

Tracking user requirements and output development

Jon-Paul Mewes, Roche Products Ltd, UK

Xavier Passera, Detour Solutions Ltd., UK

ABSTRACT

The purpose of this paper is to describe how to link electronically and document each part of the study reporting process. This covers extracting title and footnotes from a requirement document, linking them with programs, checking the progress of the programs, documenting the quality control process, assigning the resource to work the study, and viewing the outputs through Excel.

By having an electronically integrated system between requirements and programs, we're able to know the progress of the reporting effort and minimize the programming overhead. This information is particularly valuable to managers to assign resources and distribute the workload.

INTRODUCTION

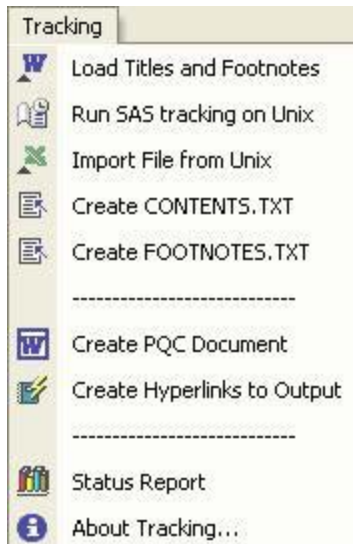
Each study reporting event starts with a DRAM (Statistical Analysis Plan) which programmers use to produce derived datasets and outputs required. This paper will show how a tool can be used to extract titles and footnotes from a DRAM and link them to programs. It also covers showing the progress of programming, the self-documentation of outputs, the documentation of QC of programs and hyperlinking requirements to outputs files.

The tool used, called TRACKING, is built in Excel using macros (VBA). By using Excel we're minimising the training and documentation required while gaining greater initial user acceptance. Some technical aspects will also be covered such as linking Excel to Word and Unix.

This paper describes the process and what can be done using TRACKING. While the presentation will provide a demo of how it works.

1) USING EXCEL

TRACKING is built-in Excel using macros (VBA) and provides the users with a specific toolbar and menubar.



The advantages of using Excel for this process are:

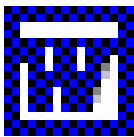
- Tracking programs and outputs is done easily in a spreadsheet
- DRAM is written in Word and titles/footnotes need to be extracted using VBA
- Users already familiar with Excel, thereby reducing documentation and training, while increasing initial acceptance
- Other Excel functions can be used
- TRACKING functions kept to a minimum

The main disadvantages are that support personnel require VBA knowledge and that the tracking tool can only be used by a single person at one time in write mode.

2) LINKING TITLES/FOOTNOTES WITH PROGRAMS

Tracking uses VBA which is common to Microsoft applications such as Word and Excel to extract titles and footnotes from the DRAM. The assumption is that the titles and footnotes in the DRAM are in a pre-defined style and format.

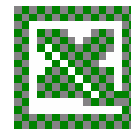
METHOD



DRAM



Send VBA code to Word and produce TOC
Copy TOC to temporary Excel sheet
Combine temp sheet with Footnotes sheet



TRACKING

The programmer then has to assign program/output names to each output.

This is done once, so when new extractions are made all outputs and programs should match.

Titles and programs that don't match are highlighted in TRACKING. This situation occurs when the DRAM has been updated.

The next step is to create a title and footnote file that programs on Unix can access. This will ensure that all outputs have the titles and footnotes specified in the DRAM.

VBA EXAMPLE

The following examples assumes that titles and footnotes have been extracted once and that program and output names have already been assigned.

PhUSE 2006

Titles and Footnotes Load (31/07/2006)							
Program Type	File name	Output name	Table #	Output Order	Title / Footnote label	Page #	Source Document
TYPE	FILE	OUTPUT	TAB	ORDER	LABEL	PAGE	SOURCE
\$40.	\$100.	\$30.	\$20.	8	\$200.	8	\$40.
15 - Tables	ETI.sas	ETI20MINR.r18	T001	0	ACR20 Logistic Regression Analysis at Week 104 - Main E	4	DRAM.doc
15 - Tables	ETI.sas	ETI20MINR.r18	T001	1	Patients with insufficient data to calculate an ACR20 respor	4	DRAM.doc
15 - Tables	ETI.sas	ETI20MINR.r18	T001	2	Patients who entered rescue therapy or withdrew from the st	4	DRAM.doc
15 - Tables	ETI.sas	ETI20MINR.r18	T001	3	Patients who received a single dose of Remicade or 4-weeks	4	DRAM.doc
15 - Tables	ETI.sas	ETI20MIOD.r18	T002	0	ACR20 Logistic Regression Analysis at Week 104 - Main E	4	DRAM.doc
15 - Tables	ETI.sas	ETI20MIOD.r18	T002	1	Repeat the analysis in Table 1 for the ITT population using t	4	DRAM.doc
15 - Tables	ETI.sas	ETI20MIOD.r18	T002	2	Patients with insufficient data to calculate an ACR20 respor	4	DRAM.doc
15 - Tables	ETI.sas	ETI20MIOD.r18	T002	3	Patients who entered rescue therapy or withdrew from the st	4	DRAM.doc

