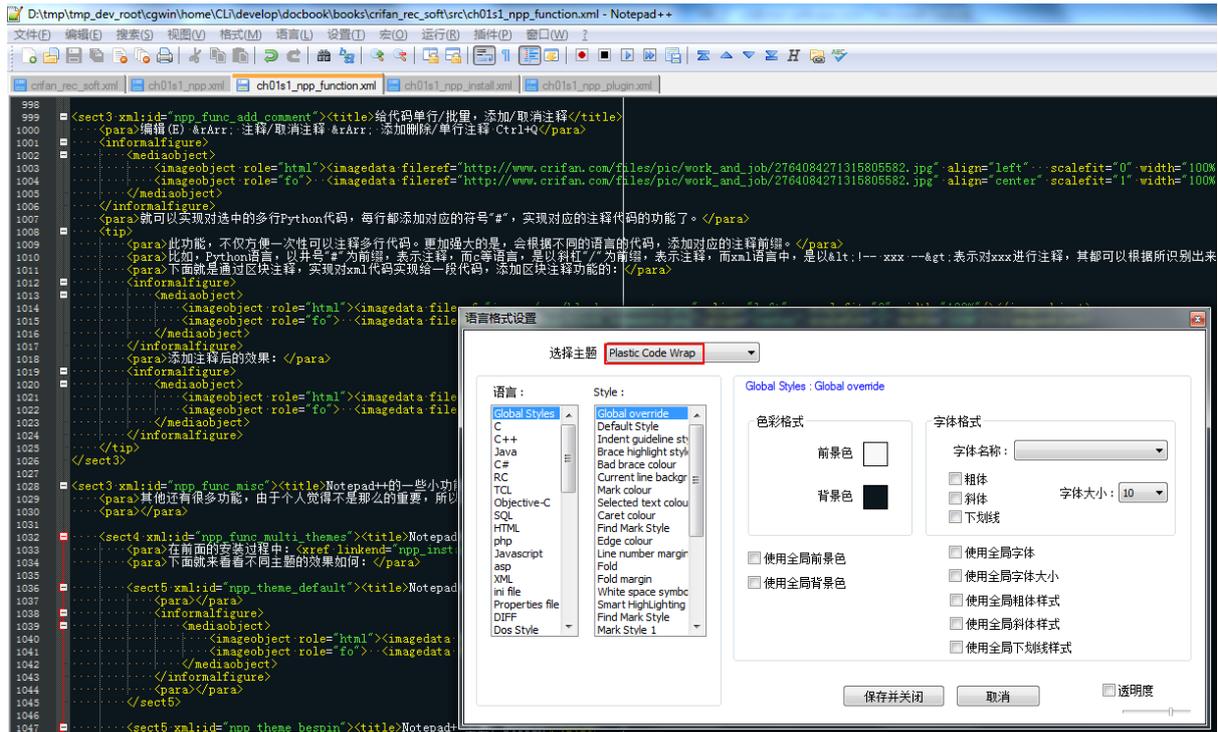
Notepad++ 的功能详解



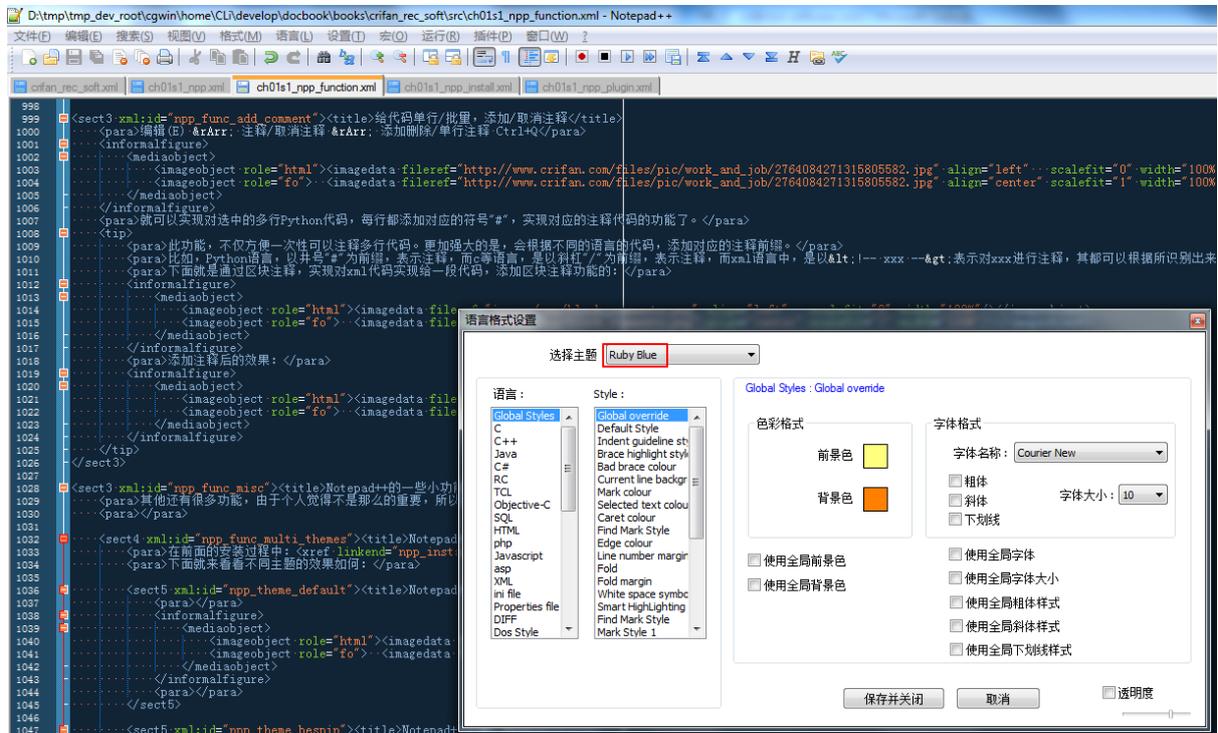
3.18.10. Notepad++ 主题: Obsidian



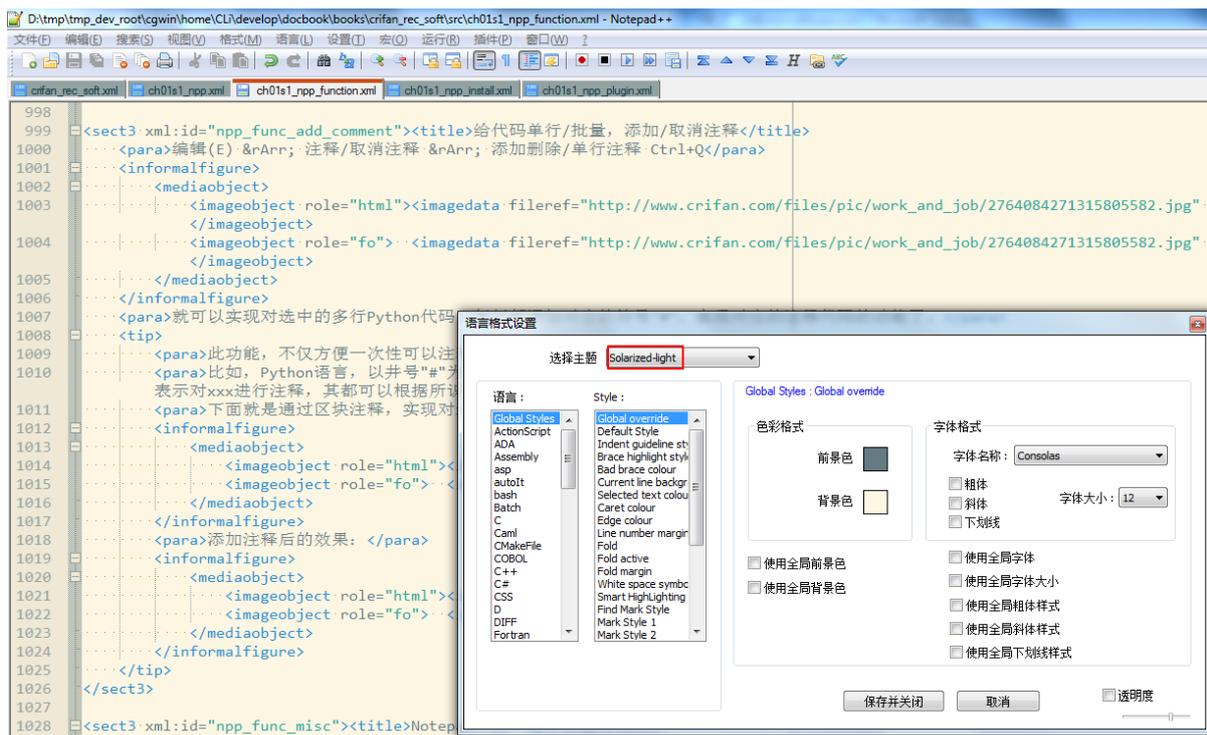
3.18.11. Notepad++ 主题: Plastic Code Wrap



