# Oracle Web View Export 8.5.3.0 文档

某次测试组发现，win7需要安装sp1才能成功转换。

## 1、使用说明

1、下载。<https://www.oracle.com/technetwork/cn/middleware/webcenter/content/downloads/oit-dl-otn-088510-zhs.html>，中下载Web View Export 8.5.3（目前只能在Windows 64位和linux64位上运行）。

2、解压。并把redist的所有文件，拷贝到项目静态资源的某一个目录中，把sdk/demo中的wv\_sample\_exporter.exe也拷贝到同一个目录上。（redist中有个jar包，是java demo需要的，所以可能有一些文件并不是wv\_sample\_exporter.exe进程所必须的）

3、测试是否成功。随便一个路径新建一个测试目录（如c:\test），打开cmd输入

"..path..\wv\_sample\_exporter.exe" "源文件路径\a.docx" "c:\test\output.html" –p "c"

（双引号防止路径带空格）正常会输出生成文件之类的语句，观察c:\test里是否生成了文件。

4、显示。使用webstorm新建一个web项目。引入以下文件，这些文件都在sdk/assets目录中，把它们都放进web项目的静态资源目录中。并把刚才测试生成的c:\test目录下的所有文件都放进web项目如review目录中。

<link rel="stylesheet" href="static/assets/oit.css">

<link rel="stylesheet" href="static/assets/oit.navbar.css">

<link rel="stylesheet" href="static/assets/oit.statusbar.css">

<!—body中增加一个标签，要设置高度-->

<div class="myViewContainer" id="container"></div>

<script src="static/assets/oit.js"></script>

<!--打开左侧的apitest-->

<!--<script src="static/assets/apitest.js"></script>-->

<!--打开顶部操作栏-->

<script src="static/assets/oit.navbar.js"></script>

<!--底部状态栏-->

<script src="static/assets/oit.statusbar.js"></script>

<script>

var viewContainer = document.getElementById("container");

OIT.view.attach(viewContainer);

OIT.view.load("review/output.html");

</script>

以上是最基本的转换文件并进行网页预览的示例。

用户手册：<https://docs.oracle.com/outsidein/852/oit/OIWVD/toc.htm>

## 2、可控性

### 2.1、转换控制

转换命令：wv\_sample\_exporter InputFile OutputFile

其他可选参数如下：

|  |  |  |
| --- | --- | --- |
| 可选项 | 参数 | 说明 |
| -p | 'f', 'c', or 's', indicating structure that is flat, chunked (Ajax), or streaming (Ajax) | Chooses the structure of the output files generated during the export process. |
| -r | ResourcePath | ResourcePath is the base URL for .js, .css, and .png assets. |
| -o | OutputPath | OutputPath is the base URL for links to generated output files. |
| -s | ScriptUrl | ScriptUrl is a the URL of a javascript file to be linked from the output files. |
| -c | CssUrl | CssUrl is the URL of a CSS file to be linked from the output files. |
| -n | ArchiveNode | ArchiveNode indentifies a node from the input file, to be used as the exported document (applies only to archive files, e.g., .zip) |
| -a | AttachmentId | AttachmentId identifies an attachment from the input file, to be used as the exported document (applies only to email message files.) |
| -k | Key, Value | Key and Value are a key/value pair to be made available to scripts |
| -f | FontDirectory | FontDirectory indicates where the exporter should find fonts to use. |

虽然是可选，但我们必须把-p "c"加上（-p "s"也行，未发现两者是什么区别，生成的文件c比s多了ttf文件），必须加上，代表使用ajax mode，不然生成的文件在预览的时候会报该文件不是通过ajax mode生成的无法预览。

### 2.2、显示控制

这里只说明javascript api，其他语言不作说明。

oit.css和oit.js是必须的，提供OIT全局变量，然后通过OIT调用各种API。

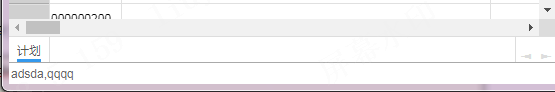
assets中的图片资源也是必须的，用来显示某些工具图标。

oit.navbar.css和oit.navbar.js是用来显示下图的工具栏



oit.statusbar.css和oit.statusbar.js用来显示底部状态栏

webview\_statusbar.message(["adsda","qqqq"]);



主要有9大模块可以控制：

**OIT.archive**: This object allows the application to listen to events related to archive files.

**OIT.attachments**: This object allows the application to listen to events related to email attachments.

**OIT.document**: Methods of this object control zooming, searching, and selection of the content.

**OIT.highlights**: Management of highlighted text, cell ranges, or rect ranges.

**OIT.hyperlinks**: This object allows the customer to listen to events related to hyperlinks.

**OIT.local**: Localized user interface strings.

**OIT.pages**: Methods of this object control navigation through the pages of the document.

**OIT.toolbars**: This object allows the application to add custom toolbars to the view.

**OIT.view**: This object provides control over the view area.

每个模块都有很多子方法，如

OIT.view.load( url ) 设置显示的文件

OIT.document.setOptions( { 设置某些属性

emailHeaderStyle: {

font: "bold 12px Arial, sans-serif",

color: "red"

