

## Obtaining an Automated Summary of PROC COMPARE Results: &SYSINFO and VB Script

Umesh Gautam, Trial Runners; Jeff Roberts, Chiltern

### ABSTRACT

In the life cycle of a clinical analysis project, it's often necessary to do complete validation of datasets and outputs multiple times as datasets and outputs are created more than once for draft, dry run, data cuts, etc. During program development there's no substitute for solid program construction and close attention to the comparison results of double programming validation. However once robust production and validation programs are in place, the validation of programs and outputs against new data may be obtained much more efficiently with automated comparison and summary of the results. This paper will show how such a process may be implemented using SAS® automatic variables and VBScript.

### INTRODUCTION

Double programming is the industry standard for program validation in clinical trials analysis. PROC COMPARE is the best means of validating datasets. Tables, figures, and listings may also be validated using this procedure when a clear process is developed for creating persistent output datasets that accurately represent the underlying data for the visual artifacts.

The results of the PROC COMPARE are stored at runtime in the &SYSINFO macro variable generated by SAS® automatically. Treating this as a Boolean result, with 0 as the success value permits a pass/fail summary of the comparison. When this pass/fail result is captured and sent to the LST file output generated by the procedure we can then iterate through all the LST files in a folder and generate a validation summary report using VB Script. This represents tremendous time savings when repeat validations are needed for new data. We show a Windows-based solution but the logic could be transferred to a UNIX environment as well.

### USING SYSINFO TO SUMMARIZE THE COMPARISON RESULT

The results of PROC COMPARE are multifaceted, and completely understanding these nuances of comparison results is essential for competent validation. Addressing the many ways in which two datasets may differ is not discussed here, but has been the subject of many papers. What we want to focus on here is the Boolean evaluation, &SYSINFO. Quite simply, if &SYSINFO=0 then the comparison is considered as "QC Pass", and any other value is evaluated as "QC Fail".

The various permutations of mismatch could be captured and put into the summary, but the practical reality is that anything other than a 100% match is going to require going into an interactive SAS® session and looking at the specific differences and engaging in the process of resolving them, at which point we've moved past the summary review stage anyway, so there's little to be gained by presenting more detail in the summary results.

What we want to do is take this pass/fail evaluation of the results, and put it into the LST output as a title with either a STATUS: QCPASS or STATUS: QCFAIL value. We'll then use this line in our summary output VB Script routine.

There's a tricky part to this, though. We want to put the summary results of the comparison into the title of the LST file. But the title statement needs to precede the procedure code. By the time the procedure has run, and we've evaluated the results then the title has already been rendered.

#### Here's what we do:

1. Run PROC COMPARE once with the NOPRINT option applied

2. Capture and evaluate the results in &SYSINFO as either STATUS: QCPASS or STATUS: QCFAIL
3. Put this into a TITLE2 statement
4. Run the same PROC COMPARE once again without the NOPRINT option so that all details of the comparison are sent to the LST file

**Here's the code that accomplishes this:**

\*Note that we use the `_ALL_` operator so that there can be no doubt as to whether every variable in the production dataset has been compared or not. Comparisons that need to use the `CRITERION` argument to handle floating point values would require additional parameters or a different version of the macro.

```
%macro output_comp_return (dst= , base= , comp= , idvars= );

  proc compare noprint base = <base> comp =<comp>;
    id &idvars.;
    var _all_;
  run;

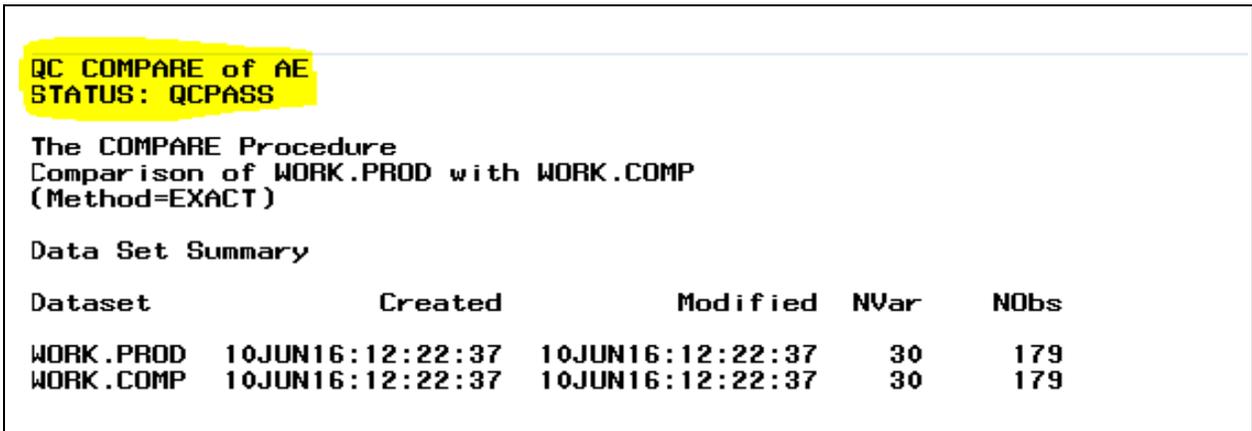
  if %eval(&sysinfo) = 0 then do;
    %let status= QCPASS;
  end;
  else do;
    %let status = QCFAIL;
  end;

  title1 "QC COMPARE of &dst.";
  title2 "STATUS: &status.";

  proc compare base = &base. comp = &comp. listall ;
    id &idvars.;
    var _all_;
  run;

%mend output_comp_return;
```

