

## **Comparing ODS RTF Output Files in Batch Using SAS (or, 500 ODS RTF Documents to Compare in 15 minutes!)**

David Franklin, Litchfield, NH

### **ABSTRACT**

Before ODS came along most tables and listings generated from a SAS program were text files that were sometimes converted into RTF files using Microsoft VBA macros or other such techniques. Comparing the text files between different versions of the output was easy as it was generally just comparing one text file against another text file. Then came ODS RTF and suddenly the tables and listings were near publication ready but also a huge headache developed when it came to comparing multiple documents in one version of the output versus another version. This paper looks at a technique that was born to out of a need to compare over five hundred ODS RTF generated output tables and listings covering over six thousand pages against a previous version and finding only one number that had changed!

### **INTRODUCTION**

Comparing output documents generated in SAS® has always been a necessary chore. Before ODS RTF arrived comparing the output was a relatively easy task as the output files were just text files that were usually transformed into RTF files using VBA macros or other text to RTF conversion tools. The compare itself could be done using simple tools like the DOS FC command. Then ODS RTF came and the output that this new feature unleashed was almost breathtaking to those who received the output but when it came to compare the output from a version to a previous version, new tools had to be found. Some converted the two versions of the RTF documents into Text files and used their old compare file strategies. Others compared files one by one manually using document compare tools available in most word processors like Microsoft Word.

This paper looks at one approach that was taken using SAS and VBA within Microsoft Word to compare all ODS RTF output files in a directory against a previous version of the files in another directory. The idea for this approach was when over five-hundred RTF generated files needed to be compared quickly to make sure that only a single listing had altered due to a database change that involved amending only one data value in one table.

### **OVERVIEW**

In order to compare so many RTF output files quickly it was realized early on that a portable SAS macro was needed to:

- a) get the list RTF files in the Base directory and compare that list against the files in the COMPARE directory, keeping only those files that are common (the two directory parameters could not be the same), and
- b) on this common list do a compare of the RTF files using VBA and generate a report that showed if there were any differences, and if so, what these are.

The resulting report, it was decided, would be just a text file for ease of review and programming considerations.

The SAS macro itself only needed the following parameters:

- location of BASE directory
- location of COMPARE directory
- location and file name of report file

The actual compare of RTF files would be done inside a Microsoft VBA macro that would be called by SAS.

### **GETTING THE LIST OF RTF FILES TO COMPARE**

A list of the files to compare was constructed using a call system command and reading the file into SAS datasets. The piping technique could have been used but this was found to be problematic due to the technical structure of the network.

```

data _null_;
  call command("dir &basedir\*.rtf /b /l > c:\_basedir.txt");
  call command("dir &compdir\*.rtf /b /l > c:\_compdir.txt");
run;
data _basedir;
  length fn $50;
  infile "c:\_basedir.txt" length=len;
  input fn $;
run;
data _compdir;
  length fn $50;
  infile "c:\_compdir.txt" length=len;
  input fn $;
run;
data complst nomatch;
  merge _basedir (in=a) _compdir (in=b);
  by fn;
  if a and b then output comlst; *Match found;
  else output nomatch; *No match found;
run;

```

If there were no file matches found between the two directories then the macro would stop and no compare is done. However if matches were found then a list of the files to compare would be generated, and it would be this file that the VBA macro would use as input for the list of files to compare. The file was generated using the following SAS code:

```

data _null_;
  length txt $256;
  file "c:\_complst.txt";
  set complst;
  txt=trim(left("&basedir"))||"\\"||
    trim(left(fn))||'\~'||
    trim(left("&compdir"))||"\\"||
    trim(left(fn));
run;

```

Refer to Appendix A for an example of a file and contains the list of the files that are going to be compared.

## AND NOW FOR THE VBA

As was noted earlier the actual compare of RTF files would be done inside a Microsoft VBA macro that would be called by SAS. VBA was chosen since Microsoft Word had a very good way of comparing RTF files and generating a report on the results.

The call to action the Word VBA macro within the SAS environment was handled with the following call using an X statement:

```
"WinWordDir\WINWORD.EXE" /mCompareDocs "fIn"
```

where *WinWordDir* is the directory where the Microsoft Word Application resides, and *fIn* is the name of the file created from the code generated from SAS Code B.

The WINWORD.EXE call makes a call starts Microsoft Word and begins a VBA macro CompareDocs call with the parameter containing the files to compare. It is most important that the VBA macro CompareDocs must reside inside the Word Normal.dot template1. Refer to Appendix B for the source code of the VBA macro CompareDocs.

## THE COMPARE RESULTS DOCUMENT

The VBA macro will output the differences found to a single file called DD\_CompareResults.txt. An example of the output is in Appendix C. Note that the compare of each set of documents include any differences in the Header and Footer - in the case of the 034 project the dates and times of the documents were different in many cases but what the compare is looking for is content changes. If a table is long then tracking down a comparison can be difficult in

that table. If no differences were found between the two documents a message indicating that they are identical is generated to the DD\_CompareResult.txt document.

## **TWO (VERY NICE) UNEXPECTED SIDE EFFECTS**

After using the macro to compare RTF documents in batch it was decided that it would be an interesting test to determine if the macro would work against a directory that contained RTF format files against another directory that contained Word format files. The test was constructed the same way as shown in the above examples except this time the Base file was filename.DOC and the Compare file was compare.RTF and the program worked as expected. A later use of the macro was to compare SAS programs in two directories with the same successful result.

## **CONCLUSION**

ODS RTF created a new and very exciting way for programmers to generate output for their customers that the customers welcomed. Along with this was the need to develop new tools including one that could compare RTF output files on a large scale. The macro presented here using SAS and RTF filled that need and went even further in being able to compare other types of file.