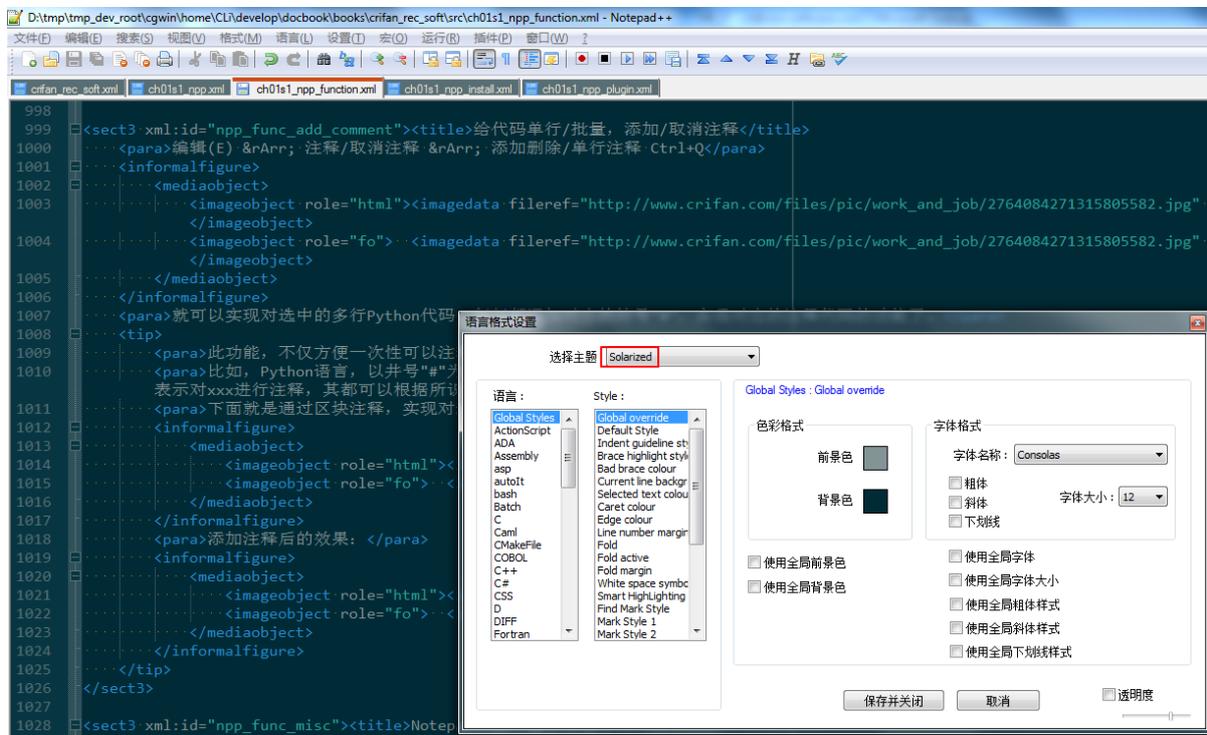
3.18.12. Notepad++ 主题: Ruby Blue



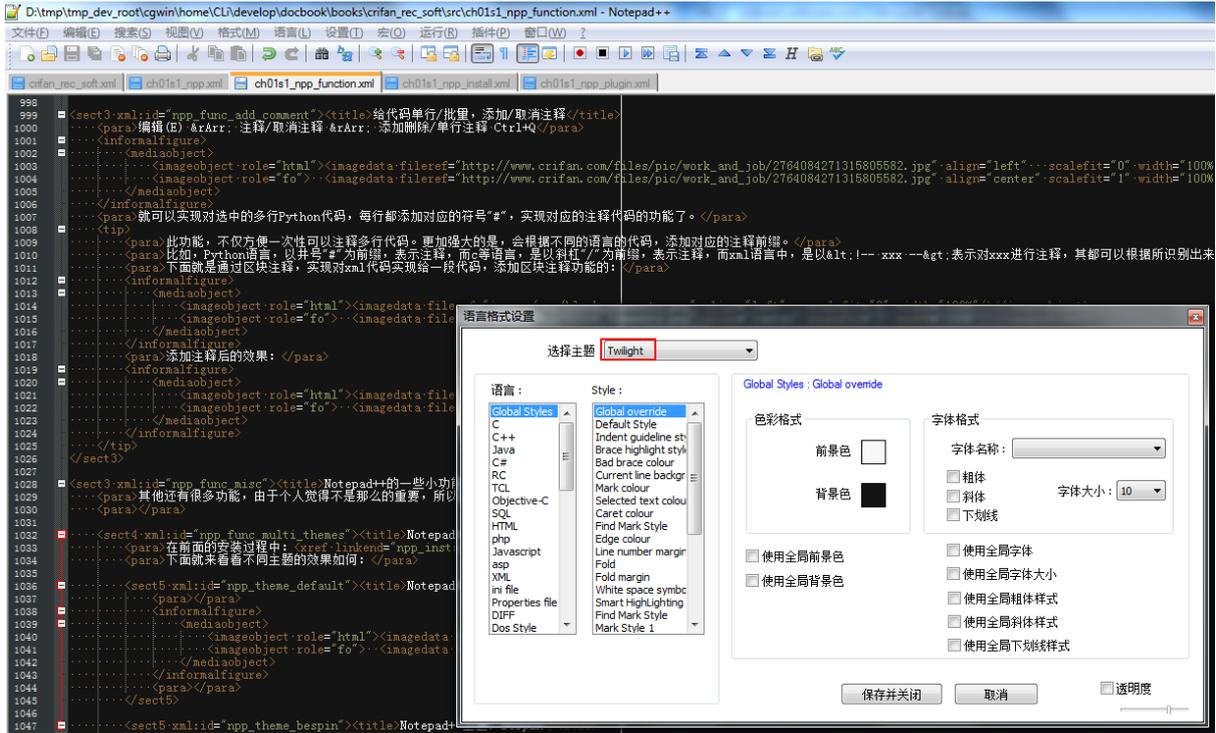
3.18.13. Notepad++ 主题: Solarized-Light



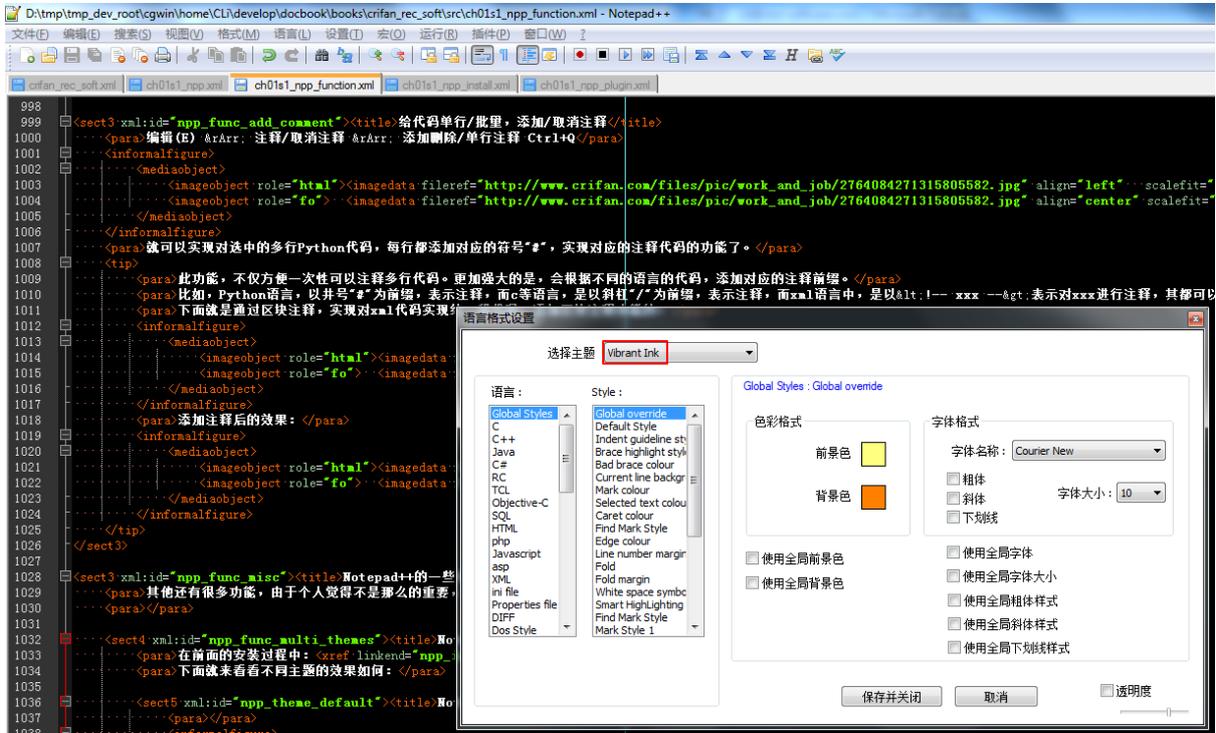
3.18.14. Notepad++ 主题: Solarized



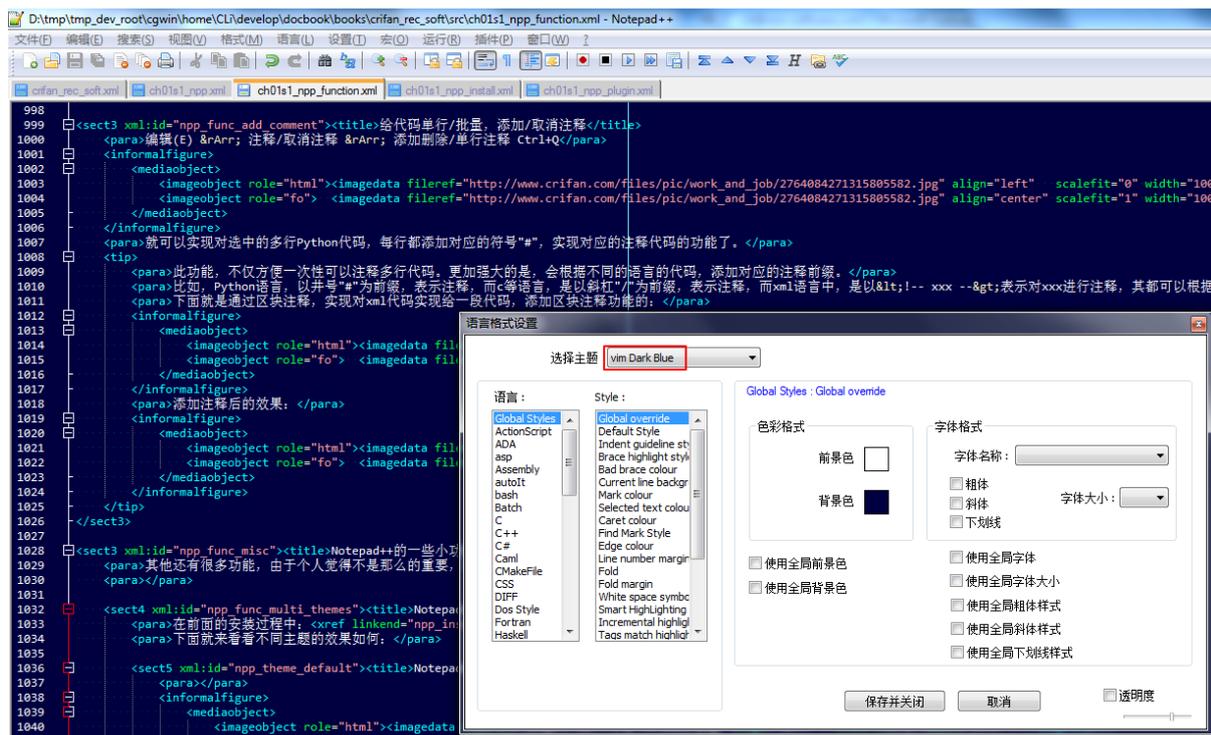
3.18.15. Notepad++ 主题: Twilight



3.18.16. Notepad++ 主题: Vibrant Ink



3.18.17. Notepad++ 主题: Vim Dark Blue



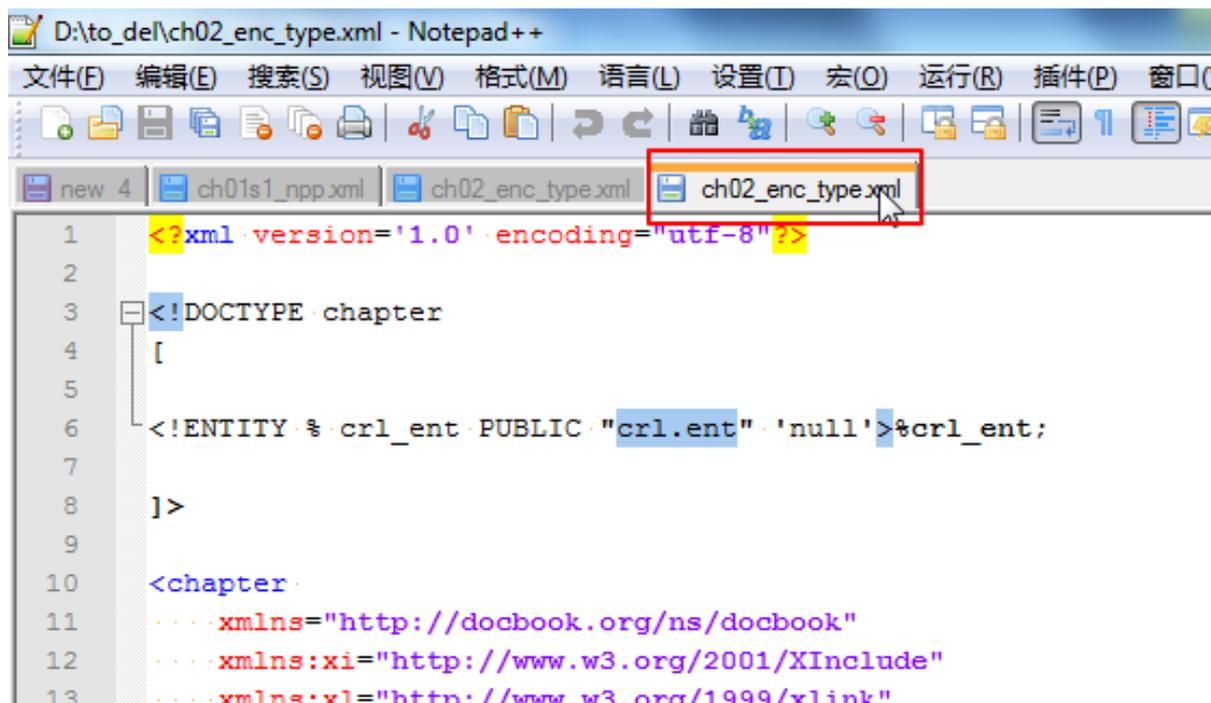
3.19. Notepad++的一些小功能

其他还有很多功能，由于个人觉得不是那么的重要，所以此处暂称其为小功能。