**Here's how the LST file will appear:**



The screenshot shows the output of a PROC COMPARE procedure. The first two lines are highlighted in yellow: "QC COMPARE of AE" and "STATUS: QCPASS". Below this, the procedure title is "The COMPARE Procedure Comparison of WORK.PROD with WORK.COMP (Method=EXACT)". A "Data Set Summary" table follows, comparing two datasets: WORK.PROD and WORK.COMP. Both datasets have 30 variables and 179 observations, and were created and modified at the same time (10JUN16:12:22:37).

Dataset	Created	Modified	NVar	NObs
WORK.PROD	10JUN16:12:22:37	10JUN16:12:22:37	30	179
WORK.COMP	10JUN16:12:22:37	10JUN16:12:22:37	30	179

**Output 1. LST output with status on the header**

## VBSCRIPT TO PARSE THE SYSINFO VALUE

Now have a macro that produces LST files with the QC status in the TITLE2 level of the output. After running all of our validation programs in batch mode, we have LST files for each of them in one folder. The next step is to use a VB Script routine that will then parse every LST file found in the target folder looking for the dataset being compared and the QC PASS or QCFAIL keywords.

### Here's what we do:

1. Store the generic summarize\_lst\_results.vbs script file in the LST output folder
2. Get runtime reference to the current folder
3. Iterate through the folder and evaluate every file found with .LST extension
4. Open a text stream for each LST file and do keyword searches for the dataset name and the QC results
5. Keep a running count of total number of LST files review and the count of the QC status results for a summary.
6. Send these results to a single output text file that is named with a timestamp

### Here's the code that accomplishes this:

```
Dim fso, strCurPath, ofolder, ofiles, txtFileList, folderIdx
Dim curlogfile, strBinary, lngNumber1, lngNumber2, strDigit

Set fso = CreateObject("Scripting.FileSystemObject")

strCurPath = fso.GetAbsolutePathName(".")

curdate = formatnumber(Day(Date),0,-1) & "-" &
UCase(Left(MonthName(Month(Date)), 3)) & "-" & Year(Date) _
    & "-" & formatnumber(hour(time),0,-1) & "-" &
formatnumber(minute(time),0,-1) )

Set txtFileList = fso.CreateTextFile(strCurPath & "\PROC Compare Review" &
curdate & ".txt")
Set ofolder = fso.GetFolder(strCurPath)
Set ofiles = ofolder.Files

i=0
pass=0
fail=0

For Each folderIdx In ofiles

If LCase(Right(folderIdx.Name, 3)) = "lst" Then
i=i+1
txtFileList.WriteLine (" ")
txtFileList.WriteLine (i & "." & "Review of: " & ucase(folderIdx.Name))
txtFileList.WriteLine (" ")

Set curlogfile = fso.opentextfile(folderIdx.Name, 1)

Do While curlogfile.AtEndOfStream <> True
Line = curlogfile.ReadLine

If InStr(LCase(Line), "qc compare of") then
txtFileList.WriteLine (Line)
End If
```

```
    If InStr(LCase(Line), "qcpass") Then
        txtFileList.WriteLine ( Line )
        pass=pass+1
    End If

    If InStr(LCase(Line), "qcfail") Then
        txtFileList.WriteLine ( Line )
        fail=fail+1
    End If
Loop

txtFileList.WriteLine ( " " )

txtFileList.WriteLine("*****")

    curlogfile.Close

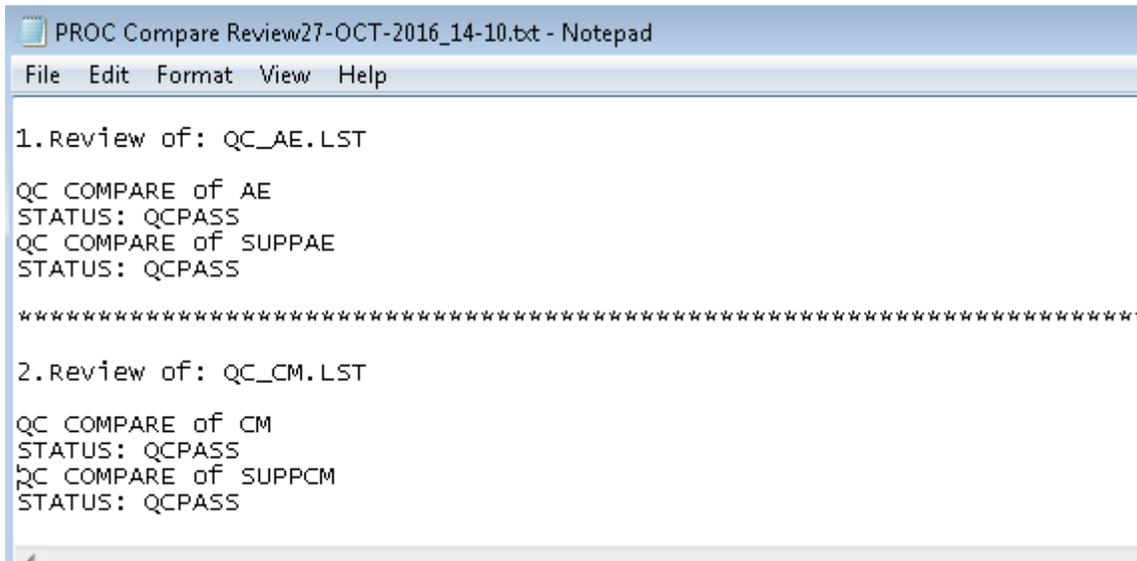
End If
Next

txtFileList.WriteLine(" ")
txtFileList.WriteLine("*****")
txtFileList.WriteLine("*****SUMMARY *****")
txtFileList.WriteLine("*****")
txtFileList.WriteLine ("Total numbers of lst file read: " & i)
txtFileList.WriteLine ("QC PASS:" & pass)
txtFileList.WriteLine ("QC FAIL:" & fail)

txtFileList.Close
```

