

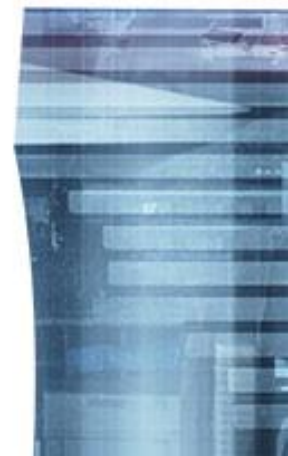
Business & Decision

Life Sciences

Fingerprinting
your TFL outputs
to monitor changes
between batch runs

Pamphile Beaujean

CT11 – PHUSE Virtual EU Connect 2021



Business & Decision AGENDA Life Sciences

- Introduction
- Proposition and context
- Key code bits
- Results
- Limitations/Alternatives
- Conclusion



#Data #DataDriven
#DataStrategy #DataIntegration
#DataViz #DataStandards
#DataScience #Datagovernance
#HealthCare #DataAnalysis
#ClinicalTrials
#ArtificialIntelligence
#AdvancedAnalytics
#MachineLearning #Digital
#Chatbot #Biometrics #CRO
#ClinicalData
#LifeSciencesExperts
#Whodrug #MedDRA
#Pharmacovigilance
#ClinicalResearchLifeCycle
#MedicalWriting
#MedicalInformation
#MedicalReview #SDTM
#ADaM #CDISC #Innovation
#RegulatorySubmissions
#Protocol #Coding

Business & Decision

INTRODUCTION

Life Sciences

Automatic comparison of TFL outputs is:

- Required to ensure outputs were not modified
- More than meets the eye (timestamps)
- An old problem that has been tackled using
 - SAS Macros
 - Microsoft VBA
 - Proc compare

Business & Decision PROPOSITION

Life Sciences

Goal: Identify outputs that changed between two versions

1. Start from deliverables
2. Generate copies of deliverables with neutral dates
3. Synthesize file content with fingerprints (aka checksums)
4. Add TFL outputs comparison to other SAS QC checks



FileInfos SAS macro was developed

Business & Decision

Life Sciences

- Requirements
 - Windows powershell
 - Outputs produced by ODS rtf
 - x commands allowed
- Nice to have
 - Automated structure, titles and footnotes

Business & Decision

Life Sciences

- Standard used everyday to check file integrity
- Nice properties overall
 - ✓ Fast execution
 - ✓ Dimensional reduction: input space $2^{(2^{64}-1)}$ to output space 2^{256} for SHA-256 algorithm
 - ✗ Collisions happen in theory

```
data _null_;  
    original=sha256('I am the original');  
    modified=sha256('I am the original');  
    put original=$hex64.; put modified=$hex64.;  
run;
```

```
original=B176665A760F276474B2FF3D067D47BE72D5EA4046BF4EA8E134046B5F46758E  
|  
modified=809F2FF98797109C456251BA5B8A0B65EB02B603D7A6E7BEC71D3B70DDC0409C
```

Avalanche Effect

Business & Decision

Life Sciences

```
KEY CODE BITS – GENERAL OUTLINE

%macro FileInfo (refdat=, scope=, tdir=, fdir=, ldir=, outdir=, tmpdir=, clean=);
    %init (&tmpdir);                *- Create tmp folder and set environment -*
    %copyTFLs (tdir=&tdir, fdir=&fdir, ldir=&ldir, outdir=&tmpdir);
    %listfiles (&tmpdir);
    %modTFLruntimes (TFLs=&list);    *- Replace timestamps using regex -*
    %TFLinfos (in=&tmpdir, out=&outdir); *- Collects fingerprints and other info -*

    %if &clean eq Y %then %do;
        %clean;                    *- Delete tmp folder and clean environment -*
    %end;

%mend FileInfo;
```

Business & Decision

Life Sciences

```
KEY CODE BITS – GENERAL OUTLINE

%macro FileInfo(refdat=, scope=, tdir=, fdir=, ldir=, outdir=, tmpdir=, clean=);
    %init(&tmpdir);                *- Create tmp folder and set environment -*
    %copyTFLs(tdir=&tdir, fdir=&fdir, ldir=&ldir, outdir=&tmpdir);
    %listfiles(&tmpdir);
    %modTFLruntimes(TFLs=&list);    *- Replace timestamps using regex -*
    %TFLinfos(in=&tmpdir, out=&outdir); *- Collects fingerprints and other info -*

    %if &clean eq Y %then %do;
        %clean;                    *- Delete tmp folder and clean environment -*
    %end;

%mend FileInfo;
```