虽然此处称为小功能，但是真正等你需要用到时候，也会发现，真的很贴心，很好用。

3.19.1. 支持双击Tab标签关闭文件

想要快速关闭某个文件，可以通过双击该文件的Tab：

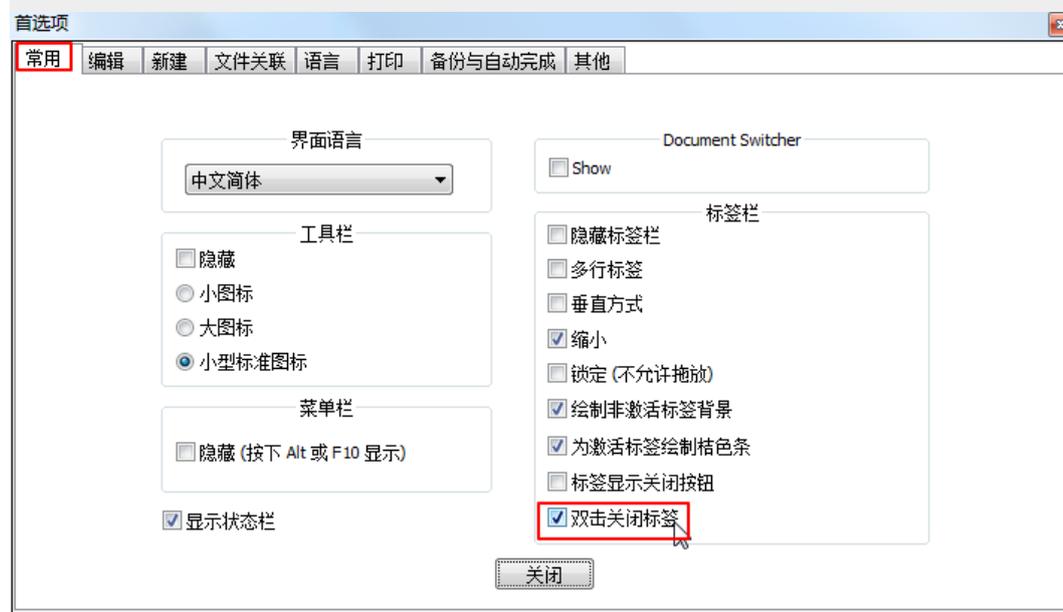




注意

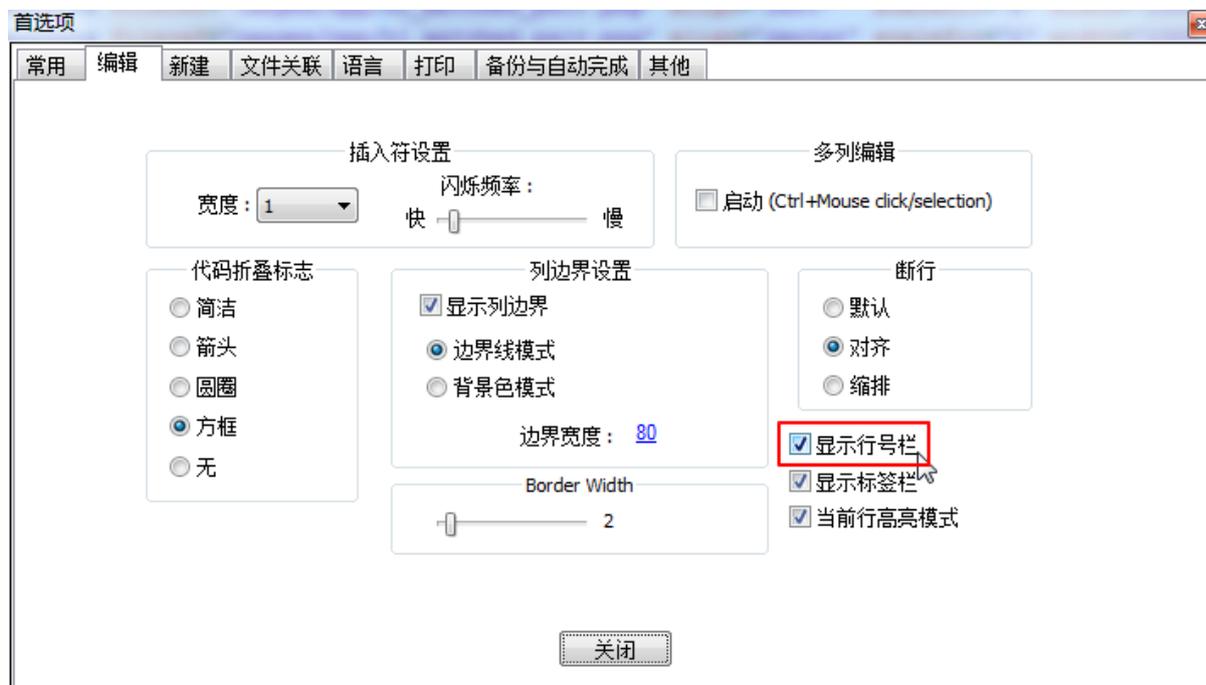
不过需要注意的是，默认是关闭了此功能的，需要的话，去开启此功能：

设置(T) ⇒ 首选项... ⇒ 常用 ⇒ 标签栏 ⇒ 勾选 "双击关闭标签"

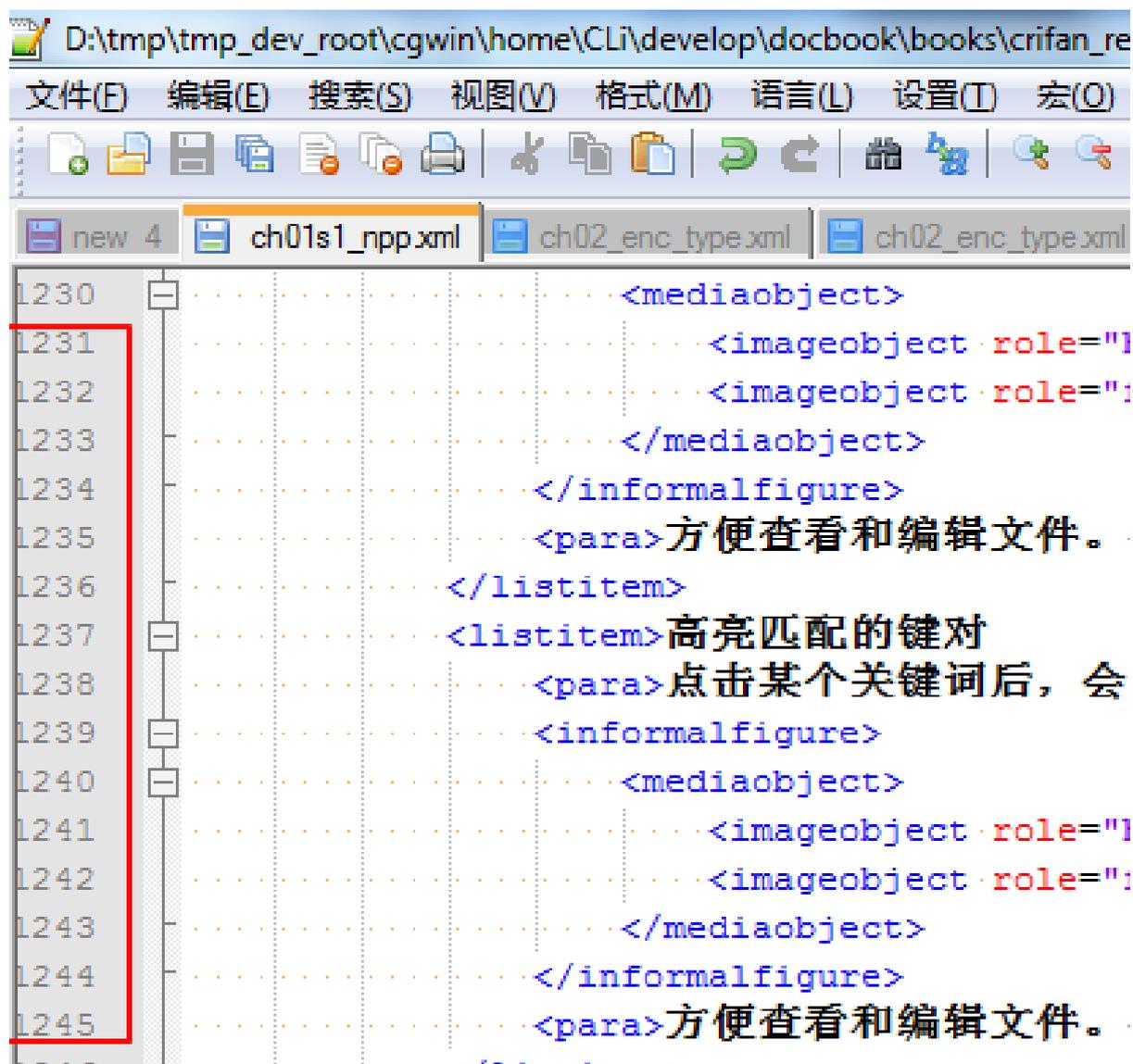


3.19.2. 显示文件行号

设置(T) ⇒ 首选项... ⇒ 编辑 ⇒ 勾选 "显示行号栏"



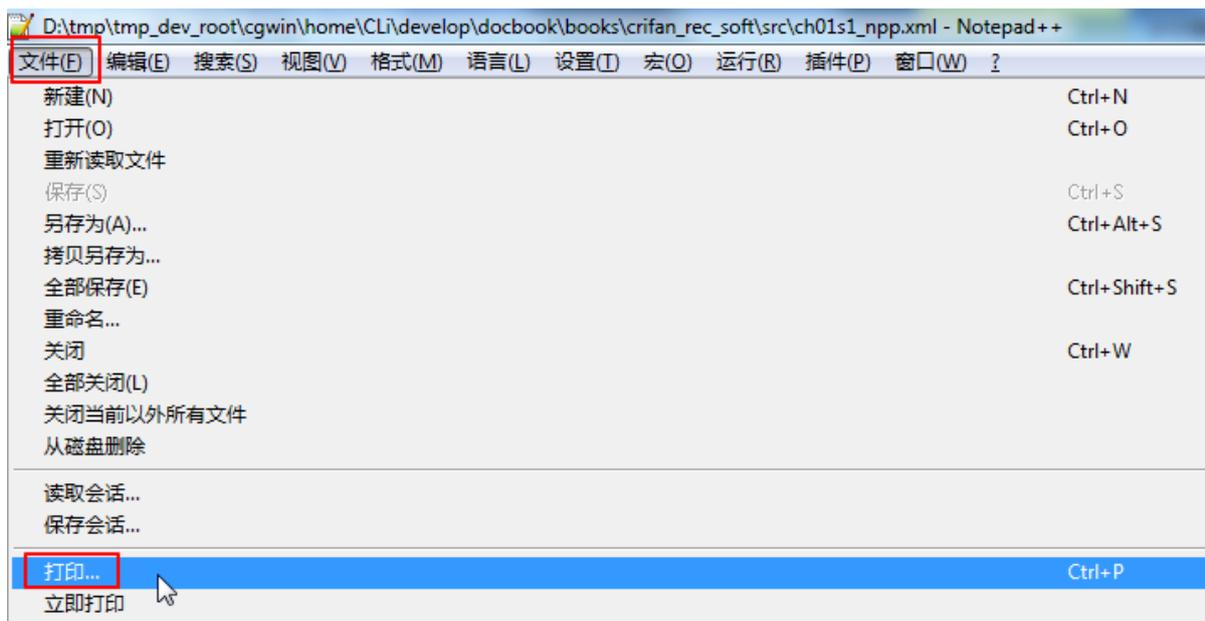
这样，就可以显示行号了：



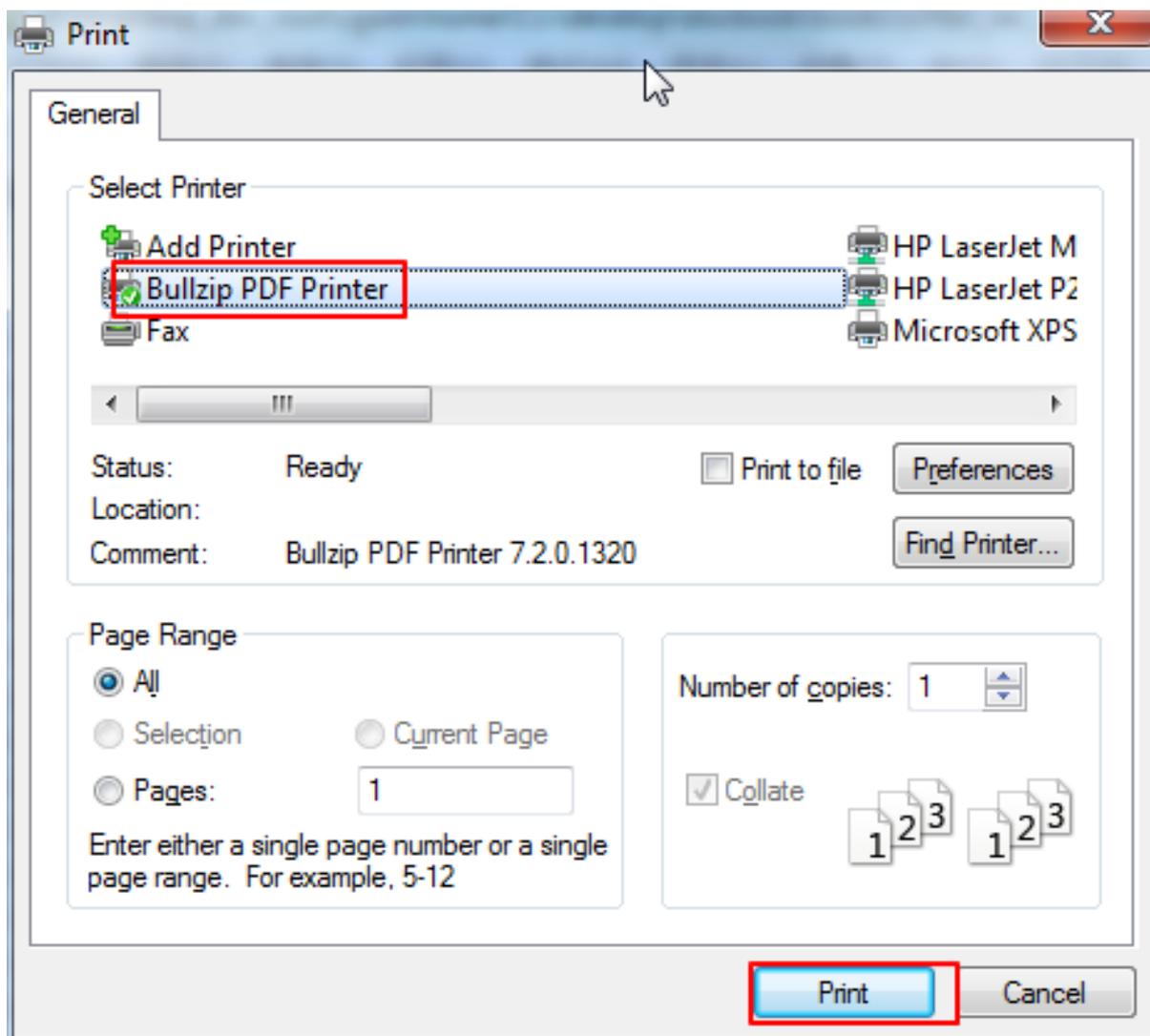
3.19.3. 支持是否打印行号

设置(T) ⇒ 首选项... ⇒ 打印 ⇒ 勾选 "打印行号"