### Here's what we get:

A summary result for each dataset



```
PROC Compare Review27-OCT-2016_14-10.txt - Notepad
File Edit Format View Help

1.Review of: QC_AE.LST

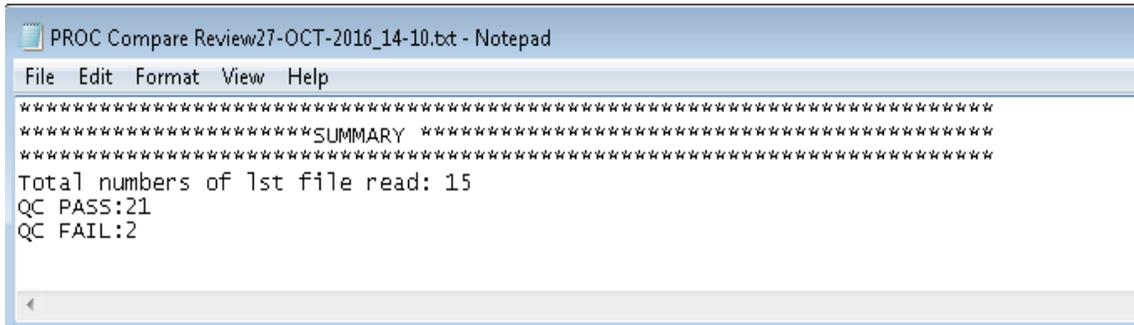
QC COMPARE of AE
STATUS: QCPASS
QC COMPARE of SUPPAE
STATUS: QCPASS

*****

2.Review of: QC_CM.LST

QC COMPARE of CM
STATUS: QCPASS
QC COMPARE of SUPPCM
STATUS: QCPASS
```

An overall summary at the end of the text file



```
*****  
*****SUMMARY*****  
*****  
Total numbers of 1st file read: 15  
QC PASS:21  
QC FAIL:2
```

## Output 2. Summary of the LST output

When any of the programs is QC fail it's time to open the specific LST file and review the discrepancies found.

## CONCLUSION

There's no substitute for well-constructed validation programs and careful review of the first PROC COMPARE results done in the QC process. However, once this is done we can automate the subsequent reviews that are done against new data in a much more systematic and economical process.

## CONTACT INFORMATION

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

Umesh Gautam  
Trial Runners, LLC  
116 Villard St W  
City, State ZIP: Dickinson, ND, 58601  
E-mail: [umeshgautam@trialrunners.com](mailto:umeshgautam@trialrunners.com)  
Web: [www.trialrunners.com](http://www.trialrunners.com)

Jeff Roberts  
Chiltern  
City, State ZIP: King of Prussia, PA, 19406  
E-mail: [jeff.roberts@chiltern.com](mailto:jeff.roberts@chiltern.com)  
Web: [www.chiltern.com](http://www.chiltern.com)

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.