Business & Decision

KEY CODE BITS – WRITE POWERSHELL SCRIPT

Life Sciences

```
%macro modTFLruntimes(TFLs=);  
    data _null_;  
        file "&tmpdir.\chgdt.ps1";  
        %let fregex=%sysfunc(prxparse(/F.*/io));    *- To identify figure outputs -*  
        %let nb_TFLs=%sysfunc(countw(&TFLs));  
        %do i = 1 %to &nb_TFLs;  
            %let TFL = %scan(&TFLs, &i);  
            put "$file = '&tmpdir.\&TFL..rtf'"; *- Define file path in ps1 script -*  
            %regex; *- Macro with powershell regex instructions to be written -*  
        %end;  
    run;  
    x powershell -file "&tmpdir.\chgdt.ps1";    *- Execute script -*  
%mend modTFLruntimes;
```

Business & Decision

Life Sciences

KEY CODE BITS – WRITE POWERSHELL SCRIPT

```
%macro modTFLruntimes(TFLs=);  
    data _null_;  
        file "&tmpdir.\chgdt.ps1";  
        %let fregex=%sysfunc(prxparse(/F.*/io)); *- To identify figure outputs -*  
        %let nb_TFLs=%sysfunc(countw(&TFLs));  
        %do i = 1 %to &nb_TFLs;  
            %let TFL = %scan(&TFLs, &i);  
            put "$file = '&tmpdir.\&TFL..rtf'"; *- Define file path in ps1 script -*  
            %regex; *- Macro with powershell regex instructions to be written -*  
        %end;  
    run;  
    x powershell -file "&tmpdir.\chgdt.ps1"; *- Execute script -*  
%mend modTFLruntimes;
```

Business & Decision

KEY CODE BITS – POWERSHELL REGEX MACRO

Life Sciences

```
%macro regex;
```

```
/* Replace general timestamps (DDMMMYYYY hh:mm) */
```

```
put "$regex = '(?<=Date\s&\sTime.*run:\s*)(\d{2}\w{3}\d{4}\s\d+:\d{2})'";
```

```
put "(Get-Content $file) -replace $regex, '00MOK0000 00:00'
|Set-Content $file";
```

```
/* Replace timestamps in figures (yr2000\mo1\dy1\hr14\min24\sec30) */
```

```
%if %sysfunc(prxmatch(&fregex, "&TFL")) %then %do;
```

```
put "$regex = '(?<=\\creativ\\)(yr\d+\\mo\d+\\dy\d+\\hr\d+\\min\d+\\sec\d+)'";
```

```
put "(Get-Content $file) -replace $regex, 'yr2000\mo1\dy1\hr14\min24\sec30'
|Set-Content $file";
```

```
%end;
```

```
%mend regex;
```

Business & Decision

Life Sciences

```
KEY CODE BITS – GENERAL OUTLINE

%macro FileInfo(refdat=, scope=, tdir=, fdir=, ldir=, outdir=, tmpdir=, clean=);
    %init(&tmpdir);                *- Create tmp folder and set environment -*
    %copyTFLs(tdir=&tdir, fdir=&fdir, ldir=&ldir, outdir=&tmpdir);
    %listfiles(&tmpdir);
    %modTFLruntimes(TFLs=&list);    *- Replace timestamps using regex -*
    %TFLinfos(in=&tmpdir, out=&outdir); *- Collects fingerprints and other info -*

    %if &clean eq Y %then %do;
        %clean;                    *- Delete tmp folder and clean environment -*
    %end;

%mend FileInfo;
```

Business & Decision

KEY CODE BITS – FINGERPRINTS. OVERVIEW

Life Sciences

```
filename folder "&in";
filename fn pipe [OS COMMAND FOR FINGERPRINTS];
data tfls;
  length File_name $100 SHA256_checksum $64;
  did=dopen("folder");          *- folder id = 1 -*
  do i=1 to dnum(did);
    File_name=dread(did,i);      *- Read name of file i in myfolder -*
    infile fn; input;           *- Read from virtual file with fileref fn -*
    SHA256_checksum=_infile_;   *- Access current input buffer of INFILE statement -*
    output;
  end;
  rc=dclose(did);               *- close folder with id = 1 -*
run;
```