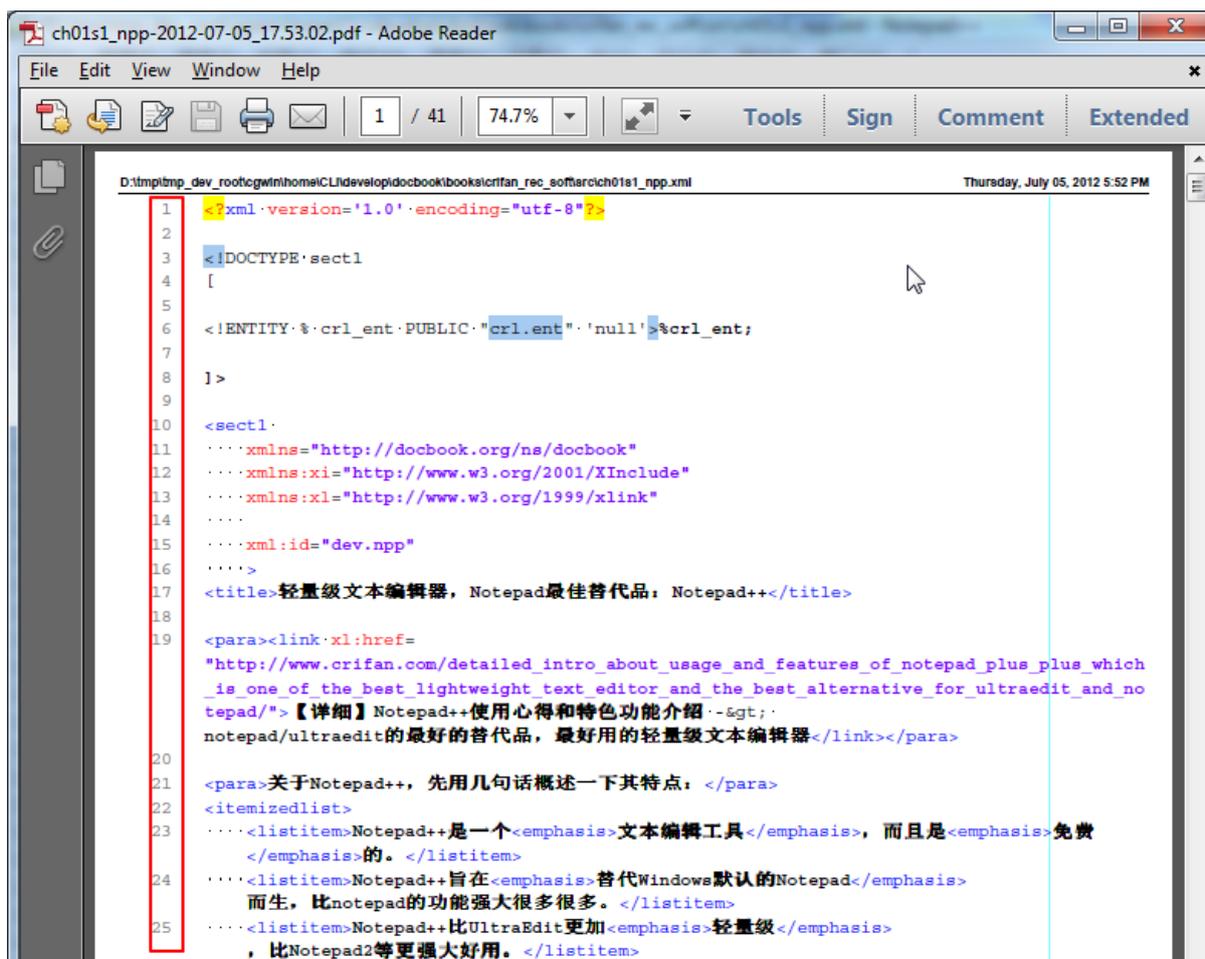
这样，再去：文件 ⇒ 打印



选择一个pdf打印机(BullZip PDF Printer), 然后打印 :



输出的pdf中, 就包括了行号了 :



```

1 <?xml version='1.0' encoding='utf-8'?>
2
3 <!DOCTYPE sect1
4 [
5
6 <ENTITY %crl_ent PUBLIC 'crl.ent' 'null' %crl_ent;
7
8 ]>
9
10 <sect1
11 .....xmlns='http://docbook.org/ns/docbook'
12 .....xmlns:xi='http://www.w3.org/2001/XInclude'
13 .....xmlns:xl='http://www.w3.org/1999/xlink'
14 .....
15 .....xml:id='dev.npp'
16 .....>
17 <title>轻量级文本编辑器，Notepad最佳替代品：Notepad++</title>
18
19 <para><link xl:href=
20 "http://www.crifan.com/detailed_intro_about_usage_and_features_of_notepad_plus_plus_which
21 _is_one_of_the_best_lightweight_text_editor_and_the_best_alternative_for_ultraedit_and_no
22 tepad/">【详细】Notepad++使用心得和特色功能介绍</link></para>
23
24 <para>关于Notepad++，先用几句话概述一下其特点：</para>
25 <itemizedlist
26 .....<listitem>Notepad++是一个<emphasis>文本编辑工具</emphasis>，而且是<emphasis>免费
27 </emphasis>的。</listitem>
28 .....<listitem>Notepad++旨在<emphasis>替代Windows默认的Notepad</emphasis>
29 而生，比notepad的功能强大很多很多。</listitem>
30 .....<listitem>Notepad++比UltraEdit更加<emphasis>轻量级</emphasis>
31 ，比Notepad2等更强大好用。</listitem>

```



提示

关于pdf打印机，不了解的可参考：[\[15\]](#)

3.19.4. 编辑文件后，是否备份

和Ultra Editor类似，Notepad++在编辑一个文件后，默认会生成.bak文件的。

不过对于notepad++来说，此功能默认是关闭的。需要的话可以去开启：

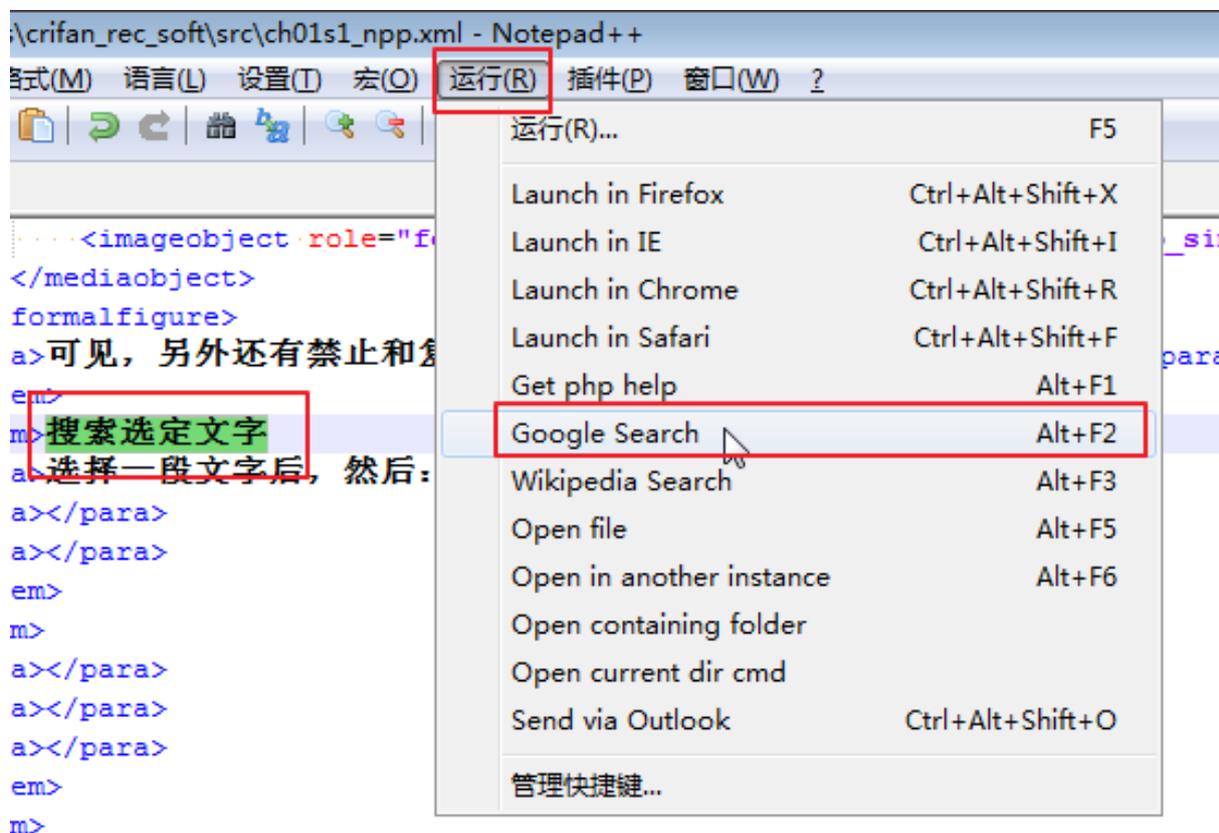
设置(T) ⇒ 首选项... ⇒ 备份 ⇒ 备份与自动完成 ⇒ 勾选 "简单备份"



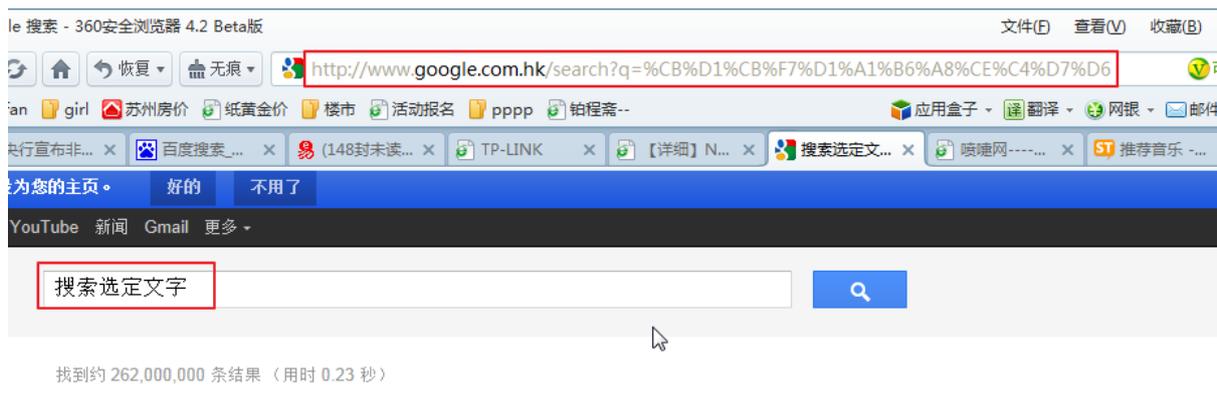
可见，另外还有禁止和复杂备份，可根据自己的实际情况而设置。

3.19.5. 调用搜索引擎搜索选定文字

选择一段文字后，然后：运行(R) ⇒ Google Search Alt+F2



就会调用google去搜索选定文字了：



与google search的，还有类似的：

- Get php help Alt+F1: 调用php帮助，对查找php函数等有帮助
- Wikipedia Search :调用维基百科搜索选定文字

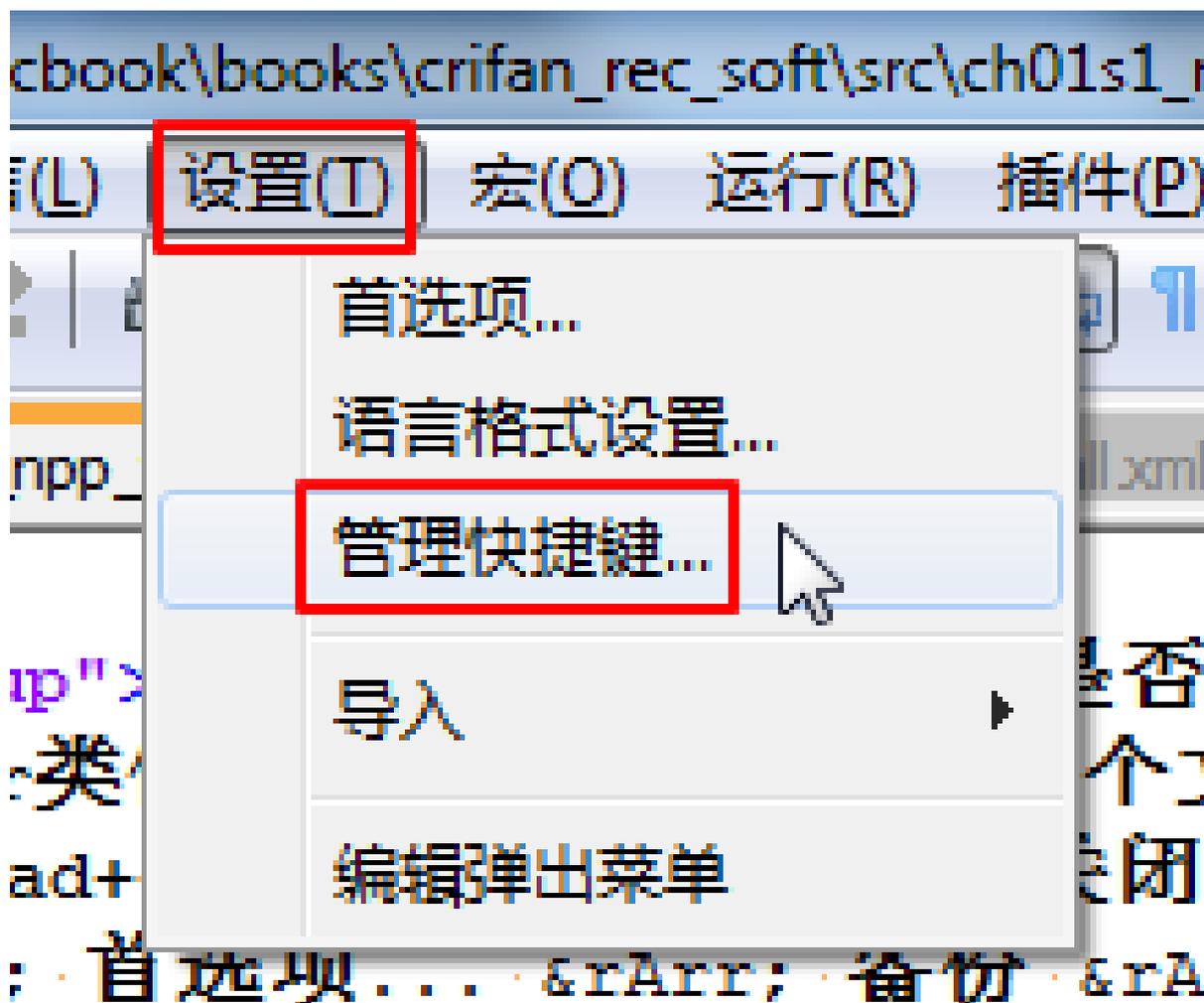
在同一界面的其他功能有：

- Open current dir cmd: 打开cmd且定位到当前文件夹，功能等价于：[\[14\]](#)
- Open cotaining folder: 打开当前文件所在文件夹