Description of columns

- 1 (TYPE): Category of output
- 2 (FILE): Name of program
- 3 (OUTPUT): Name of output file
- 4 (TAB) : Output number in DRAM
- 5 (ORDER): 0 indicates a title, other numbers indicates order of footnotes
- 6 (LABEL): Title and footnotes
- 7 (PAGE): Page number in DRAM where the template can be found
- 8 (SOURCE): Name of DRAM document

This examples has matched the DRAM with TRACKING (e.g. no additional titles are in the DRAM compared to TRACKING and vice versa) . In this case no differences were found. If new outputs had been found the program name would have been labelled "UNKNOWN.SAS".

Brief description of VBA code:

```
Dim wdApp      As Word.Application
Dim sWDtemplate As Variant
Dim sWDStyle   As String
```

**** Make sure Word is Closed ****

```
wdApp.Quit
Set wdApp = New Word.Application
```

*** Show dialog box. Let user choose DRAM document to open **

```
sWDtemplate = Application.GetOpenFilename _
    (Title:="Select DRAM with titles and footnotes, (style expected: Hd:tab1, hd:fig1, sas7", _
    MultiSelect:=False)
If sWDtemplate = False Then Exit
```

*** Open document as read-only. Go to top of first page **

```
wdApp.Documents.Open Filename:=sWDtemplate, ReadOnly:=True
wdApp.Application.Selection.HomeKey Unit:=wdStory
```

**** Remove any TOC(s) found ****

```
wdApp.Application.Selection.Find.Style = wdApp.ActiveDocument.Styles("TOC Headings")
wdApp.Application.Selection.Find.Execute
```

```
If wdApp.Application.Selection.Find.Found = True Then
    wdApp.Selection.MoveLeft Unit:=wdCharacter, Count:=1
    wdApp.Selection.MoveDown Unit:=wdLine, Count:=2, Extend:=wdExtend
    wdApp.Selection.Delete Unit:=wdCharacter, Count:=1
End If
wdApp.Application.Selection.Find.Execute
If wdApp.Application.Selection.Find.Found = True Then
    wdApp.Selection.MoveLeft Unit:=wdCharacter, Count:=1
    wdApp.Selection.MoveDown Unit:=wdLine, Count:=2, Extend:=wdExtend
    wdApp.Selection.Delete Unit:=wdCharacter, Count:=1
```

```

End If
wdApp.Application.Selection.HomeKey Unit:=wdStory

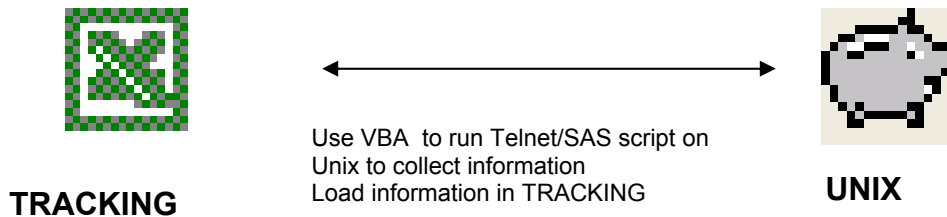
*** Add TOC in Word document based on style defined ***
sWDStyle = "Hd:Fig:1" '*** style to look for in Word **
With wdApp.ActiveDocument
    .TablesOfContents.Add Range:=wdApp.Selection.Range, _
        RightAlignPageNumbers:=True, _
        UseHeadingStyles:=False, _
        UseFields:=True, _
        IncludePageNumbers:=True, _
        AddedStyles:=sWDStyle, _
        UseHyperlinks:=True, _
        HidePageNumbersInWeb:=True, _
        UseOutlineLevels:=False
    .TablesOfContents(1).TabLeader = wdTabLeaderDots
    .TablesOfContents.Format = wdIndexIndent
End With

*** Select then copy TOC and paste it in TRACKING **
wdApp.Application.Selection.HomeKey Unit:=wdStory
wdApp.Selection.MoveDown Unit:=wdLine, Count:=1, Extend:=wdExtend
wdApp.Selection.Copy
oSheet.Paste
    
```

3) STATUS OF PROGRAMMING / DOCUMENTATION

Program names on Unix should follow the naming convention in TRACKING.