Business & Decision

Life Sciences

KEY CODE BITS – FINGERPRINTS. OS COMMAND

In theory

```
for %F in (&tmpdir.\*)
do @certutil -hashfile "%F" SHA256
| find /i /v "SHA256"
| find /i /v "certutil"
```

- For every element in tmpdir -
- Compute SHA256 fingerprint -
- Clean output string -
- Clean output string -

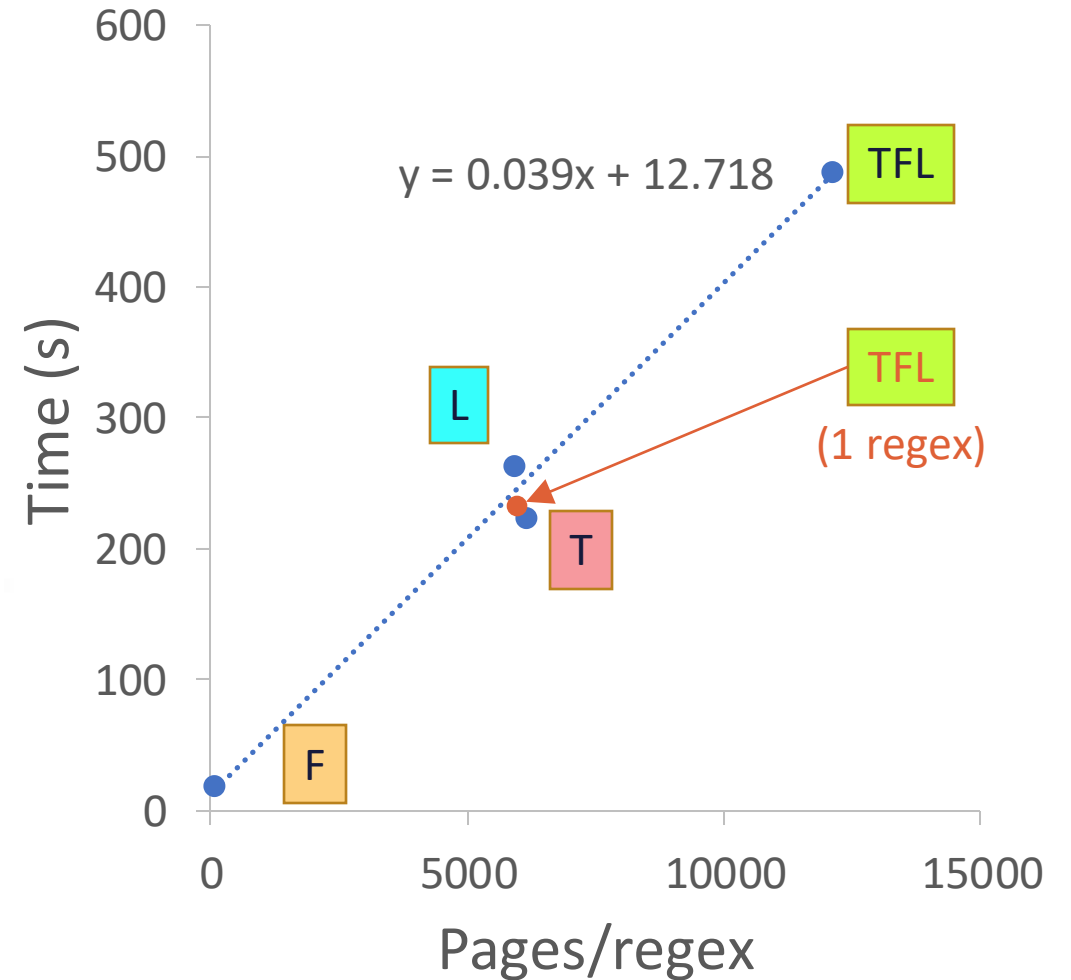
In practice

```
"%str(for %nrstr(%%F) in %(&tmpdir.\*%)
do @certutil -hashfile %"%nrstr(%%F)%" SHA256
| find /i /v %"SHA256%"
| find /i /v %"certutil%"")"
```