}

});

同样也具有事件监听的方法，如

超链接的点击事件

OIT.hyperlinks.addEventListener( "click", function(evt) {

if( evt.detail.external ) /\* don't follow external links \*/

evt.preventDefault();

});

具体想控制什么，可以从这里<https://docs.oracle.com/outsidein/852/oit/OIWVD/GUID-4BF8064A-B837-4DF0-98E1-78F0B3D7B9C4.htm#OIWVD627>查找需要调用什么方法。

## 3、接口

我们可以打开dotnet版本的wv\_sample\_exporter项目，里面有一句

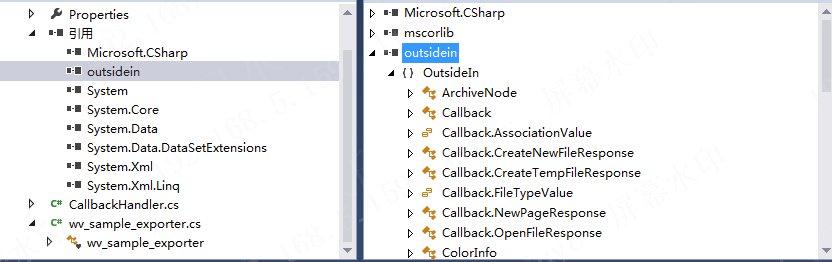
Exporter exp = OutsideIn.OutsideIn.NewLocalExporter()

exp.SetSourceFile(args[0]);

exp.Export();

等等

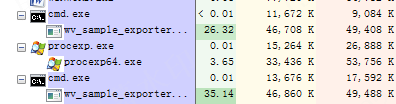
然后查看一下Exporter来之outsidein.dll，并且发现里面有很多的方法可以使用。



在php中，尝试使用COM来调用outsidein.dll，但在注册的时候失败，提示没有入口注册失败，所以应该是一个非COM的dll，然后根据网上的说明使用DynamicWrapper.dll来调用非COM的dll，但DynamicWrapper.dll的命令参数类型有限，对布尔型、对象型的数据都不知道怎么处理，执行就报参数错误，所以也未能成功调用接口。

## 4、同时转换情况

可以同时转换



## 5、网络情况

测试在断网的情况下仍然能进行转换。

## 6、测试情况

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试环境：  Windows 7 企业版  处理器：Inter(R) Pentium(R) CPU G620 @ 2.60GHz 2.60GHz  安装内存：8.00GB  系统类型：64位操作系统  转换命令：  "wv\_sample\_exporter.exe" "download5.xls" "output\output.html" –p "c" | | | | | | |
| 文件类型 | 页数或行数 | 文件大小（KB） | 测试次数 | 大概用时 | CPU和内存（procexp看到的数值） | 转换后文件夹占用空间和文件个数 |
| xls | 65523行，2列 | 4236 | 3 | 5s | 30%左右  最高占用内存78MB | 4.91MB  86个文件 |
| xls | 11559行，7列 | 2080 | 3 | 7s~9s | 45%左右  最高占用内存66MB | 5.14MB  27个文件 |
| xls | 有数据的4020行，22列  （总行14412行） | 2776 | 3 | 5s | 45%左右  最高占用内存68MB | 7.79MB  40个文件 |
| xls | 6982行，32列 | 3398 | 3 | 9s | 45%左右  最高占用内存81MB | 15.9MB  32个文件 |
| ppt | 22页 | 638 | 3 | 3s | 32%~40%左右  最高占用内存93MB | 2.73MB  59个文件 |
| docx | 34页 | 473 | 3 | 5s~7s | 43%左右  最高占用内存161MB | 1.91MB  51个文件 |
| doc | 80页 | 1860 | 3 | 5s | 42%左右  最高占用内存136MB | 2.34MB  164个文件 |

## 7、对比

这里的pdf export也是Oracle中的中间件（第一点提到的下载地址中可以找到），同样也是转转大师所使用的中间件，已经测试过把转转大师的dll替换为pdf export的dll，同样执行转转大师中的PrintableDocumentToPdf.exe –p "out\_path" "in\_file\_path" 能转换为pdf文档。

由于测试的环境不一样，所以不作具体数据比较，这里大概说明一下他们之间的差异，以ABC作为评价标准，A优于B优于C。

|  |  |  |  |
| --- | --- | --- | --- |
|  | openoffice | pdf export | web view export |
| 转换时间 | C | B | A |
| 转换效果 | A  经过处理删除了一些空白的页 | B  经典，完整的文档转换 | -  不同文件类型无法比较 |
| 输出文件占用空间 | A | B | C |
| 预览效果 | B  使用flexpaper进行预览，预览excel的时候，列多的时候同一行的数据会显示在不同的页，所以预览某行的数据极其困难 | | A  预览excel能正常显示并能选择工作表 |