PharmaSUG China 2015 - Paper 28X-B9F7B4B9P8

# A Fully Automated Approach to Concatenate RTF outputs and Create TOC

Zhiping Yan, Covance, Beijing, China
Lugang Xie, Merck, Princeton, US

## ABSTRACT

Statistical reports for clinical studies usually contain huge amounts of tables, listings and figures. As ODS RTF has gained its popularity in clinical trial reporting, it is extremely useful to concatenate RTF reports and create a hyperlinked table of contents (TOC) to facilitate review. However, the approaches introduced by the existing papers have issues in one way or another. For instance, manual process must be involved – TOC field is updated by pressing F9; anchor ID must be inserted in the reports by each individual SAS programmer, etc.

This paper introduces a relatively simple and very efficient SAS approach to concatenate reports generated by ODS RTF and create a nicely formatted hyperlinked table of contents. It utilizes the bookmark/hyperlink features of RTF, allowing the generation of TOC to involve no manual process and hidden text. At the end, this paper also attempts to discuss on potential issues and future improvement.

## INTRODUCTION

Several methods can be used to combine RTF and create a TOC. However, there are no papers to demonstrate how figures can be combined with tables and listings. As for creation of TOC, each of the methods introduced by papers available today has issues like hidden text, manual work and extra code in individual program, etc.

With approach presented in this paper, we are able to combine figures with tables and listings with mixed orientations (landscape and portrait) and accommodate different types of RTF outputs, whether individual files with page numbers printed in the header section or in the body part. If the title is printed in the header section, users have the option to retain page numbers of each individual file or let Word paginate the entire combined document.

This approach allows SAS to extract the titles from the individual files and create TOC entries with imbedded hyperlinks. Therefore, no manual process is required for TOC to be displayed. It uses \bkmkstart and \bkmkend to determine the destinations of the hyperlinks and hence hidden text is no longer necessary.

It also allows us to have more control on how files are combined and how TOC is formatted and customized. In addition, not only can the font style, size and color of the TOC be customized, but also an enriched header can be added to the TOC page.

## APPROACH/METHOD

### 1. CONCATENATE TABLES

#### Makeup of RTF code of a SAS created RTF file

As far as this paper is concerned, an RTF file consists of an opening section, a content section and a closingsection.1) Opening section is the RTF code until the first \sectd of the file, which includes information of how the file is rendered by such viewing application as MS Word, etc., e.g. font styles and colors;2) Content section includes the content to be displayed. The RTF code for each page starts with \sect\sectd or \sectd; 3) Closing section is the RTF code after the last \pard of the file, typically only a closing curly brace. (Figure 1)

#### Concatenate RTF files

RTF files can be concatenated by the following steps (Figure 2):

1. Read in RTF files and output the RTF code to a SAS dataset.
2. Remove the closing section of the first file;
3. Remove the opening section of the last file;
4. Remove both the opening section and closing section of all other files;
5. Insert \sect in front of the first page of each RTF file except for the first one. The RTF code \sect starts a new page and \sectd sets the section properties to the default. Not surprisingly, the first section (first page) of an RTF file does not have \sect and always begins with \sectd while the rest sections always with \sect\sectd. In the

combined file, the first page of each individual file is no longer the first page, except for the first file.\sect therefore needs to be inserted in front of the first \sectd of those files.

6. Output above RTF code to one single file in such an order: the first file, the rest files and the last file.

| | | |
|---|---|---|
| {\rtf 1<br>font table group<br>Color table group<br>Stylesheet<br>Document information<br>Document formatting control words<br>..... | {\rtf 1<br>\fonttbl<br>\colortbl<br>\stylesheet<br>\info<br>\margl1440\margr1757<br>.... | **Opening section** |
| section 1<br><br><br><br><br>section 2<br><br><br><br><br>section 3<br><br><br><br>.... | \sectd<br>\pgnrestart\pgnstarts1<br>Page X of Y<br>{\upr{*\bkmkstart IDX}{*\ud{*\bkmkstart IDX}}}{*\bkmkend IDX}<br><br>\sect\sectd<br>\pgnrestart\pgnstarts1<br>Page X of Y<br>{\upr{*\bkmkstart IDX}{*\ud{*\bkmkstart IDX}}}{*\bkmkend IDX}<br><br>\sect\sectd<br>\pgnrestart\pgnstarts1<br>Page X of Y<br>{\upr{*\bkmkstart IDX}{*\ud{*\bkmkstart IDX}}}{*\bkmkend IDX}<br>.... | **Content section** |
| } | } | **Closing section** |