METHOD



The information collected from unix is:

- physical location of program
- program status (Development / QA / Production)
- program version
- last modification date
- the names of programmers that have developed and QCed the program
- when the output was created

At the time of database lock, we expect all programs to be in production and all outputs to have been run. The information collected provides an instant check.

The reporting directory may also contain other types of programs, such as ad-hoc request programs or dataset derivation/processing programs. In TRACKING the programmer is asked to categorize programs, in effect documenting what they have created. This brief documentation is most helpful to the team.

PhUSE 2006

Once the information is collected, it needs to be loaded in TRACKING. The link is done using the program and output name.

At this point, in TRACKING, we have a link indicating what's in the DRAM and the current status of the programs on Unix.

Project PHuse: Program/Output Tracking Information					
Directory	Program Type	Output title	DRAM Number	File name	Output name
DIR	TYPE	TITLE	DRAM	FILE	OUTPUT
\$40.	\$40.	\$200.	\$10.	\$100.	\$30.
	22 - Figures	Overview of Patients Disposition	F001	ByHand.sas	ByHandF001.doc
cd11295t	22 - Figures	Plot of Mean CD19 (cells/uL) and Mean DAS28 over Time during the First Treatment	F002	EGcdx.sas	EGcdxdasY1C.cgr
cd11295t	22 - Figures	Plot of Mean CD19 (cells/uL) and Mean DAS28 over Time during the First Treatment	F003	EGcdx.sas	EGcdxdasN1C.cgr
cd11295t	22 - Figures	Plot of Change in Mean DAS28 (from Original Baseline) over Time by Treatment Cou	F004	EGdas.sas	EGdasOHY3Cl.cg
cd11295t	22 - Figures	Plot of Change in Mean DAS28 (from Original Baseline) over Time by Treatment Cou	F005	EGdas.sas	EGdasOHN3Cl.cg
cd11295t	22 - Figures	Plot of Mean CD19 (cells/uL) and Mean ACRn over Time during the First Treatment	F006	EGcdx.sas	EGcdxacrN1C.cg

Description of columns

- 1 (DIR): Unix directory where program is located
- 2 (TYPE): Categorisation of outputs/programs. In this example we're only showing figures
- 3 (TITLE): Title of output
- 4 (DRAM) : Output number in DRAM
- 5 (FILE): Program name on unix
- 6 (OUTPUT): Output name on unix

Programmers	QC names	Programming and Reviewing Comments	Program Status [DEV/QA/PROD]	Program version	Date last modified	Output Date	Output Time
NAME	QC	COMMENT	STATUS	VERSION	LASTMOD	OUTDATE	OUTTIME
\$100.	\$100.	\$200	\$4.	\$6	DDMMYY10.	DDMMYY10.	TIME5.
[Fred]	[Pat]	Hand Generated			.	.	.
zoro1	[Pat]Post1		PROD	1.2	16/11/2005	16/11/2005	17:04
zoro1	[Pat]Post1		PROD	1.2	16/11/2005	16/11/2005	17:04
alice1	Pat		PROD	1.2	24/11/2005	24/11/2005	13:57
alice1	Pat		PROD	1.2	24/11/2005	24/11/2005	13:57
alice1	[Pat]Post1		PROD	1.2	16/11/2005	16/11/2005	17:04

- 7 (NAME): Name of programmer(s) that has/have developed and modified this program
- 8 (QC): Name of programmer(s) that has/have QCed this program
- 9 (COMMENT): Free text.
- 10 (STATUS) : Status of program on Unix. In this example they'll have passed QC and in Production
- 11 (VERSION): Version of program on unix
- 12 (LASTMOD): Date when program was last modified (Information stored on unix).
- 12 (OUTDATE): Date when output was produced on unix
- 12 (OUTTIME): Time when output was produced on unix.

At the end of the study, you would expect to have all your programs/outputs in production and the run date/time to be after the last data delivered.

4) STATUS OF QC AND DOCUMENTATION

Once programs are ready for QC, they appear in TRACKING as "ready". Depending on the type of program, the QC person should review the DRAM to use it as the basis to QC the program/output.