Business & Decision

Life Sciences

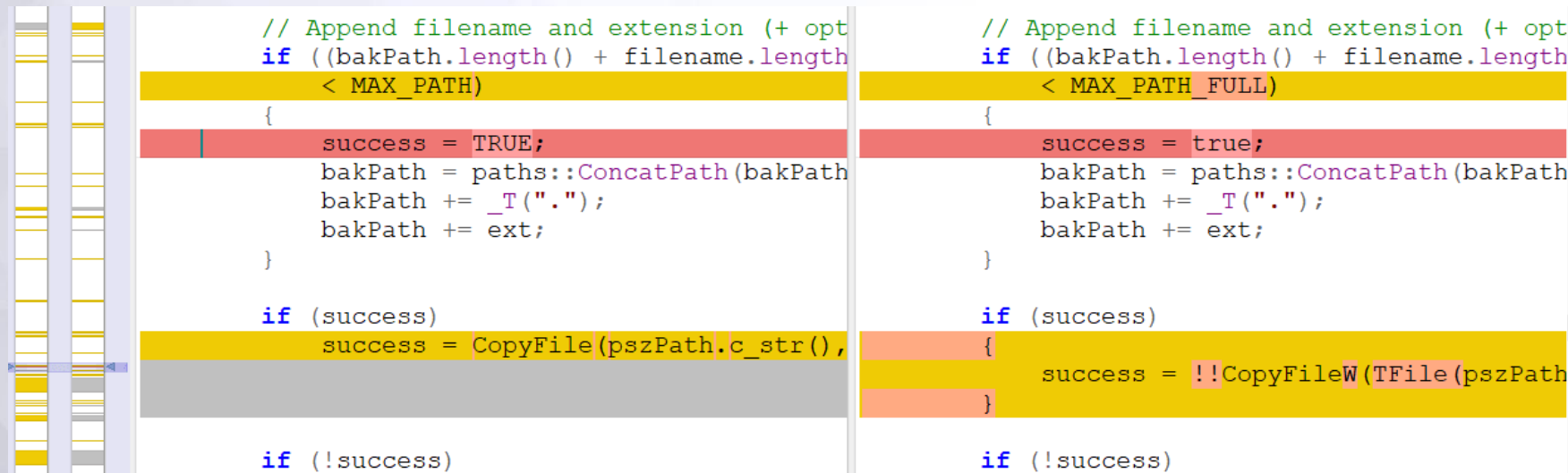
- About 25 pages/sec/regex on a ...
i5-8350U processor
- Linearity of time vs each factor
- Useful for:
 - Post database lock runs
 - Last minute changes QC



Business & Decision

Life Sciences

- Powershell regex complexity and quantity closely related to layout diversity
- Currently not fully optimized
- No detailed comparison of file content differences



The image shows a side-by-side comparison of PowerShell code in a WinMerge window. The left pane shows the original code, and the right pane shows the modified code. The differences are highlighted in yellow and red. The code is a function to append a filename and extension to a path, with a maximum path length check. The modifications include changing the success variable to lowercase 'true', using 'CopyFileW' instead of 'CopyFile', and adding a 'TFile' parameter to the 'CopyFileW' function. Arrows point from the text 'WinMerge highlights differences' to the highlighted areas in the code.

```
// Append filename and extension (+ opt
if ((bakPath.length() + filename.length
    < MAX_PATH)
{
    success = TRUE;
    bakPath = paths::ConcatPath(bakPath
    bakPath += _T(".");
    bakPath += ext;
}

if (success)
    success = CopyFile(pszPath.c_str(),

if (!success)
```

```
// Append filename and extension (+ opt
if ((bakPath.length() + filename.length
    < MAX_PATH_FULL)
{
    success = true;
    bakPath = paths::ConcatPath(bakPath
    bakPath += _T(".");
    bakPath += ext;
}

if (success)
{
    success = !!CopyFileW(TFile(pszPath

if (!success)
```

WinMerge
highlights
differences

- Solutions in other programming languages (R, Python, C, bash,...)
- Notepad++ (Compare plugin)
- Microsoft Word
- Specialized file comparison applications (WinMerge,...)

Business & Decision



Life Sciences

Issues and challenges

- Efficiently detect modified TFL outputs between two runs
- Ignore timestamps changes

Solution

- Modify TFL output copies to reach timestamp neutrality
- Fingerprint each file for comparison between two runs

Results and Benefits

- Improved efficiency and reliability of QC
- Minimal storage impact





THANK YOU FOR YOUR ATTENTION
Q&A

Business & Decision CONTACT Life Sciences



Caroline Pelckmans

Statistical Programmer/ Biostatistician

Caroline.pelckmans@businessdecision.be



Amine Khemiri

Statistical Programmer

amine.khemiri@businessdecision.be



Pamphile Beaujean

Statistical Programmer

pamphile.beaujean@businessdecision.be