**Figure 1. Makeup of RTF code of a SAS created RTF file**

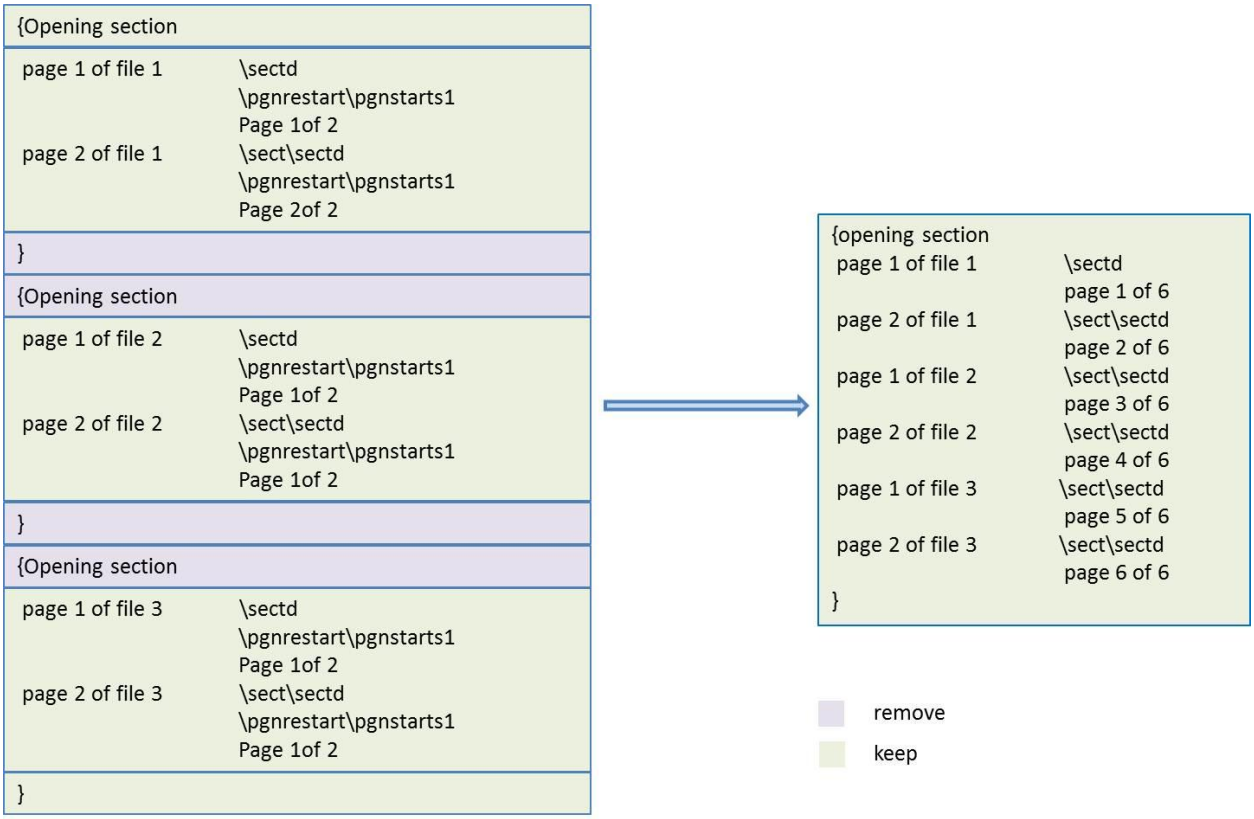


**Figure 2. Concatenate RTF files**

## 2. CREATE TOC

### Extract titles of individual files

To extract the title of the file, SAS needs to locate where the title starts and ends. Fortunately, the RTF code surrounding the title of each file will be identical if all files are produced with the same ODS style template, which makes extraction an easier task.

