命令：wv\_sample\_exporter.exe input output -p c

## 1、wv\_sample\_exporter.exe路径是可以包含中文的，测试下面命令是能正常转换

"E:/中文/WebViewExport/wv\_sample\_exporter.exe" E:/english/WebViewExport/1M.doc E:/english/WebViewExport/output/1M.html -p "c"

## 2、测试 input 和 output 的中文情况，甚至是其他语言的情况

### （1）input可以包含中文

input的中文可以出现在路径上及文件名上。文件名中出现其他语言也可以（已进行过8种语言的测试），目录路径出现其他语言的没有测试过（只测试过中、英文）

### （2）output不能包含中文

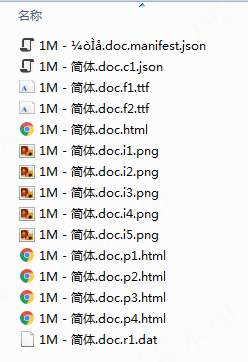
如果是文件名有中文，就转换成功，但无法预览，报 Failed to load resource: net::ERR\_FILE\_NOT\_FOUND。

如果output参数的中文（俄语、日语、简繁体）是出现在路径上，会报下面错误

EXRunExport() failed: filter couldn't allocate enough memory (0x0003)

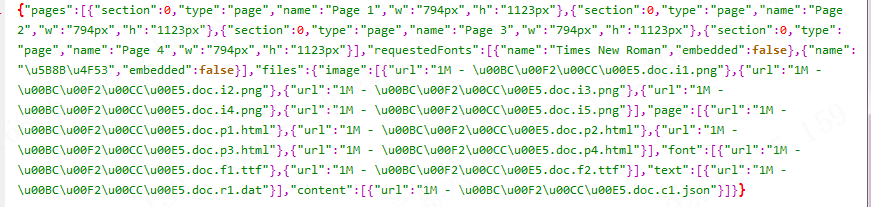
然后我们把路径改为全英，只有文件名中有中文（俄语、日语、简繁体），可以进行转换，但转换出来会有一个文件名乱码，但其它文件名是正常的，如

1M - 简体.doc 转换后：

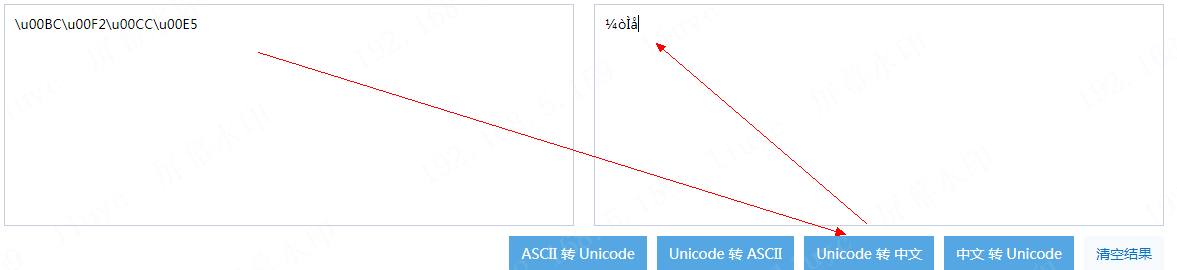


这个文件乱码的后果是在预览的时候，报Failed to load resource: net::ERR\_FILE\_NOT\_FOUND

然后我们查看该乱码文件的内容



并把部分内容进行转码，发现转出来就已经是乱码了，所以找的文件都会是1M - ¼òÌå.c1.json之类的，但生成的是1M - 简体.c1.json之类的正确的文件名



1M - ドラゴンボール.doc ===> 1M - ¥É¥é¥´¥ó¥Ü©`¥ë.doc.manifest.json

只有这一个文件乱码，如果可以解决这个文件名及它的内容编码，那预览应该是没有问题。

## 3、其他语言的情况

注意：下面的测试是以output参数为此格式 output/文件名/文件名.html

例如，测试日文时，output为：output/1M - ドラゴンボール.doc/1M - ドラゴンボール.doc.html

阿拉伯语、泰语、韩语：

DAOpenDocument() failed: no file is currently open (0x0010)

child process exited with code:16

俄语、日语、简体、繁体：

EXRunExport() failed: filter couldn't allocate enough memory (0x0003)

child process exited with code:3

--------------------------------------------------------------------

注意：然后尝试把路径改为全英，只是生成文件保留测试语言字符

例如，测试日文时，output为：output/1566443068719/1M - ドラゴンボール.doc.html

阿拉伯语、泰语、韩语：仍然是相同的0x0010错误。

俄语、日语、简体、繁体：不报0x0003错误，而是[第二点](#_（2）output也可以包含中文)提到的乱码