To document and have a consistent approach to QC, a QC template is used to collect this information. A hyperlink is used in TRACKING to link it to the QC document. This standard approach documents the reasons why programs/outputs pass or fail QC and what type of QC is done. Tracking also highlights a program if the person that

PhUSE 2006

developed the program promoted it to PRODUCTION status – this is not allowed in Roche's QC SOP.

This approach eliminates the QC paper trail and allows sharing QC tasks with remote users.

VBA EXAMPLE

Creating a PQC document requires a QC template (shown below). The header part is filled in automatically by TRACKING. It searches for "&&" names and matches them with column names in TRACKING. When the match is found, it replaces the "&&" name with the information in TRACKING

Program Quality Check (PQC) Checklist

Program	&&file		
Name of Programmers	&&name		
Name of QC persons (Unix)	&&QC		
Program Status	&&status	Program Version	&&version
Name of PQC person	&&excel1	Date of PQC	&&excel2

Brief description of VBA code:

```
...
*** Search for keyword in Word template **
With WordMyDoc.Application.Selection.Find
    .Text = "&&"
    .Forward = True
    .Wrap = wdFindContinue
End With

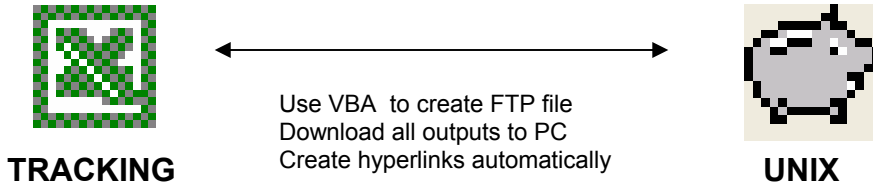
*** Continue search until no more keywords are found **
Do While WordMyDoc.Application.Selection.Find.Execute
    bFound = False
    WordMyDoc.Application.Selection.MoveRight Unit:=wdWord, Count:=1, Extend:=wdExtend
*** Create variable with value of keyword found. eg. AppEXstr=name when &&name is found
    AppEXstr = Trim(UCase(Mid(WordMyDoc.Application.Selection.Text, 3, _
        Len(WordMyDoc.Application.Selection.Text))))
    i = 1
*** Loop on all the column in TRACKING **
    Do
        EXvalue = Trim(Workbooks(sFile).Worksheets(sSheet).Cells(y, i).Value)
        If EXvalue = "" Then EXvalue = "***"
*** If column name = template value (eg. name) then replace text in word with value in TRACKING **
        If Trim(Workbooks(sFile).Worksheets(sSheet).Cells(3, i).Value) = AppEXstr Then
            WordMyDoc.Application.Selection.TypeText Text:=EXvalue
            i = iMaxCol
            bFound = True
        End If
        i = i + 1
    Loop Until i > iMaxCol
Loop
```

5) VIEWING OUTPUTS THROUGH HYPERLINKS

Using TRACKING to view outputs by clicking on hyperlinks is an add-on. This feature can be used for QC purposes but it's most useful for output reviews within and outside the department.

METHOD

PhUSE 2006



The TRACKING folder can be copied to a memory stick and the user has all the requirement documents (DRAM) and outputs linked. This approach reduces the amount of paper used to a minimum, which is particularly useful for off-site meetings.

The process to do this copies all the outputs from unix to the TRACKING folder on PC. There, the link is done on the output name.

VBA EXAMPLE

This example is assuming that the outputs have already been downloaded from unix to pc.

Project PHUSE: ProgramOutput Tracking Information					
Directory	Program Type	Output title	DRAM Number	File name	Output name
DIR	TYPE	TITLE	DRAM	FILE	OUTPUT
\$40.	\$40.	\$200.	\$10.	\$100.	\$30.
	22 - Figures	Overview of Patients Disposition	F001	ByHand.sas	ByHandF001.doc
cd11295t	22 - Figures	Plot of Mean CD19 (cells/uL) and Mean DAS28 over Time during the First Treatment	F002	EGodx.sas	EGodxdasY1.C.pdf
cd11295t	22 - Figures	Plot of Mean CD19 (cells/uL) and Mean DAS28 over Time during the First Treatment	F003	EGodx.sas	EGodxdasN1.C.pdf
cd11295t	22 - Figures	Plot of Change in Mean DAS28 (from Original Baseline) over Time by Treatment Cou	F004	EGdas.sas	EGdasQHY3C.pdf
cd11295t	22 - Figures	Plot of Change in Mean DAS28 (from Original Baseline) over Time by Treatment Cou	F005	EGdas.sas	EGdasQHNR.C.pdf
cd11295t	22 - Figures	Plot of Mean CD19 (cells/uL) and Mean ACRn over Time during the First Treatment	F006	EGodx.sas	EGodxascm.Y1.C.pdf