Here is an example of RTF code surrounding the title of a table:

```
}}{\upr{\*\bkmkstart IDX}{\*\ud{\*\bkmkstart IDX}}}{\*\bkmkend IDX}\trowd\trkeep\trhdr\trqc
\cltxlrtb\clvertalb\clcbpat8\clpadt10\clpadft3\clpadr10\clpadfr3\cellx12642
\pard\plain\intbl\keepn\sb10\sa10\ql\f1\fs18\cf1{\ql{Protocol\~No:\~XXXXXX}\ptablnone\pmartabqr{Page\~1\~of\~1}\par{\line}
\ql{Table\~14.1.1\~\~Summary\~of\~Subject\~Disposition\~All\~Randomized\~Population}\par\brdrb\brdrs\brdrw20\cell}
{\row}
```

**Figure 3. An example of RTF code surrounding the title of a table**

The title in this example is preceded by \pard\plain\intbl\keepn\sb10\sa10\ql\f1\fs18\cf1{{ and followed by \cell}. This is also the case for all other files as long as they are produced with the same ODS style template. In other words, the title is the text between \pard\plain\intbl\keepn\sb10\sa10\ql\f1\fs18\cf1{{ and \cell}, with \~ being replaced with a blank and other RTF control words (\par, \brdrb, etc.) removed.

### Calculate page numbers to be displayed in the TOC

Now that we know a page always starts with \sect\sectd, the positions of the first \sect\sectd of the individual files in the combined file are the page numbers to be displayed in the TOC. The following figure illustrates how pages numbers are calculated for the TOC.

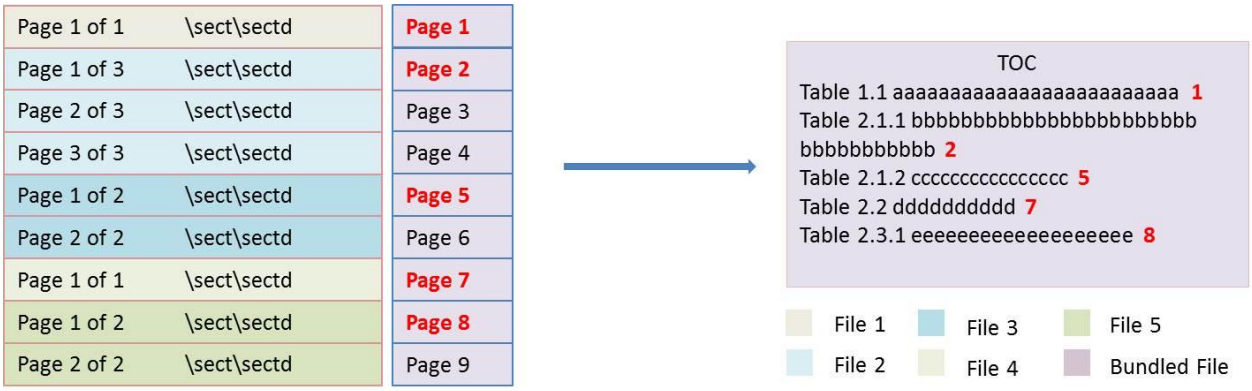


**Figure 4. Calculate page numbers to be displayed in the TOC**

### Link each TOC entry to the corresponding TFL

The following two steps will link the TOC entries to their corresponding TFL outputs in the combined file:

1. Adding the following RTF code to the TOC page for each output to create hyperlinks.

   ```
   {\field {\*\fldinst HYPERLINK \\l "bookmark_name"}{\fldrslt title_of_RTF_output}}
   ```

   title_of_RTF_output is the title extracted at the earlier steps and bookmark_name must be a unique bookmark name for each output. bkidx1, bkidx2, bkidx3…are used in the examples of this paper.

2. Embedding the RTF code at the beginning of the first page of each corresponding individual output to mark the destinations of the hyperlinks.

   ```
   {\upr{\*\bkmkstart bookmark_name }{\*\bkmkend bookmark_name }
   ```

   The bookmark_name here must be the same as the bookmark_name for its corresponding TOC entry.

SAS procedures such as proc report and proc gplot automatically create RTF codes for bookmarks. We can take advantage of that to make implementation easier. SAS procedure created bookmark code for the first page always looks like:

```
{\upr{\*\bkmkstart IDX#}{\*\ud{\*\bkmkstart IDX#}}}{\*\bkmkend IDX#}
```

IDX# can be replaced with bkidx1or bkidx2 or bkidx3, etc.