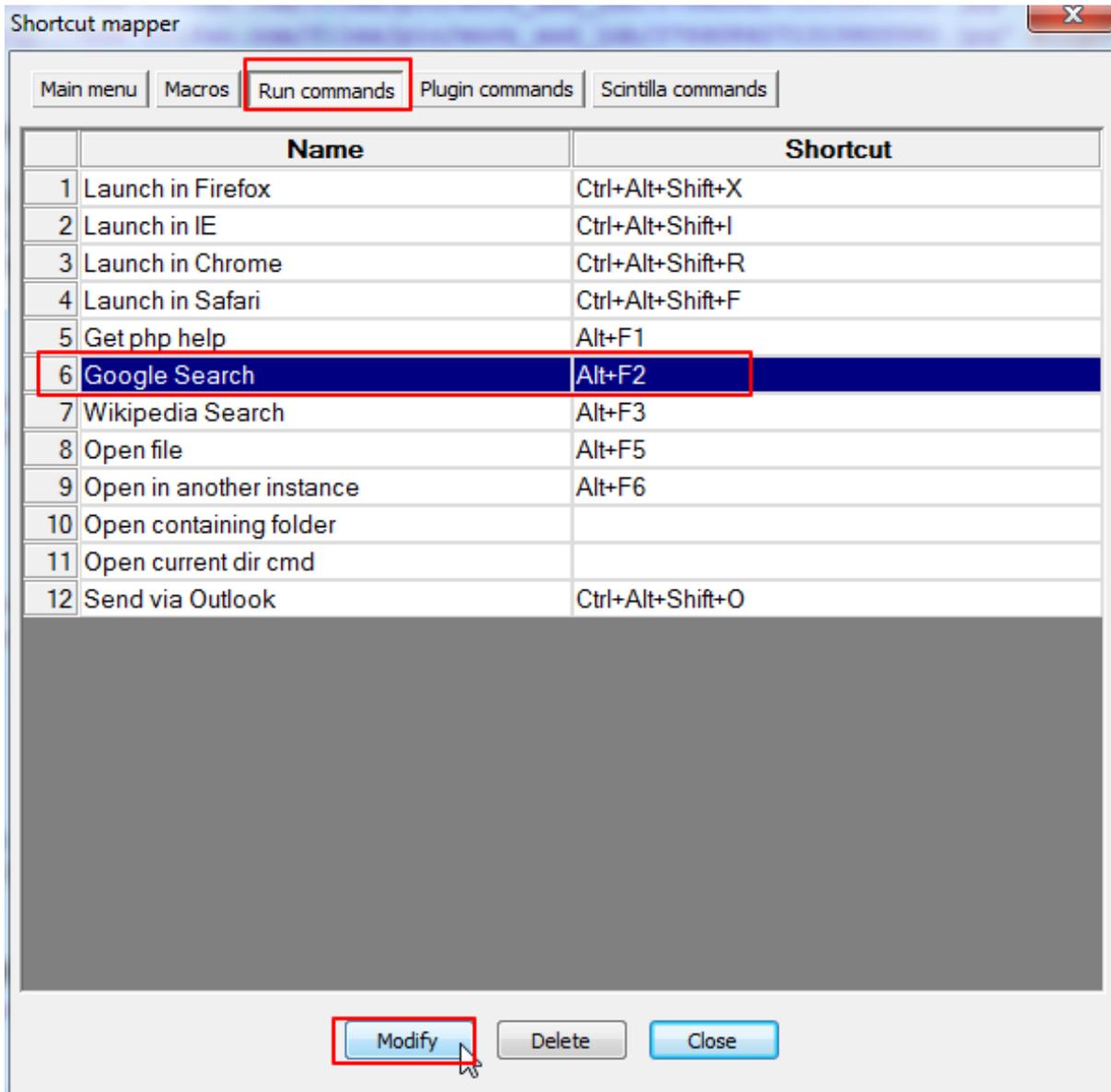
3.19.5.1. 更改google搜索功能的快捷键

如果想要改变这些Alt+F2之类的快捷键，Notepad++也是支持的：

设置(T) ⇒ 管理快捷键...