**Click on hyperlink
Opens output file**

Brief description of VBA code:

```

...
*** Create FTP script file that sends commands to unix***
sFTPlocation=server.ftp.roche.com
Set objMyFile = objFSO.CreateTextFile("u:\script.txt", True)
objMyFile.WriteLine ("open " & sFTPlocation)

** Create dialog box for user to enter Connect to unix via FTP**
objMyFile.WriteLine (sUsername)
objMyFile.WriteLine (sPassword)
objMyFile.WriteLine ("cd " & strFTPServerFolder)
If sFile = "*.cgm" Then
    objMyFile.WriteLine ("binary")
Else
    objMyFile.WriteLine ("ascii")
End If
objMyFile.WriteLine ("lcd " & strLocalFolderName)
*** Copy all files to PC **
objMyFile.WriteLine ("mget " & "*.")
objMyFile.WriteLine ("cd /home/" & sUsername)
    
```

*** Disconnect from unix **

```
objMyFile.WriteLine ("bye")
objMyFile.Close
Set objMyFile = Nothing
```

*** The following code executes the FTP script. It creates a Shell ***

```
*** object and run FTP program on top of it. ***
Set objShell = CreateObject("WScript.Shell")
objShell.Run ("ftp -i -s:" & Chr(34) & "u:\script.txt" & Chr(34))
```

...
Loop on all outputs copied and loop on all outputs in TRACKING, when a match is found add hyperlink information to cell:

```
ActiveSheet.Hyperlinks.Add Anchor:=Selection, *** Cell where match is found
Address:="c:\tracking\out\table.out", *** Full address of output file on PC.
TextToDisplay:="Table.out" *** Text to display in cell. Usually the output name,
```

6) PROJECT MANAGEMENT

As well as aiding the work flow process when programming for study work the Tracking tool provides Lead Programmers with the ability to Project Manage resource and the progress of work.

Using the columns that are filled in within Tracking (Programmers, QC names, Program Status) a number of reports can be run to provide a high level summary:

- Of resource ensuring the work load is distributed amongst the programming team. Example. column G and H indicate who is assigned to do the development of the program/output and who is assigned to QC it. Names in "[xxx]" are the name assigned to do the work, where other names indicate who did the work. All the names of people that have worked on the program/output are listed

DRAM Number	File name	Output name	Programmers	QC names	Programming and Reviewing Comments
FILE	OUTPUT	NAME	QC	COMMENT	
T025	ETacrNR24.sas	ETacrNB24OHN3CA.r8	crossa1	[Keith]willis18	
T026	ETacrNR24.sas	ETacrNB24THY3CA.r8	crossa1	[Keith]willis18	
T027	ETacrNR24.sas	ETacrNB24THN3CA.r8	crossa1	[Keith]willis18	
T028	ETuACR.sas	ETu&CBI042rr.r8	[Keith]watsonk2/willis18	[Bharat]mitchev1	QC Program: qc_long_acr_rescue.sas
T029	ETuACR.sas	ETu&CBI042rn.r8	[Keith]watsonk2/willis18	[Bharat]mitchev1	QC Program: qc_long_acr_rescue.sas
T030	ETfEULAR.sas	ETfEULARBOHY2CI.r8	mistryb/patelh4	mitchev1	

- The status of programs tracking the progress of program development. The nearer to Data Base Lock the more programs you would expect to see in PROD and passed QC. The example on the following pages shows the status of the programs and outputs. The programs are in PROD (indicating that they've passed QC), the program version number and when they were last modified are noted for (audit) information purposes. The output date/time columns attest when the outputs have been created, which is a sure way to know that nothing has been missed.