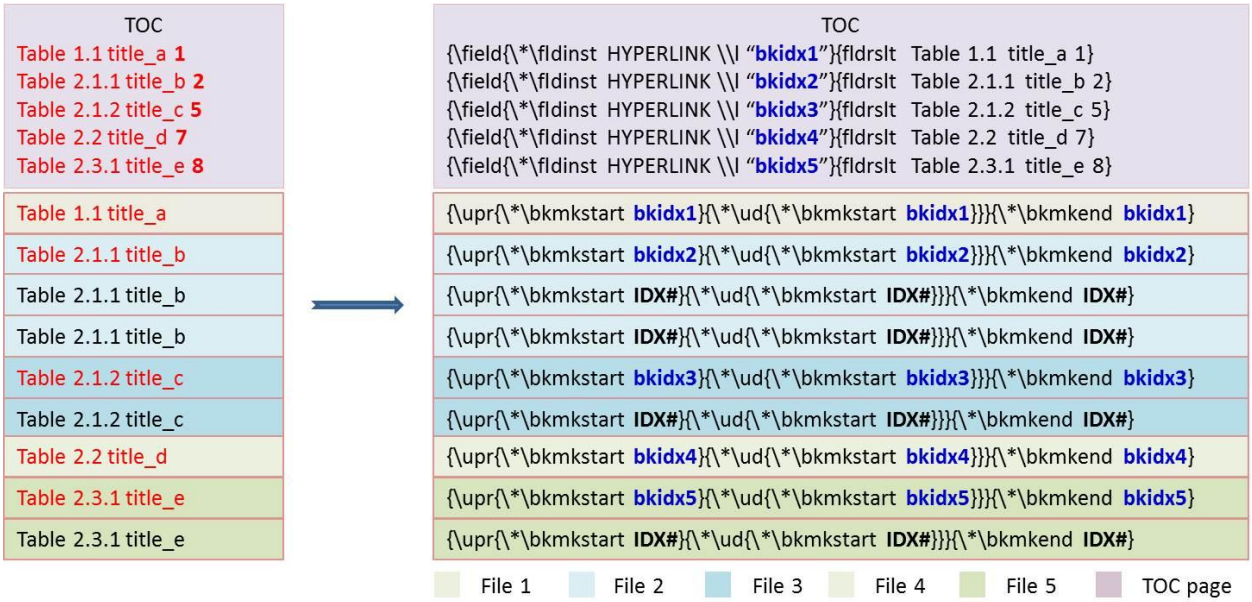


**Figure 5. Link each TOC entry to its corresponding TFL**

## 3. FORMAT TOC

### Adding readability features

At this point, the TOC that we have created looks like Figure6a.Further work is required to improve its readability. Following steps and Figure 6 demonstrate how Figure6a can be formatted into Figure6d.

1. Inserting hanging indents if a title is too long to fit in one line in the TOC (Figure6b).
2. Left aligning the titles (Figure6c).
3. Right aligning page numbers and inserting dotted lines between titles and page numbers (Figure6d).

### Enable TOC to break across pages automatically

It is easier to control the formatting of TOC in a single cell. Figure 7 shows RTF code producing TOC in Figure 6d. By default, \trkeep is usually added when constructing a table row. Removing \trkeep enables TOC to break across pages. Besides, font style, font size and font color can be customized by changing the control words in green in Figure 7.

## 4. APPEND TOC PAGE WITH PREVIOUS CONCATENATED RTF FILE

The same approach illustrated in Figure 2 can be used here to append the combined RTF to the TOC page (Figure 8a). There is no need to remove closing section of TOC page since it consists of only content section. In addition, an opening section is required. There are two ways to provide an opening section. One is manually creating an opening section for the TOC page, and the other taking advantage of existing opening section of concatenated file. To make the TOC page more informative, study title and data cutoff date can be added to the TOC page. An example is showed in Figure 8b.

## LIMITATIONS

Figure titles must be printed outside of the graph since the titles to be displayed in TOC are extracted from RTF files.

MS Word cannot open an RTF file of over 50MB. If a combined file is over 50MB, the VBA approach can be an alternative choice.

