

Automating Production of Validation Documentation

Tangi Sanséau, ICON Clinical Research, Dublin, Ireland

ABSTRACT

When outputting data and/or statistics with the SAS software®, the regulatory requirements mean that validation of these outputs has to be performed to ensure their accuracy. One of the most common processes used for validation is independent/duplicate programming. As a result of this method, Validation Forms are to be used as well as Tracking Forms to document the progress of the Development programs as well as the Validation programs.

While necessary, this documentation procedure can be very time-consuming and therefore creates a need to automate the production of this documentation as much as possible.

In this paper, I will describe how, from a shared Tracking Excel file where every programmer can enter his progress on his own program, a validation package is created and automatically transferred to a secure Archive location. This package includes the SAS programs, outputs and logs. The signed Validation forms can automatically be generated whenever the person in charge of the validation confirms that the process is complete. In addition, they are e-mailed and put together in that same secure location.

I will present a possible structure for the tracking file and the options necessary in the validation forms so that the forms corresponding to a validated program can be created automatically using some Visual Basic® code.

INTRODUCTION

In the CRO Business, when data has to be submitted to the regulatory Authorities, documentation and tracking are two primary key points of success. As the double programming method is common in this sector, there is a need for documentation, which can become very time-consuming. Therefore, it is useful when any step of the process can be automated. This paper describes the tool created to automatically generate the documentation required by the double programming process and automatically archive the corresponding electronic files. The structure presented here is an example of what can be done as this tool is designed using Visual Basic® code, specifically VBA® applied to Microsoft Word® and Microsoft Excel®.

Even if the tracking tools used can slightly differ from one company to the other, the method presented in this paper is flexible and can be adapted to any situation. The examples I will use in this paper will only deal with Validation Forms for Datasets, but the process and Visual Basic Code is exactly the same for Listings, Tables and Graphs. This tool currently transfers the SAS Programs, Outputs and Logs of the program that is deemed as validated, but other electronic documents could be added to that package as required.

The general structure of the Excel file is also for example and could also be easily modified according to specific templates.

EXCEL TRACKING SPREADSHEET

The Tracking Spreadsheet should contain the necessary information such as the name and initials of the developer of the SAS program, the name and initials of the person who validated the SAS program, the date when the SAS program was validated, and the standard information regarding the project, such as the study Number, the name of the corresponding client, the study team and their role.

I chose to have two separate Tabs in this Tracking Spreadsheet:

1 – The “Study Team” tab will contain the Initials, Role Title and Full Name of each member of the team. The information regarding the type of study, phase, and sponsor Name can also be put in this Tab. Finally, the location to store the electronic packages that will be created upon validation of a program is stored as an e-mail address. This address location is an Intranet website called “e-Room” that contains all electronic documents related to a project.

PhUSE 2008

Here is how the Tab looks like:

Microsoft Excel - Validation Tracking Spreadsheet 123-456.xls					
Type a question for help					
A1 Sponsor					
1	Sponsor	SPONSOR_NAME			
2	Title	A Double-Blind, Randomized Study to Evaluate the Efficacy and Safety of PRODUCT_NAME versus Placebo in Subjects with a disease			
3	Phase	I			
4	Protocol Number	SPONSOR-123-456			
5	e-Room e-mail address	123-456@iconplc.com			
6					
7					
8	Study Number	123/456			
9	Clinical PM	To be confirmed			
10	DM Project Manager	To be confirmed			
11	DM Project Lead	To be confirmed			
12	DM Study Lead	To be confirmed			
13	DM DBA	To be confirmed			
14					
15	Study Team:	Name	Initials	Location	Role
16	Project Statistician	John Doe	JD	Dublin	Senior Statistician
17	Project Programmer	Tangi Sanséau	TS	Dublin	Project Programmer
18	Peer Reviewer	To be defined			
19	Additional Support	Mark Doe	MD	Dublin	Support Programmer
20		Philip Doe	PhD	Dublin	Support Programmer
21					
22					

I will later describe the main parts of Visual Basic code put together to read all the information contained in this Tab.