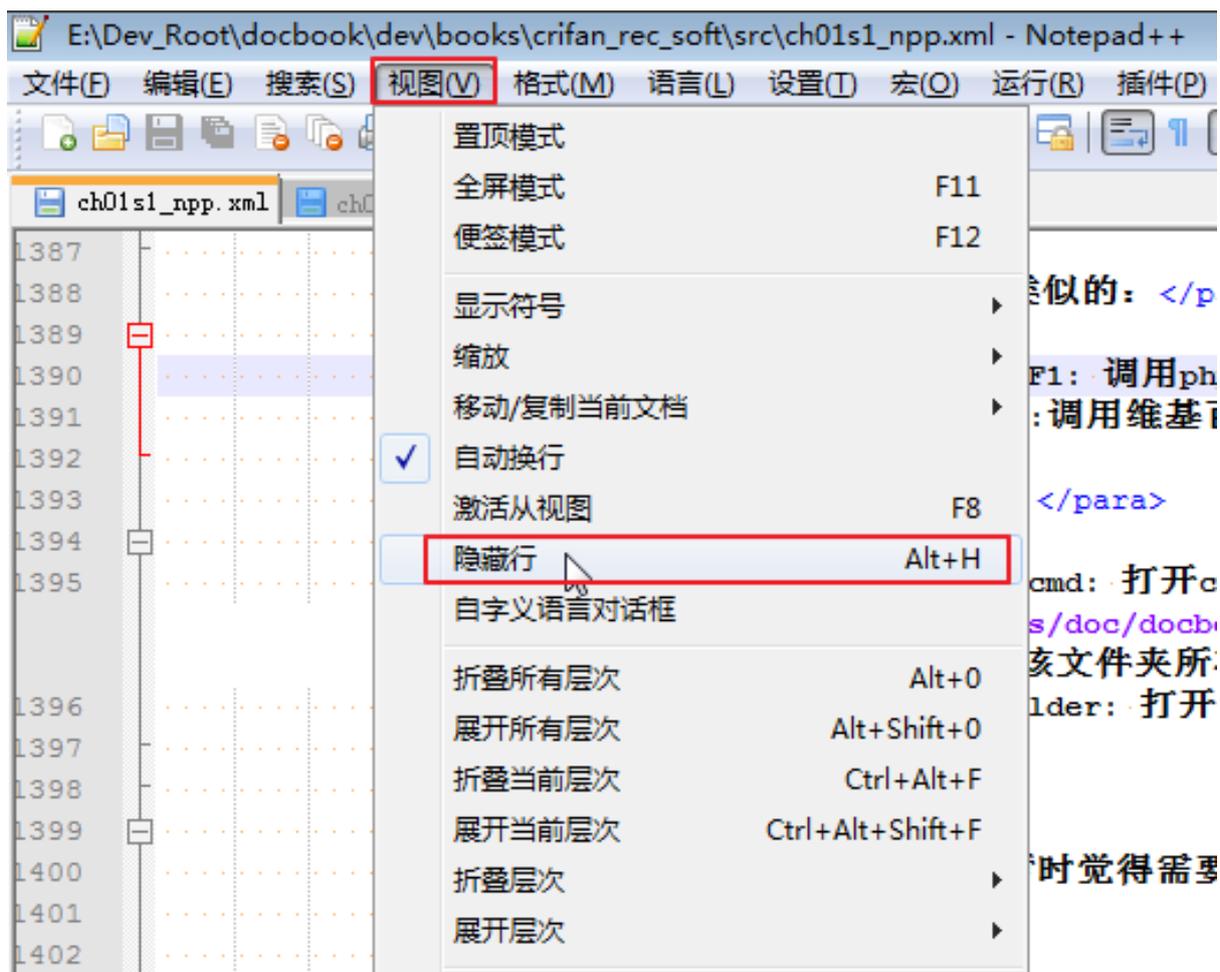
Run Commands下面，可以修改快捷键：



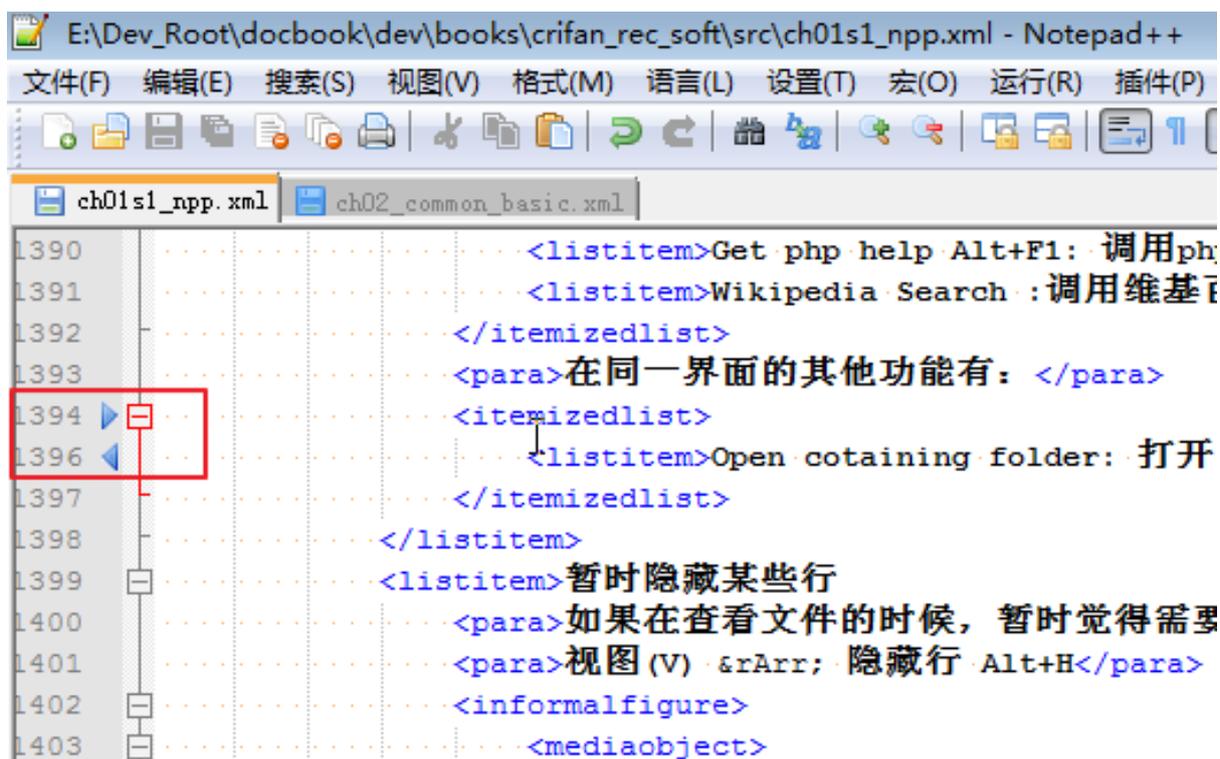
3.19.6. 暂时隐藏某些行

如果在查看文件的时候，暂时觉得需要将某些行隐藏掉，在需要的时候再正常显示，就可以先选中那些行，然后：

视图(V) ⇒ 隐藏行 Alt+H



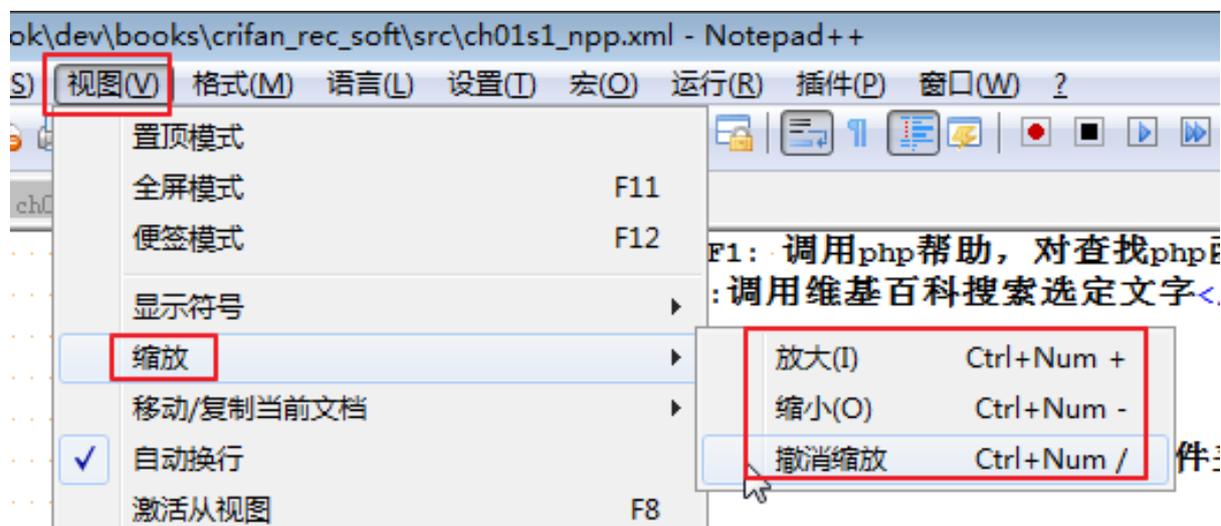
就可以实现暂时隐藏对应的行了：



很明显，点击对应的蓝色三角型，就又可以恢复显示该行了。

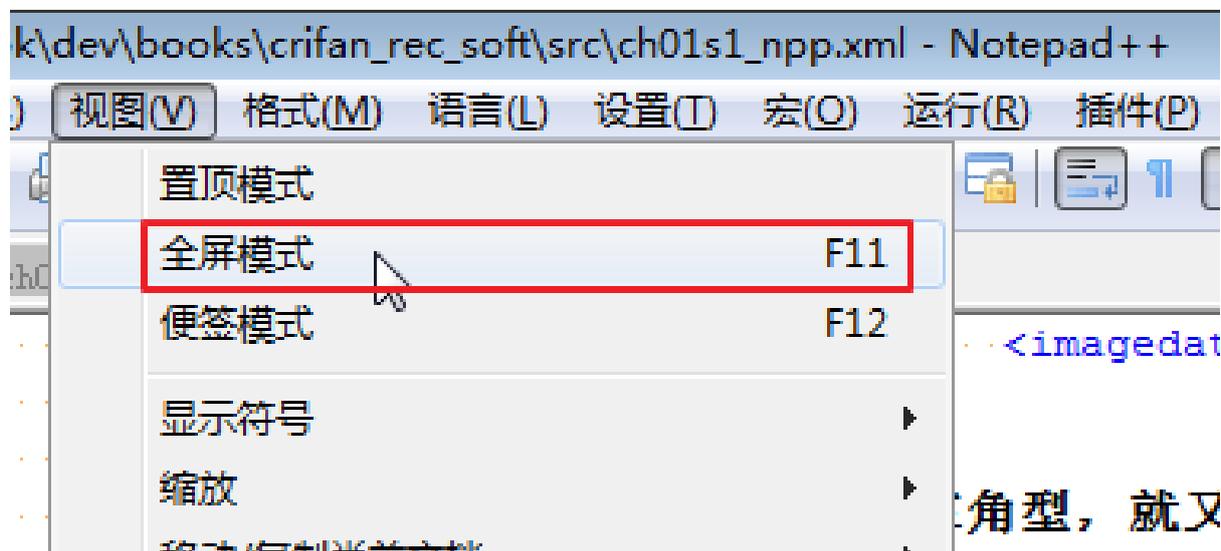
3.19.7. 支持鼠标滚轮控制缩放

Notepad++不仅支持快捷键Ctrl + Num+ / Num- 去控制放大/缩小，而且还支持Ctrl+鼠标滚轮去控制当前显示内容的缩放，很是方便。



3.19.8. 全屏显示

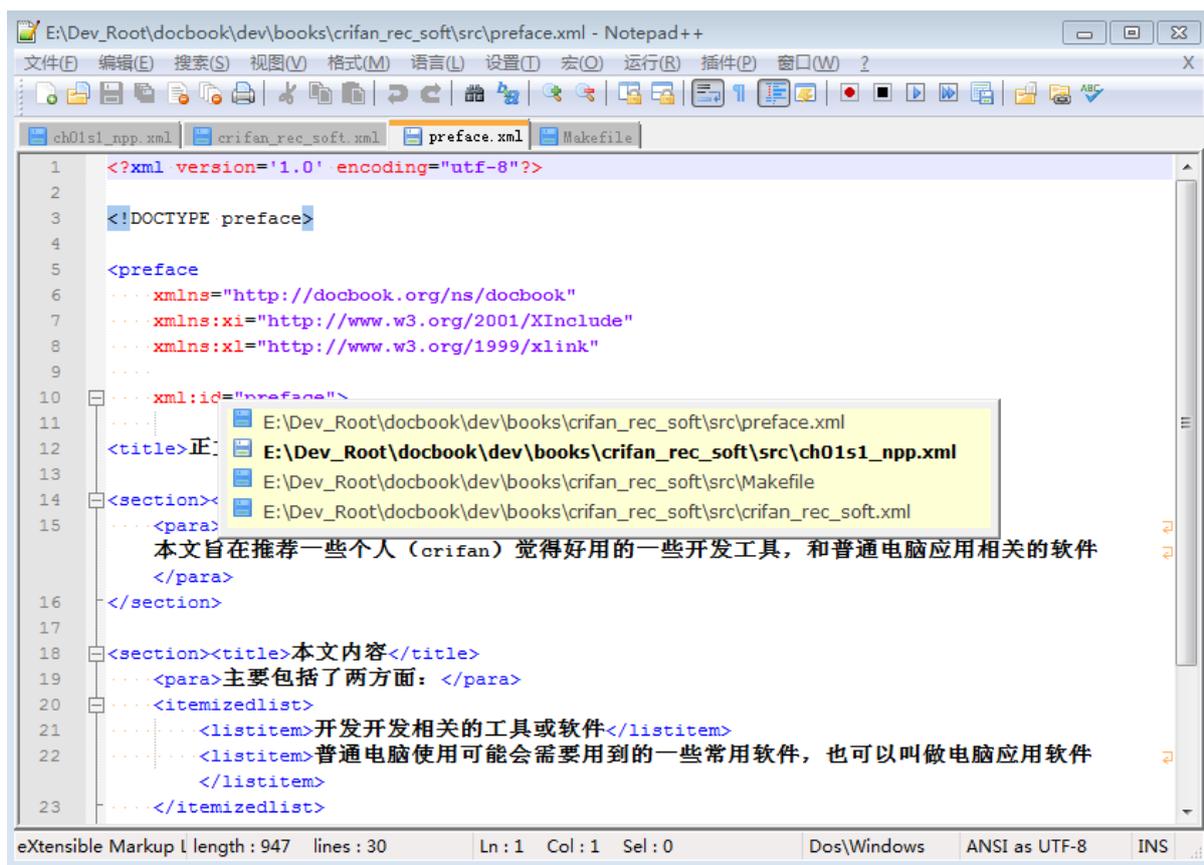
有时候显示文档的话，希望显示更多的内容，希望用到全屏显示，通过：视图(V) ⇒ 全屏显示 F11



实现全屏缩放。想要返回正常查看模式，点击右上角的加号，即可。

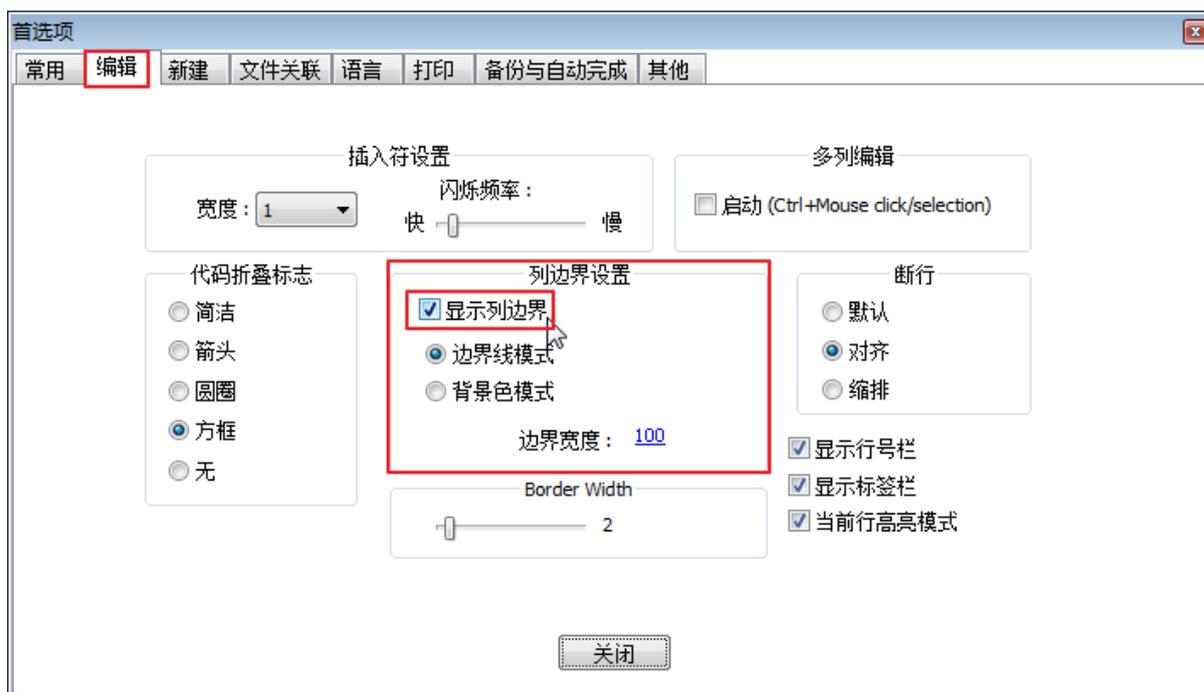
3.19.9. Ctrl+Tab实现在多个打开的窗口间切换

通过快捷键Ctrl+Tab，就可以实现在打开的多个文件中，来回切换：



3.19.10. 支持设置是否显示列边界（右边线）

设置(T) ⇒ 首选项... ⇒ 编辑 ⇒ 列边界设置 ⇒ 勾选 "显示列边界"和"边界线模式"



此时，文件中，宽度100的位置，就会有对应的一条竖线提示：

```

4472 <listitem>暂时隐藏某些行
4490 <listitem>支持鼠标滚轮控制缩放
4499 <listitem>全屏显示
4509 <listitem>Ctrl+Tab实现在多个打开的窗口间切换
4518 <listitem>支持设置是否显示列边界（右边界）
4519 <para>设置(T) &Arr; 首选项... &Arr; 编辑 &Arr; 列边界设置 &Arr; 勾选 "显示列边界"和"边界线模式"</para>
4520 <informalfigure>
4521 <mediaobject>
4522 <imageobject role="html"><imagedata fileref="images/npp/col_border_line.png" align="left" scale
4523 <imageobject role="fo"> <imagedata fileref="images/npp/col_border_line.png" align="center" scale
4524 </mediaobject>
4525 </informalfigure>
4526 <para>此时，文件中的内容，宽度100的位置，就会有对应的一条竖线提示：</para>
4527
4528 </listitem>
4529 <listitem>
4530 <para></para>
4531 <para></para>
4532 <para></para>
4533 </listitem>
4534 </listitem>

```

这样，如果代码是否超过便捷，就很容易看清了。



列边界功能的由来

此功能的出现，主要是由于对于写代码的人，一个好的习惯是，不要让代码太长。

一般的列宽限制，都定位80个字符。

超过的部分的代码，应该养成好习惯，以多行显示。

可以看到，另一种配置是把超过列边界的内容加上一定的底色，效果如下：

```

499 # following refer: http://python.u85.us/viewnews-335.html
500 para = {'hl':'zh-CN','ie':'UTF-8','text':strToTranslate,'langpair':'%s|%s'%(fromLanguage, toLanguage)}
501 url_google_trans = 'http://translate.google.cn/translate_t'
502 encoded_data = urllib.urlencode(para)
503 trans_req = urllib2.Request(url_google_trans, encoded_data)
504 # note: according to:
505 # http://imtx.me/archives/650.html
506 # here must use IE6, otherwise will return forbidden 403 error
507 trans_req.add_header('User-Agent', "Mozilla/4.0 (compatible;MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727)")
508 trans_resp = urllib2.urlopen(trans_req)
509 trans_soup = BeautifulSoup(trans_resp)
510 logging.debug("response html for translate %s:\n%s", strToTranslate, trans_soup.prettify())
511 result_box_span = trans_soup.find(id='result_box')
512 if result_box_span and result_box_span.span and result_box_span.span.string :
513     translated_str = result_box_span.span.string.encode('utf-8')
514     logging.debug("translate string: from='%s' to='%s'", strToTranslate, translated_str)
515 else :
516     logging.debug("***Error: can not translate string: %s", strToTranslate)
517
518 return translated_str
519
520
521 -----
522 # replace "&#39;" with "'" in "Creepin's up on you"
523 def replaceUnicodeNumbers(text):
524     rx = re.compile('[0-9]+;')
525     def one_xlat(match):
526         return unichr(int(match.group(0)[2:-1]))
527     return rx.sub(one_xlat, text)