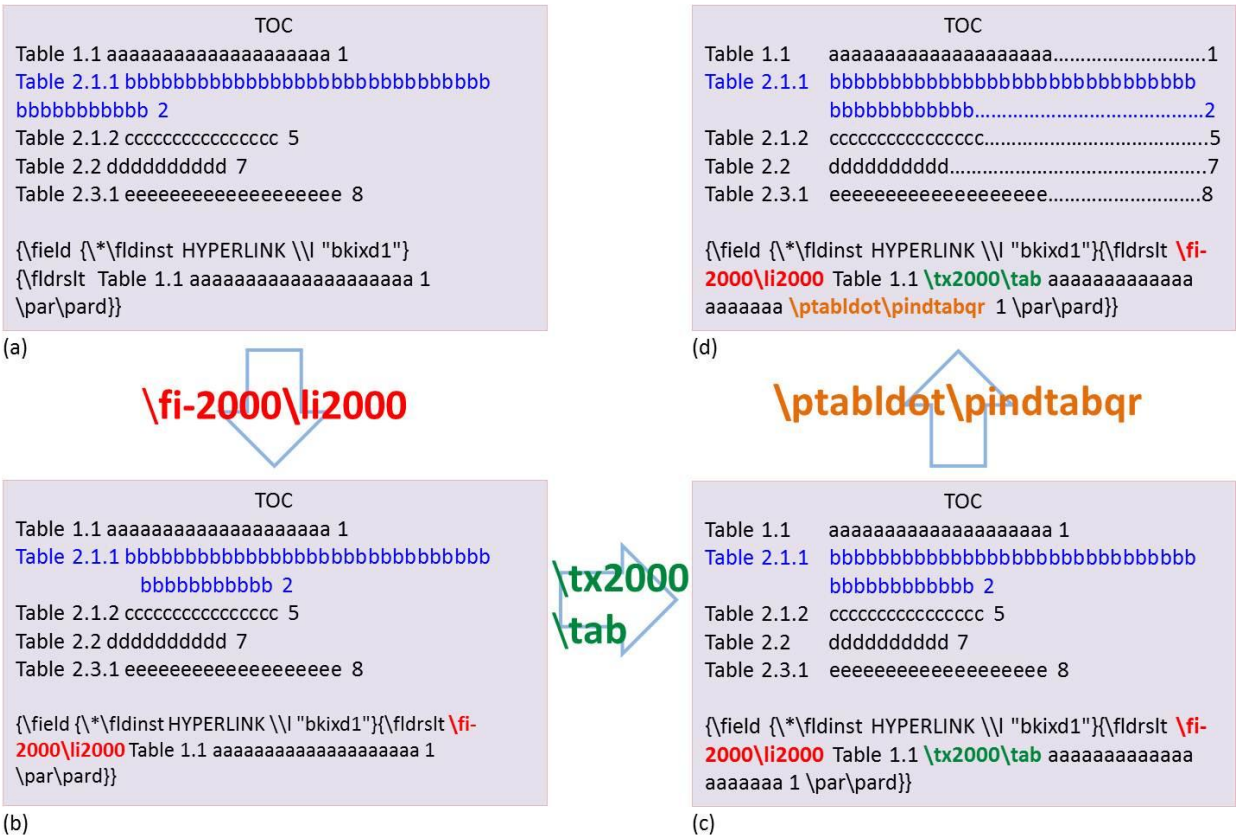


**Figure 6. Align TOC**

```
\trowd\trqc\trkeep\cellx9000
\pard\plain\intbl\ql\f1\fs20\cf1{

{\field {\*\fldinst HYPERLINK \\l "bkixd1"}{\fldrslt \fi-2000\li2000  Table 1.1 \tx2000\tab aaaaaaaaaaaaaaaaaaa
\ptabldot\pindtabqr 1 \par\pard}}

{\field {\*\fldinst HYPERLINK \\l "bkixd1"}{\fldrslt \fi-2000\li2000  Table 2.1.1 \tx2000\tab bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbb  \ptabldot\pindtabqr 2 \par\pard}}

{\field {\*\fldinst HYPERLINK \\l "bkixd1"}{\fldrslt \fi-2000\li2000  Table 2.1.2 \tx2000\tab ccccccccccccccc \ptabldot\pindtabqr 5
\par\pard}}

{\field {\*\fldinst HYPERLINK \\l "bkixd1"}{\fldrslt \fi-2000\li2000  Table 2.2 \tx2000\tab ddddddddddd \ptabldot\pindtabqr 7
\par\pard}}

{\field {\*\fldinst HYPERLINK \\l "bkixd1"}{\fldrslt \fi-2000\li2000  Table 2.3.1 \tx2000\tab eeeeeeeeeeeeeeeeeee
\ptabldot\pindtabqr 8 \par\pard}}

}
\cell
\row
```