PhUSE 2006

E	F	J	K	L	M	N
File name	Output name	Program Status [DEV/QA/ PROD]	Program version	Date last modified	Output Date	Output Time
FILE	OUTPUT	STATUS	VERSION	LASTMOD	OUTDATE	OUTTIME
\$100.	\$30.	\$4.	\$6.	DDMMYY1	DDMMYY1	TIMES.
ETacrNR24.sas	ETacrNR24QHN3CA.r18	PROD	1.3	08/11/2005	07/11/2005	11:40
ETacrNR24.sas	ETacrNR24THY3CA.r18	PROD	1.3	08/11/2005	07/11/2005	11:40
ETacrNR24.sas	ETacrNR24THN3CA.r18	PROD	1.3	08/11/2005	07/11/2005	11:40
ETuACR.sas	ETuACR1042rr.r18	PROD	1.30	18/11/2005	18/11/2005	11:59
ETuACR.sas	ETuACR1042rn.r18	PROD	1.30	18/11/2005	18/11/2005	11:59
ETFEULAR.sas	ETFEULARQH2CI.r18	PROD	1.3	19/10/2005	09/11/2005	16:56
ETFEULAR.sas	ETFEULARQH2CI.r18	PROD	1.3	19/10/2005	09/11/2005	16:56
ETFEULAR.sas	ETFEULARTHY2CI.r18	PROD	1.3	19/10/2005	09/11/2005	16:56
ETFEULAR.sas	ETFEULARTHN2CI.r18	PROD	1.3	19/10/2005	09/11/2005	16:56

Brief description of VBA code:

A SAS program on unix collects the information that's supplied to columns F to N (except I) and creates a temporary spreadsheet. This spreadsheet is merged with TRACKING based on column E and F, when a match is found TRACKING is updated with the latest information. No information is deleted in TRACKING, it's up to the users to clean redundant information.

Worksheets("TRACKING").Activate

Set oUnix = Worksheets("UNIX")

For y = 5 To iRowImport <- Loop on rows from file on unix

For i = 5 To iRowTracking <- Loop on rows in TRACKING

If Trim(Cells(i, 6).Value) = Trim(oUnix.Cells(y, 5).Value) Then

....

....

Column J *** 10 - status ***

Cells(i, 10).Value = Trim(oUnix.Cells(y, 8).Value)

Column K *** 11 - version ***

Cells(i, 11).Value = oUnix.Cells(y, 9).Value

Column L *** 12 - Date last modified in Versions ***

Cells(i, 12).Value = oUnix.Cells(y, 10).Value

....

....

Next i

Next y

PhUSE 2006

Overall status of programs/outputs. The report below shows the current status of the study.

Study ABC		Status				
		DEV	QA	PROD	blank	Total
Listing Status	Report					
Type	blank	2	2	1	14	19
	04 - PreProcessing	0	1	7	0	8
	08 - Derived datasets	0	0	12	0	12
	20 - Efficacy Tables	0	0	186	16	202
	22 - Efficacy Figures	0	0	57	2	59
	24 - Efficacy Listing	0	0	14	0	14
	Total	2	3	277	32	314

7) OTHER

TRACKING has a number of other functions that help perform certain tasks, such as creating titles/footnotes ASCII files, create autorun files, create a print file and convert outputs to rtf (pdf) with watermark. These tasks are used to ensure to all the outputs required are accounted for and documented.

- Titles/Footnotes: Two ASCII files are created with the program name, output suffix and the title/footnote. These appear as they do in the DRAM. They will be wrapped on unix as need be.
- Autorun file: Preprocessing programs may need to be run in a certain order or there may be many programs producing the outputs. By running the autorun function, it ensures that no programs is missed. Categorizing programs is the key here as the programs names will be sorted that way.
- Print file: Printing the outputs in the order of the DRAM is useful (!). All the outputs are sorted by program category and DRAM number.
- Convert outputs to rtf (pdf) with watermark. The outputs produced on unix are in ASCII format (except for graphs). This function converts them to RTF (word) and can also convert them to PDF. Adding a watermark is useful if you want to emphasize something. For example, adding DRAFT to ensure they're not used for publications.
-

CONCLUSION

This paper has shown how requirements and programming status can be linked electronically to give current information on a reporting event. This information is useful to biometrics to assess progress, assigning resources, ensure study and QC documentation, ensure that titles/footnotes are accurate and maintain standards. By using Excel, training is minimized and deals only with macros used by TRACKING.

Although the paper has shown how Word → Excel → SAS work together to gather this information, the main point is that it can be done electronically.

REFERENCES

CONTACT INFORMATION

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

Xavier Passera

Detour Solutions Ltd.

Welwyn Garden City, UK

Mobile: +44 (0)7930 472 690

Email: Xavier.passera@detoursolutions.co.uk

Web: www.detoursolutions.co.uk

Jon-Paul Mewes

Roche Products Ltd.

Welwyn Garden City

jon-paul.mewes@roche.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.