2 - The "Validation" Tab: This Tab contains the Sponsor Name, Study Number and Protocol Number. The folders in which are located the SAS Development programs and the SAS Validation program are also included in this Tab. Other information such as the SAS Version Number or Operating System are displayed.

In this paper, I chose to include only one example of SAS Outputs: the Datasets. However, the process would be identical if Validation Forms need to be created for Listings, Figures or Tables. Here is an example of how the "validation" Tab looks like:

Microsoft Excel - Validation Tracking Spreadsheet 123-456.xls										
Type a question for help										
Q27										
A1 Sponsor										
1	Sponsor:	SPONSOR_NAME								
2	Study:	123/456								
3	Protocol:	SPONSOR-123-456								
4	Development:	A:\Data\Sponsor\123_456\Development								
5	Validation:	A:\Data\Sponsor\123_456\Validation								
6	SAS Version:	SAS V8.2								
7	Operating System:	Windows Server 2003								
8										
9	Program Name (*SAS)	Title	Type	Program Author	Status	Date Completed	Method of Validation	Program Validator	Status	Date Validated
10	D_AE	Adverse Events	Datasets				Independent Program			
11	D_CD	Concurrent Medical Conditions	Datasets				Independent Program			
12	D_CM	Medication History and Concomitant Meds.	Datasets				Independent Program			
13	D_DM	Demography Data	Datasets	TS	Completed	20-Feb-08	Independent Program	JD	Completed	6-Mar-08
14	D_GL	Global Information per Study	Datasets	TS	Completed	26-Feb-08	Independent Program	JD	Completed	27-Feb-08
15	D_IE	Inclusion/Exclusion Criteria	Datasets	PhD	Completed	26-Feb-08	Independent Program	MD	Completed	6-Mar-08
16	D_KV	Key Variables	Datasets	PhD	Completed	16-Jan-08	Independent Program	MD	Started	
17	D_LB	Lab Data	Datasets				Independent Program			
18	D_MH	Medical History	Datasets				Independent Program	JD	Started	
19	D_PE	Physical Examination	Datasets	TS	Started		Independent Program			
20	D_RD	Randomization	Datasets				Independent Program			
21										
22										

PhUSE 2008

You can see from this screenshot that the mandatory fields are the name of the SAS program, the corresponding Title, the Type of Output, the Program Author, status of the Development Program, and Date Completed, the Method of validation, initials of the program Validator, status of the validation program and date validated.

PROCESS

Here is how it works during the course of the study:

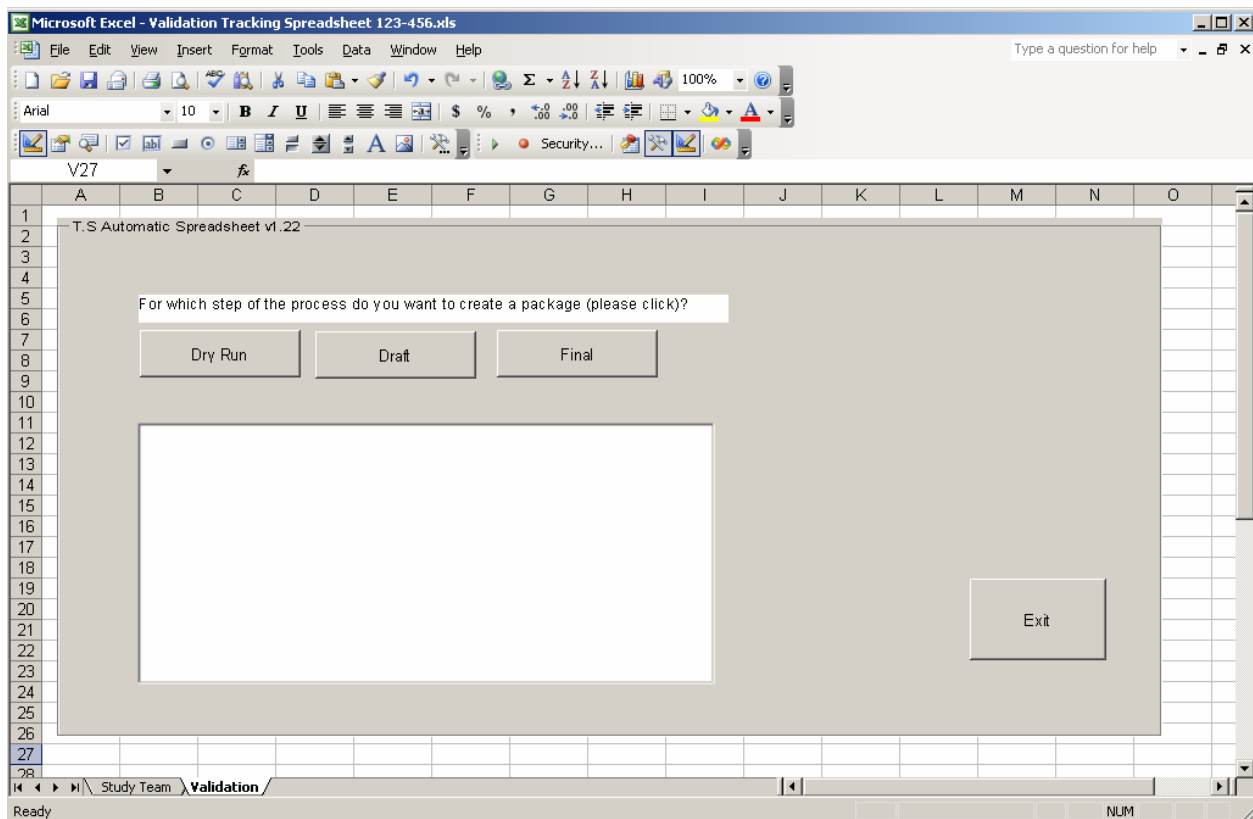
Upon initiation of the project, all needed outputs are listed in this "Validation" Tab, along with a title and a type (I only displayed "Datasets" in the example here above, but as stated earlier, "Listings", "Tables" or "Figures" could be displayed as well...).

As the programming starts, each Programmer is responsible for updating the Spreadsheet by putting his initials in the Program Author column (column D here above) or Program Validator column (column H here above) depending on which program he is programming/validating and the status of the program. Only 3 Status exist: if the cell is empty, the program is not started yet; otherwise, the only possible values for the Development program are "Started" and "Completed" whereas the Validator can choose between "Started", "Validated" or "Query Pending" if he came back to the Development programmer with issues to be checked and/or corrected in his program.

Once a validation programmer set the status of his program as "Complete" in the column I herea bove, the objective is to generate automatically the corresponding Validation Form and automatically save a package including every electronic file validated (generally the SAS development program, SAS Validation program, as well as their corresponding SAS logs and SAS outputs), along with an electronic copy of the signed Validation Form.

This is the basis of the Visual Basic macros I created to automate the complete process of Validation documentation once a program is finalized.

Upon validation of a SAS program, the user clicks on the Command Button "Generate Validation documentation" which displays a Frame such as the one herebelow:



As shown here above, the user is left with 4 possible actions (corresponding to the 4 "Command Button"s displayed on the frame).

PhUSE 2008

1 - The “Exit” button exits the procedure and allows the user to come back to the initial Tracking Spreadsheet. No action is taken.