**Figure 7. Further format TOC**

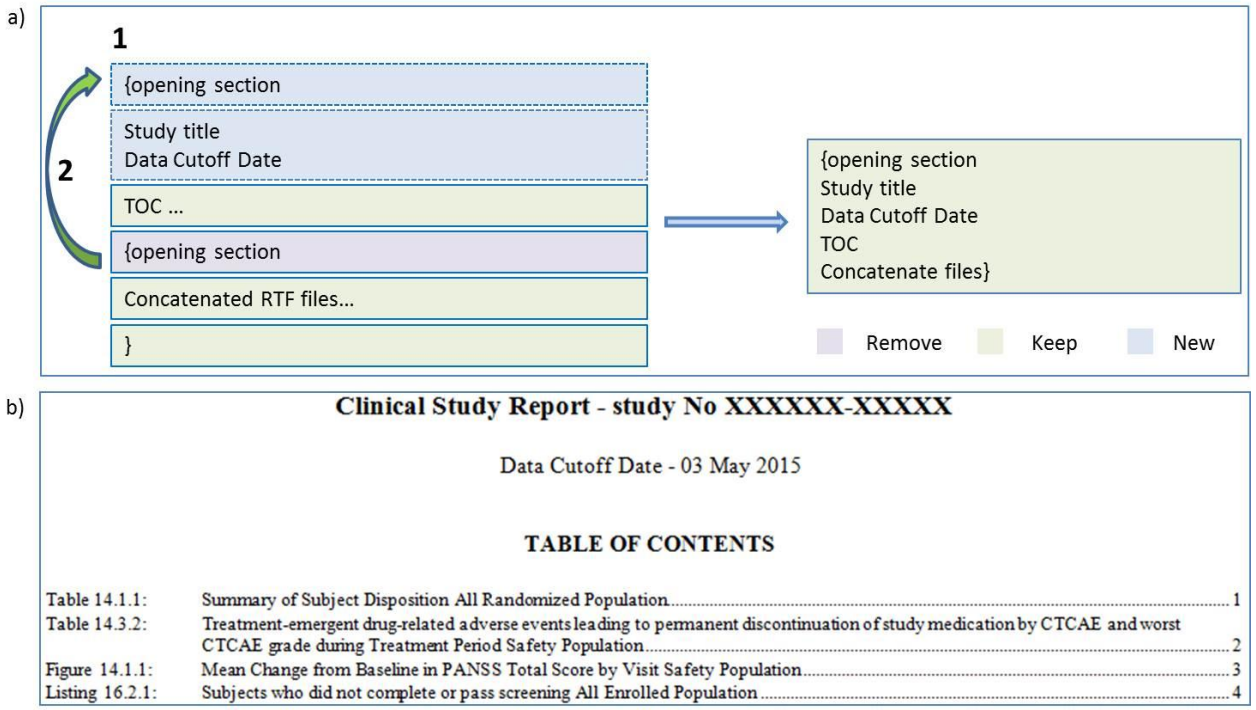


**Figure 8. Two ways to append TOC page with previous concatenated RTF files**

## FUTURE IMPROVEMENT

If two or more special colors other than the default 16 ones are used, further work such as renumbering color table entries is required. This is also true for the case where multiple font styles are applied.

## CONCLUSION

This paper presents a method to combine RTF files and create a hyperlinked table of contents using a fully automated process implemented by a single SAS program. By utilizing the hyperlink/bookmark features of RTF, this approach completely eliminates the hidden text issue and manual processes to update the TOC. It has the ability to combine figures and tables with very limited requirements for the way individual RTF files are created. It is a CPU efficient approach and can be executed in batch together with other programs.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Zhiping Yan
Work Phone: +86 13691423469
E-mail: yanzhiping1987@hotmail.com

Name: Lugang Xie
Work Phone: +001 7325108345
E-mail: larryxie@gmail.com