```

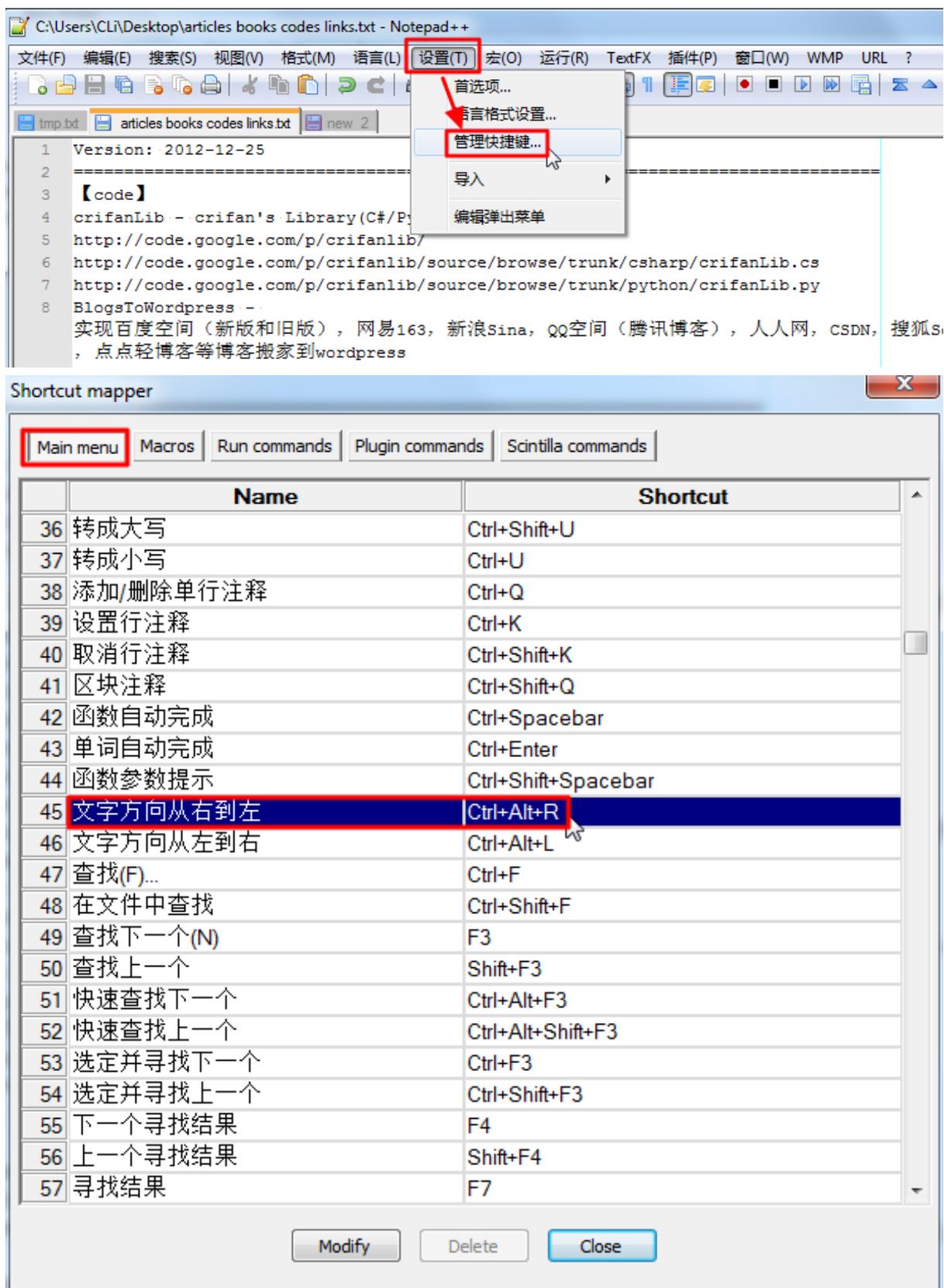
3.19.11. 文字显示：右对齐和左对齐

如果需要的话，可以设置文字右对齐。

3.19.11.1. 如何设置文字右对齐

默认的快捷键是：Ctrl+Alt+R

设置(T) ⇒ 管理快捷键... ⇒ Menu ⇒ 文字方向从右向左

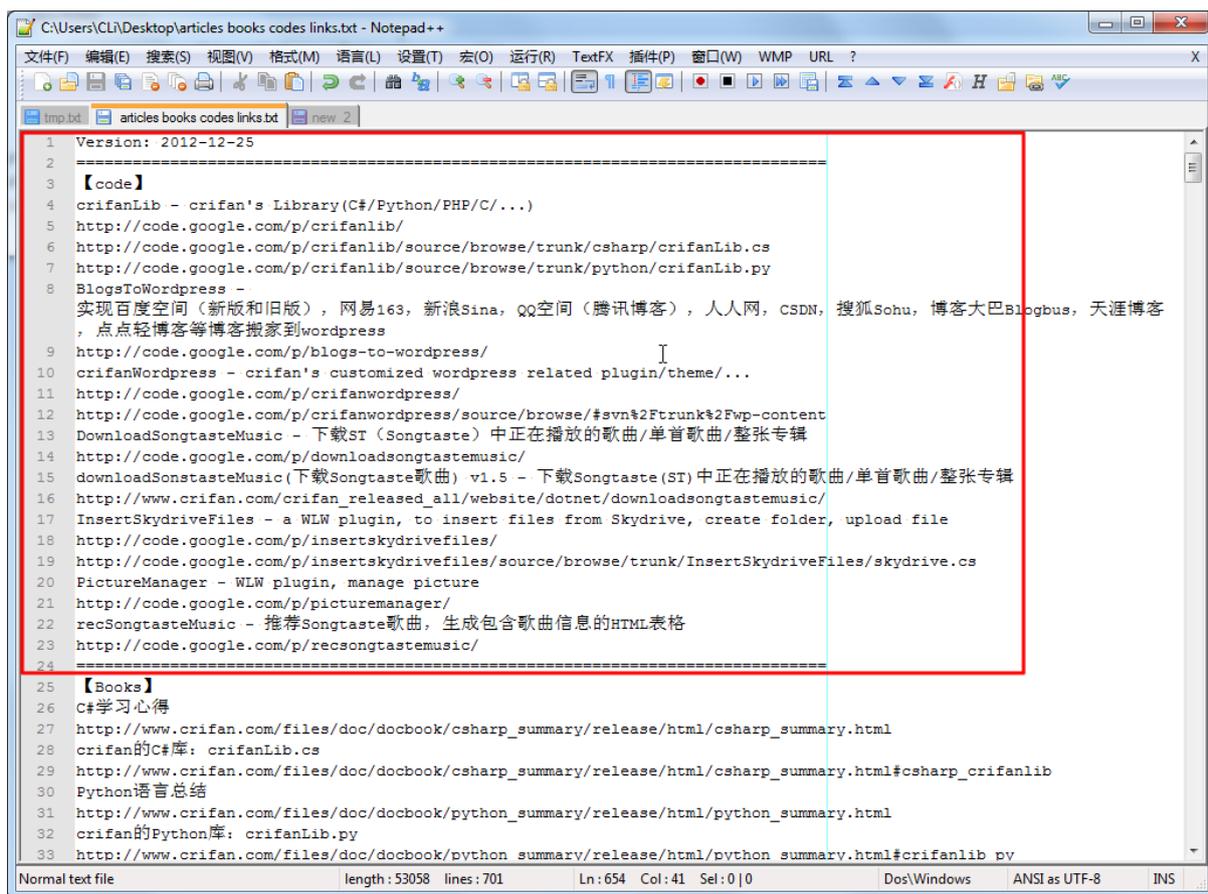


想要取消右对齐，可以使用快捷键Ctrl+Alt+L，就可以恢复到默认的左对齐了。

3.19.11.2. 文字右对齐的效果

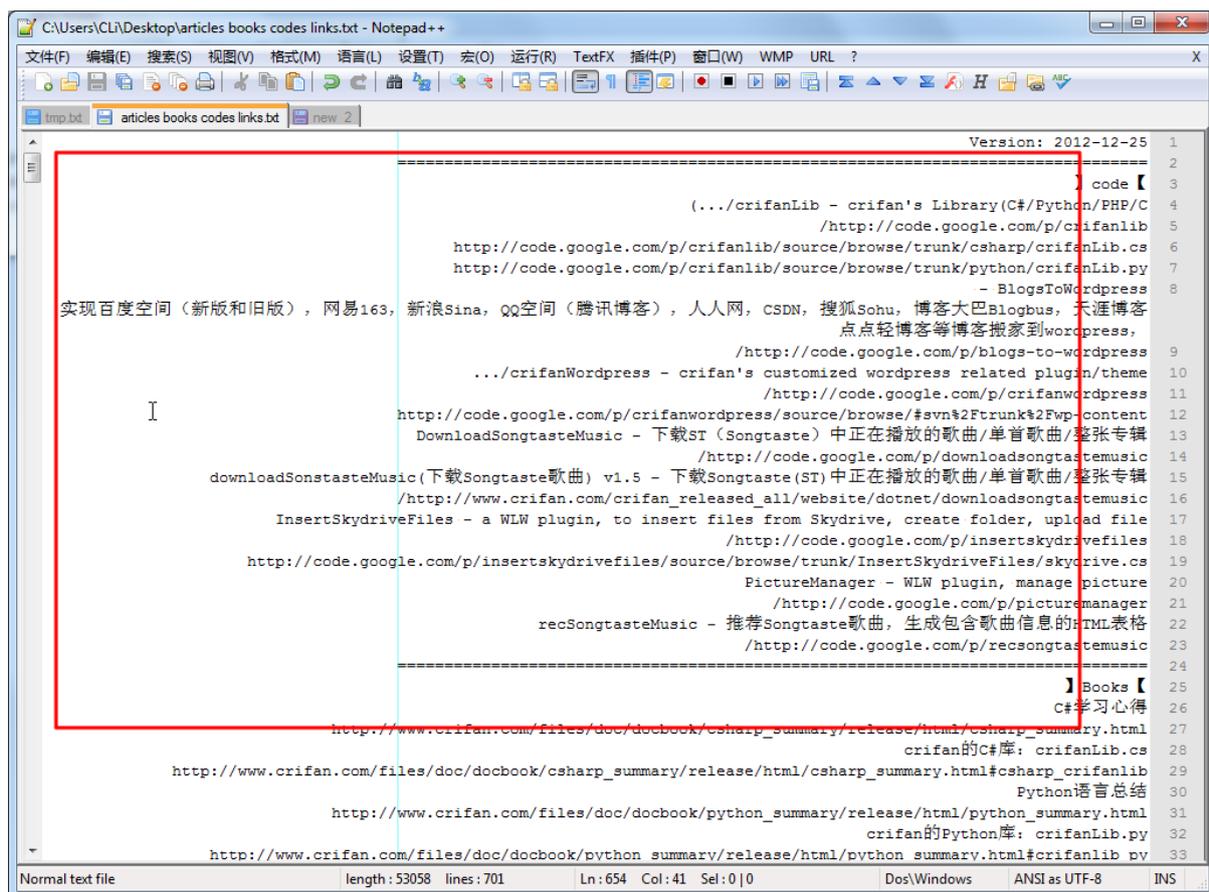
原先默认左对齐的是这样的：

Notepad++的功能详解



```
1 Version: 2012-12-25
2
3 【code】
4 crifanLib - crifan's Library(C#/Python/PHP/C/...)
5 http://code.google.com/p/crifanlib/
6 http://code.google.com/p/crifanlib/source/browse/trunk/csharp/crifanLib.cs
7 http://code.google.com/p/crifanlib/source/browse/trunk/python/crifanLib.py
8 BlogsToWordpress -
9 实现百度空间（新版和旧版），网易163，新浪Sina，QQ空间（腾讯微博），人人网，CSDN，搜狐Sohu，博客大巴Blogbus，天涯博客
10 ，点点轻博客等博客搬家到wordpress
11 http://code.google.com/p/blogs-to-wordpress/
12 crifanWordpress - crifan's customized wordpress related plugin/theme/...
13 http://code.google.com/p/crifanwordpress/
14 http://code.google.com/p/crifanwordpress/source/browse/#svn%2Ftrunk%2Fwp-content
15 DownloadSongtasteMusic - 下载ST（Songtaste）中正在播放的歌曲/单首歌曲/整张专辑
16 http://code.google.com/p/downloadsongtastemusic/
17 downloadSonstasteMusic(下载Songtaste歌曲) v1.5 - 下载Songtaste(ST)中正在播放的歌曲/单首歌曲/整张专辑
18 http://www.crifan.com/crifan_released_all/website/dotnet/downloadsongtastemusic/
19 InsertSkydriveFiles - a WLW plugin, to insert files from Skydrive, create folder, upload file
20 http://code.google.com/p/insertskydrivefiles/
21 http://code.google.com/p/insertskydrivefiles/source/browse/trunk/InsertSkydriveFiles/skydrive.cs
22 PictureManager - WLW plugin, manage picture
23 http://code.google.com/p/picturemanager/
24 recSongtasteMusic - 推荐Songtaste歌曲，生成包含歌曲信息的HTML表格
25 http://code.google.com/p/recsongtastemusic/
26
27 【Books】
28 C#学习心得
29 http://www.crifan.com/files/doc/docbook/csharp_summary/release/html/csharp_summary.html
30 crifan的C#库: crifanLib.cs
31 http://www.crifan.com/files/doc/docbook/csharp_summary/release/html/csharp_summary.html#csharp_crifanlib
32 Python语言总结
33 http://www.crifan.com/files/doc/docbook/python_summary/release/html/python_summary.html
34 crifan的Python库: crifanLib.py
35 http://www.crifan.com/files/doc/docbook/python_summary/release/html/python_summary.html#crifanlib_py
```