## **CONTACT INFORMATION**

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

David Franklin  
16 Roberts Road  
Litchfield, NH 03052  
Tel/Fax 603/216-2232  
Email 100316.3451@compuserve.com  
<http://ourworld.compuserve.com/homepages/dfranklinuk>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

## APPENDIX A – EXAMPLE OF LIST OF FILES TO COMPARE

C:\CLINDATA\TUAI\SH-38\Tables\DEV\T1-DISP.rtf~C:\CLINDATA\TUAI\SH-38\Tables\ReleaseA\T1-DISP.rtf  
C:\CLINDATA\TUAI\SH-38\Tables\DEV\T2-DEMO.rtf~C:\CLINDATA\TUAI\SH-38\Tables\ReleaseA\T2-DEMO.rtf  
C:\CLINDATA\TUAI\SH-38\Tables\DEV\T3-AE.rtf~C:\CLINDATA\TUAI\SH-38\Tables\ReleaseA\T3-AE.rtf

## APPENDIX B - VBA MACRO TO COMPARE DOCUMENTS

```
Sub CompareDocs()

    Dim DocA As Document
    Dim DocB As String
    Dim DocC As Document
    Dim directory As String
    Dim outdoc As String
    Dim arev As Revision
    Dim ReportDoc As Document
    Dim i As Long
    Dim k As Long
    Dim LogFile As Variant
    Dim FileClicked As String
    Dim FileList As String
    Dim filelistbat As Boolean
    Dim lastslash As Long
    Dim nFile As Integer
    Dim mFile As Integer
    FileClicked = ActiveDocument.FullName
    FileList = FileClicked
    If Len(FileClicked) > 0 Then
        filelistbat = 1
    End If
    i = 0
    lastslash = 0
    Do While i < Len(FileClicked)
        i = i + 1
        If Mid(FileClicked, i, 1) = "\" Then lastslash = i
    Loop
    directory = Mid(FileClicked, 1, lastslash)
    outdoc = directory + "DD_CompareResults.txt"
    nFile = FreeFile
    Open outdoc For Output As #nFile
    Close #nFile
    Open outdoc For Append As #nFile
    Print #nFile, "Document Compare - " + Format$(Now, "mm-dd-yy hh:mm")
    Print #nFile, "Base Directory: " + directory
    Application.ScreenUpdating = False
    mFile = FreeFile
    Open FileList For Input Access Read As #mFile
    If filelistbat Then
        While Not EOF(mFile)
            Input #mFile, LogFile
            LogFile = Trim$(UCase$(LogFile))
            LogFile = Replace(LogFile, Chr$(9), " ")
            Set DocA = Documents.Open(Mid$(LogFile, 1, InStr(1, LogFile, "~") - 1))
            DocB = Mid$(LogFile, InStr(1, LogFile, "~") + 1, Len(LogFile) - InStr(1, LogFile, "~") + 1)
            Print #nFile, "======"
            Print #nFile, "==" Base File: " + DocA
            Print #nFile, "==" Compare File: " + DocB
            Print #nFile, " "
            DocA.Compare DocB, CompareTarget:=wdCompareTargetNew, DetectFormatChanges:=False
            Set DocC = ActiveDocument
            k = 0
            Set ReportDoc = Documents.Add
            If DocC.Revisions.count > 0 Then
                For Each arev In DocC.Revisions
                    If arev.Type = wdRevisionDelete Then
                        Print #nFile, "*** REVISION ***"
                    End If
                Next arev
            End If
        Wend
    End If
End Sub
```

```

        Print #nFile, Chr(34) & arev.Range & Chr(34) & " was deleted from " & DocB & vbCr
    ElseIf arev.Type = wdRevisionInsert Then
        Print #nFile, "*** REVISION ***"
        Print #nFile, Chr(34) & arev.Range & Chr(34) & " was added to " & DocA.Name & vbCr
    End If
Next arev
k = k + 1
Else
    k = k
End If
If k = 0 Then
    Print #nFile, "The two files are identical."
End If
ReportDoc.Close saveChanges:=wdDoNotSaveChanges
DocA.Close saveChanges:=wdDoNotSaveChanges
ActiveWindow.Close saveChanges:=wdDoNotSaveChanges
Wend
End If
Print #nFile, " "
Print #nFile, "*** END OF REPORT ***"
Close #mFile
Close #nFile
ActiveWindow.Close saveChanges:=wdDoNotSaveChanges
Application.quit
End Sub

```

## APPENDIX C - EXAMPLE OF THE COMPARE RESULTS DOCUMENT

Document Compare - 01-12-05 11:09

Base Directory: C:\CLINDATA\TUA\SH-38\Tables\DEV

=====

== Base File: T1-DISP.RTF

== Compare File: C:\CLINDATA\TUA\SH-38\Tables\ReleaseA\T1-DISP.RTF

The two files are identical.

=====

== Base File: T2-DEMO.RTF

== Compare File: C:\CLINDATA\TUA\SH-38\Tables\ReleaseA\T2-DEMO.RTF

\*\* REVISION \*\*

"1 (6" was deleted from C:\CLINDATA\TUA\SH-38\Tables\ReleaseA\T2-DEMO.RTF

\*\* REVISION \*\*

"2 (13" was added to T2-DEMO.RTF

=====

== Base File: T3-AE.RTF

== Compare File: C:\CLINDATA\TUA\SH-38\Tables\ReleaseA\T3-AE.RTF

The two files are identical.

\*\* END OF REPORT \*\*