改为右对齐后，效果是：



3.19.12. 格式化（成漂亮的输出效果）

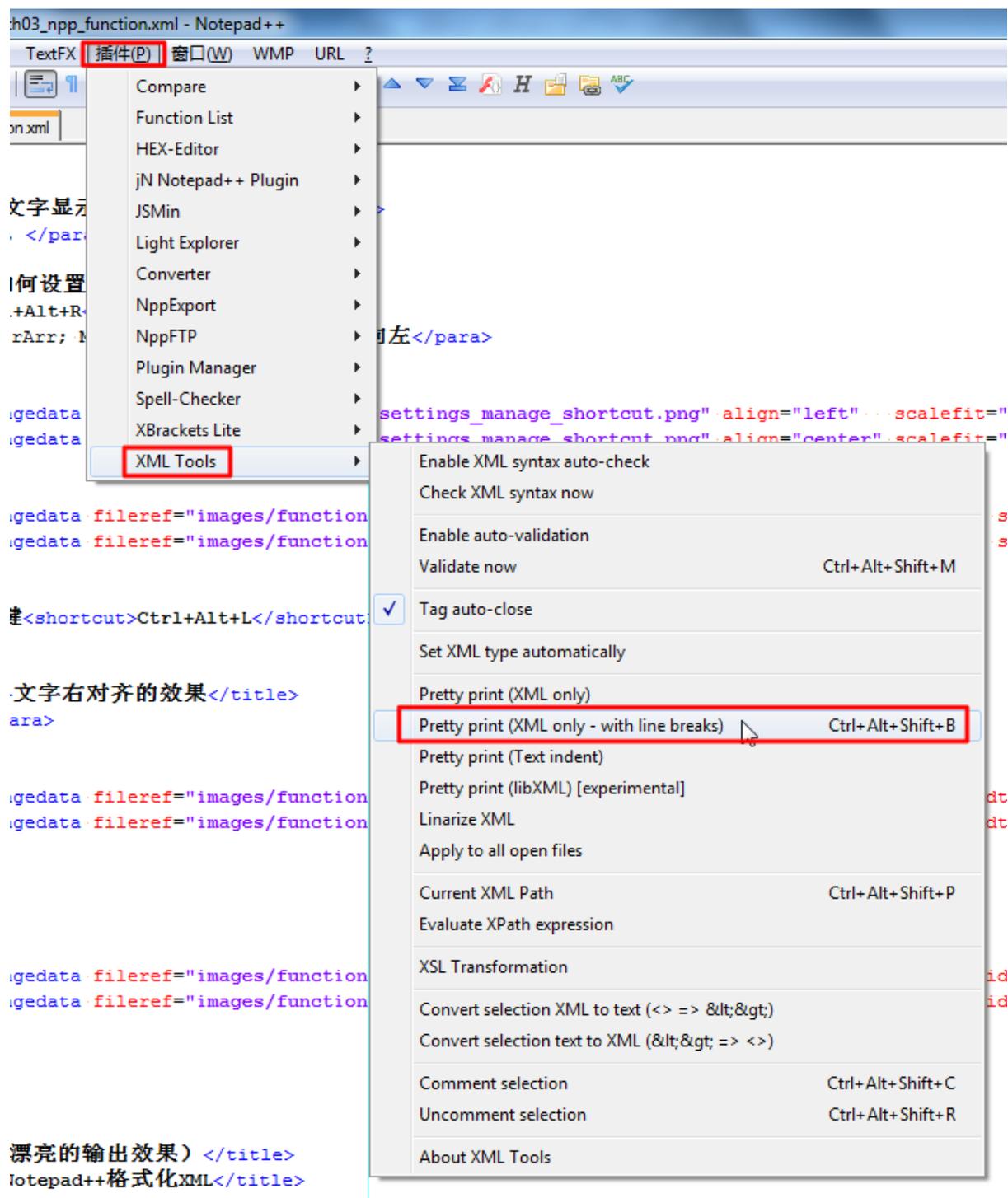
用Notepad++自带的功能，和其他第三方的插件，可以实现，针对不同的格式的文件格式化，下面依次分别介绍：

3.19.12.1. 用Notepad++格式化XML

有个插件叫做XML Tools

里面有个功能就是用来格式化输出XML的

Pretty print(XML only - with line breaks) Ctrl+Alt+Shift+B



3.19.13. Ctrl + D 拷贝单行或多行

Notepad++中，支持拷贝单行或选中的多行的功能

对应的快捷键是：**Ctrl+D**

之前，想要复制某一行，或某几行的话，都是：

用鼠标选中某行或某几行，然后**Ctrl+C**

然后光标再点击到下一行的行首，或者是多行的后面，

再去用Ctrl+V去粘贴

而后来，发现了Notepad++中，有个Ctrl+D的快捷键，

可以更加方便的实现同样的效果：

3.19.13.1. 直接复制光标所在的当前行的内容

Ctrl+D支持：

对于你当前所在的行，直接去用Ctrl+D

然后就可以复制出当前行的内容

比如，就在此刻，写docbook的xml源码时，对于：

```
<para></para>
```

鼠标放在中间：

```
.....<sect3 xml:id="copy_current_line"><title>直接复制光标所在的当前行
.....<para><command>Ctrl+D</command>支持： </para>
.....<para>对于你当前所在的行，直接去用<command>Ctrl+D</command>
.....<para>然后就可以复制出当前行的内容</para>
.....<para>比如，就在此刻，写docbook的xml源码时，对于： </para>
.....<para>|</para>
.....<para></para>      I
.....</sect3>
```

然后按Ctrl+D，就可以直接再复制出当前行了：

```
.....<sect3 xml:id="copy_current_line"><title>直接复制光标所
.....<para><command>Ctrl+D</command>支持： </para>
.....<para>对于你当前所在的行，直接去用<command>Ctrl+D</
.....<para>然后就可以复制出当前行的内容</para>
.....<para>比如，就在此刻，写docbook的xml源码时，对于：
.....<para>|</para>
.....<para></para>      I
.....<para></para>
```

如此，一下子，想要多复制个10行8行的，多按几次Ctrl+C就可以了。

就免去了之前繁琐的：赋值，移动光标，粘贴，的复杂操作了。

3.19.13.2. 复制选中的部分文字，多行内容等内容

Ctrl+D也支持，对于选择的内容，去复制出来的效果。

选择的内容，可以是部分文字，也可以是多行内容

比如，之前写docbook的xml时，想要基于当前的一个sect1的内容

复制一份出来，然后再去修改为想要的，第二个sect1的相关内容。

此时，就可以通过，选中整体的sect1的多行的内容

然后去按一次Ctrl+D，就可以达到想要的效果了：



```
17 <!-- XMASHS:HS="http://gocubook.org/hs/gocubook">
18 <title>嵌入式开发基础知识</title>
19 <para></para>
20 <para></para>
21 <para></para>
22
23 <sect1 xml:id="embedded_common_concept"><title>嵌入式领域的基本概念</title>
24 <para>关于嵌入式开发，其实也有很多的背景知识</para>
25 <para>了解了背景知识，总体概念后，才能真正搞懂每个细节内部具体实现的原因</para>
26 <para></para>
27 <para>关于嵌入式开发的总体流程和逻辑，可参考：</para>
28 <para><link xl:href="http://www.crifan.com/embedded_series_tutorial/">嵌入式开发系列教程</link></para>
29 <para>对应的，每个部分的介绍，分别是：</para>
30 <orderedlist>
31 <listitem><link xl:href="http://www.crifan.com/embedded_development_background">嵌入式开发之嵌入式背景知识</link></listitem>
32 <listitem><link xl:href="http://www.crifan.com/embedded_development_basic_concept_develop_logic/">嵌入式开发之基本概念和开发逻辑</link></listitem>
33 <listitem><link xl:href="http://www.crifan.com/embedded_development_cross_compiler_and_toolchain/">嵌入式开发之交叉编译器</link></listitem>
34 <listitem><link xl:href="http://www.crifan.com/embedded_development_bootloader/">嵌入式开发之bootloader</link></listitem>
35 <listitem><link xl:href="http://www.crifan.com/embedded_development_kernel/">嵌入式开发之kernel</link></listitem>
36 <listitem><link xl:href="http://www.crifan.com/embedded_development_rootfs/">嵌入式开发之rootfs</link></listitem>
37 <listitem><link xl:href="http://www.crifan.com/embedded_development_burn_tool_and_methods/">嵌入式开发之烧录工具和烧录方式</link></listitem>
38 </orderedlist>
39 <para></para>
40 <para></para>
41 </sect1>
42
43 <sect1 xml:id="embedded_common_concept"><title>嵌入式领域的基本概念</title>
44 <para>关于嵌入式开发，其实也有很多的背景知识</para>
45 <para>了解了背景知识，总体概念后，才能真正搞懂每个细节内部具体实现的原因</para>
46 <para></para>
47 <para>关于嵌入式开发的总体流程和逻辑，可参考：</para>
48 <para><link xl:href="http://www.crifan.com/embedded_series_tutorial/">嵌入式开发系列教程</link></para>
49 <para>对应的，每个部分的介绍，分别是：</para>
50 <orderedlist>
51 <listitem><link xl:href="http://www.crifan.com/embedded_development_background">嵌入式开发之嵌入式背景知识</link></listitem>
52 <listitem><link xl:href="http://www.crifan.com/embedded_development_basic_concept_develop_logic/">嵌入式开发之基本概念和开发逻辑</link></listitem>
53 <listitem><link xl:href="http://www.crifan.com/embedded_development_cross_compiler_and_toolchain/">嵌入式开发之交叉编译器</link></listitem>
54 <listitem><link xl:href="http://www.crifan.com/embedded_development_bootloader/">嵌入式开发之bootloader</link></listitem>
55 <listitem><link xl:href="http://www.crifan.com/embedded_development_kernel/">嵌入式开发之kernel</link></listitem>
56 <listitem><link xl:href="http://www.crifan.com/embedded_development_rootfs/">嵌入式开发之rootfs</link></listitem>
57 <listitem><link xl:href="http://www.crifan.com/embedded_development_burn_tool_and_methods/">嵌入式开发之烧录工具和烧录方式</link></listitem>
58 </orderedlist>
59 <para></para>
60 <para></para>
61 </sect1>
```

如此，高效的实现了，多行的复制。

参考书目

- [1] [【详细】Notepad++使用心得和特色功能介绍 -> notepad/ultraedit的最好的替代品，最好用的轻量级文本编辑器](#)¹
- [2] [Difference between ansi and unicode version](#)²
- [3] [支持世界上几乎所有字符的字符编码：Unicode](#)³
- [4] [Notepad++中的“获取插件”](#)⁴
- [5] [v0.9.5](#)⁵
- [6] [Hex Editor](#)⁶
- [7] [字符编码详解](#)⁷
- [8] [中文字符编码标准+Unicode+Code Page](#)⁸
- [9] [关于UTF-8的BOM：“EF BB BF”](#)⁹
- [10] [Notepad++ Regex Backreference missing](#)¹⁰
- [11] [How to use regular expressions in Notepad++ \(tutorial\)](#)¹¹
- [12] [Regular Expressions in SciTE](#)¹²
- [13] [【详解】回车 换行 0x0D 0x0A CR LF \r \n的来龙去脉](#)¹³
- [14] [右键快速打开cmd并且定位到该文件夹所在路径](#)¹⁴
- [15] [PDF打印机：Bullzip PDF Printer](#)¹⁵

¹ http://www.crifan.com/detailed_intro_about_usage_and_features_of_notepad_plus_plus_which_is_one_of_the_best_lightweight_text_editor_and_the_best_alternati

² <http://sourceforge.net/projects/notepad-plus/forums/forum/331754/topic/4955411>

³ http://www.crifan.com/files/doc/docbook/char_encoding/release/html/char_encoding.html#enc_unicode

⁴ http://sourceforge.net/apps/mediawiki/notepad-plus/index.php?title=Plugin_Central

⁵ <http://sourceforge.net/projects/npp-plugins/files/Hex%20Editor/Hex%20Editor%20Plugin%20v0.9.5/>

⁶ <http://sourceforge.net/projects/npp-plugins/files/Hex%20Editor/>

⁷ http://www.crifan.com/files/doc/docbook/char_encoding/release/html/char_encoding.html

⁸ http://www.crifan.com/files/doc/docbook/zhcn_charset/release/html/zhcn_charset.html

⁹ http://www.crifan.com/files/doc/docbook/char_encoding/release/html/char_encoding.html#utf8_bom_efbbbf

¹⁰ <http://stackoverflow.com/questions/8311027/notepad-regex-backreference-missing>

¹¹ http://sourceforge.net/apps/mediawiki/notepad-plus/index.php?title=Regular_Expressions#Example_1

¹² <http://www.scintilla.org/SciTERegEx.html>

¹³ http://www.crifan.com/detailed_carriage_return_0x0d_0x0a_cr_lf_r_n_the_context/

¹⁴ http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#right_menu_run_cmd

¹⁵ http://www.crifan.com/files/doc/docbook/crifan_rec_soft/release/html/crifan_rec_soft.html#common.bullzip_pdf_printer