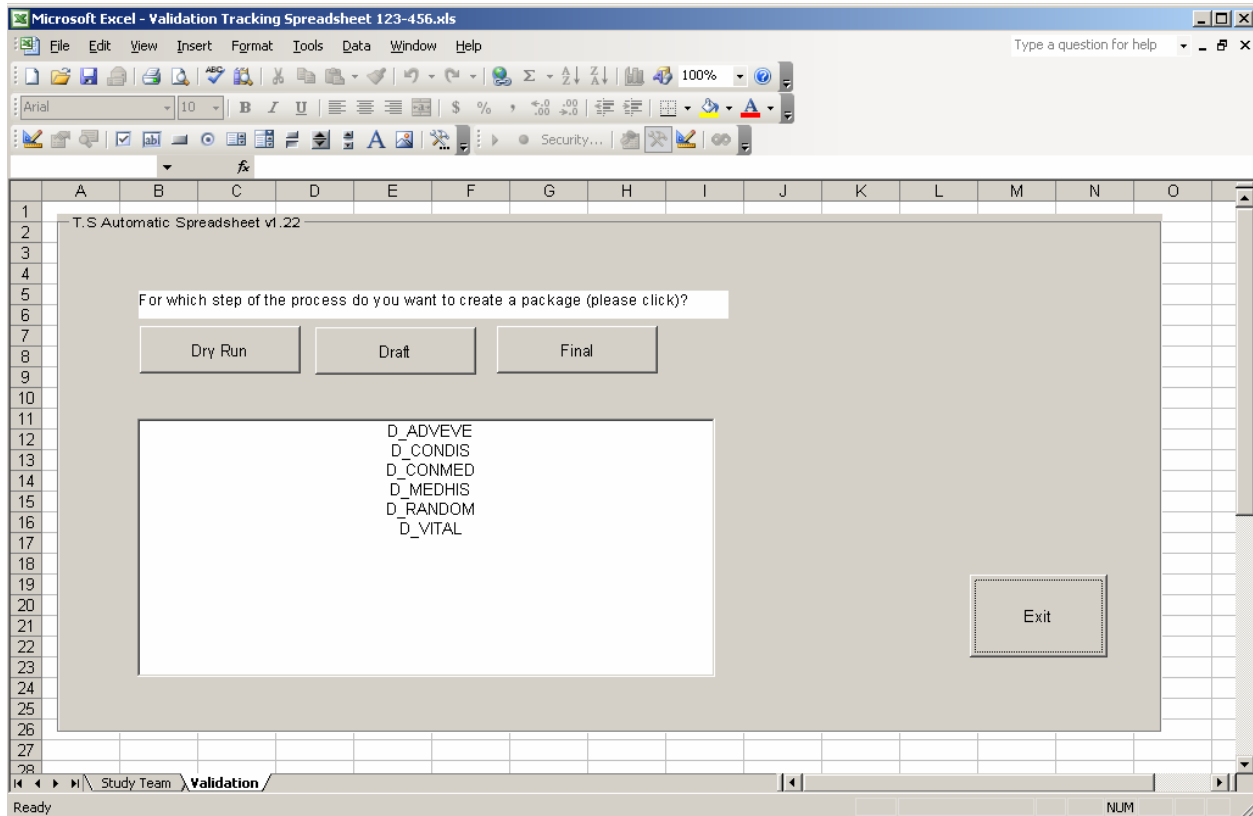
2 – The 3 other buttons are in fact 3 only because the Tracking spreadsheet (and specifically the “Validation” Tab) states that there will be 3 steps in the process of this particular project. Upon modification (for instance, the project only consists of one single Delivery to the client), there would only be one “Command Button” displayed on this Frame.

For this example, I displayed 3 steps: a “Dry Run (corresponding to the delivery of the TLGs for a certain amount of the patients based on an interim transfer of data), a “Draft” (corresponding to a delivery of the TLGs for 100% of the patients) and a “Final delivery”, in which the comment on the Dry Run and the Draft have all been updated and which will be considered as the final delivery.

From a technical point of view, clicking on any of the 3 steps of the process only modifies the location of where the program is looking for the data. Herefore, I will only describe what is the process if the user clicks on the “Dry Run” as it is identical for the 2 other remaining commands.

Once the user has selected the step of the process he’s interested in creating Validation Forms for, by clicking on the appropriate “Command Button”, the List Box displayed just under the Buttons, and initially blank, will populate with the list of programs that are validated as per the Tracking Spreadsheet, for the step that the user chose.

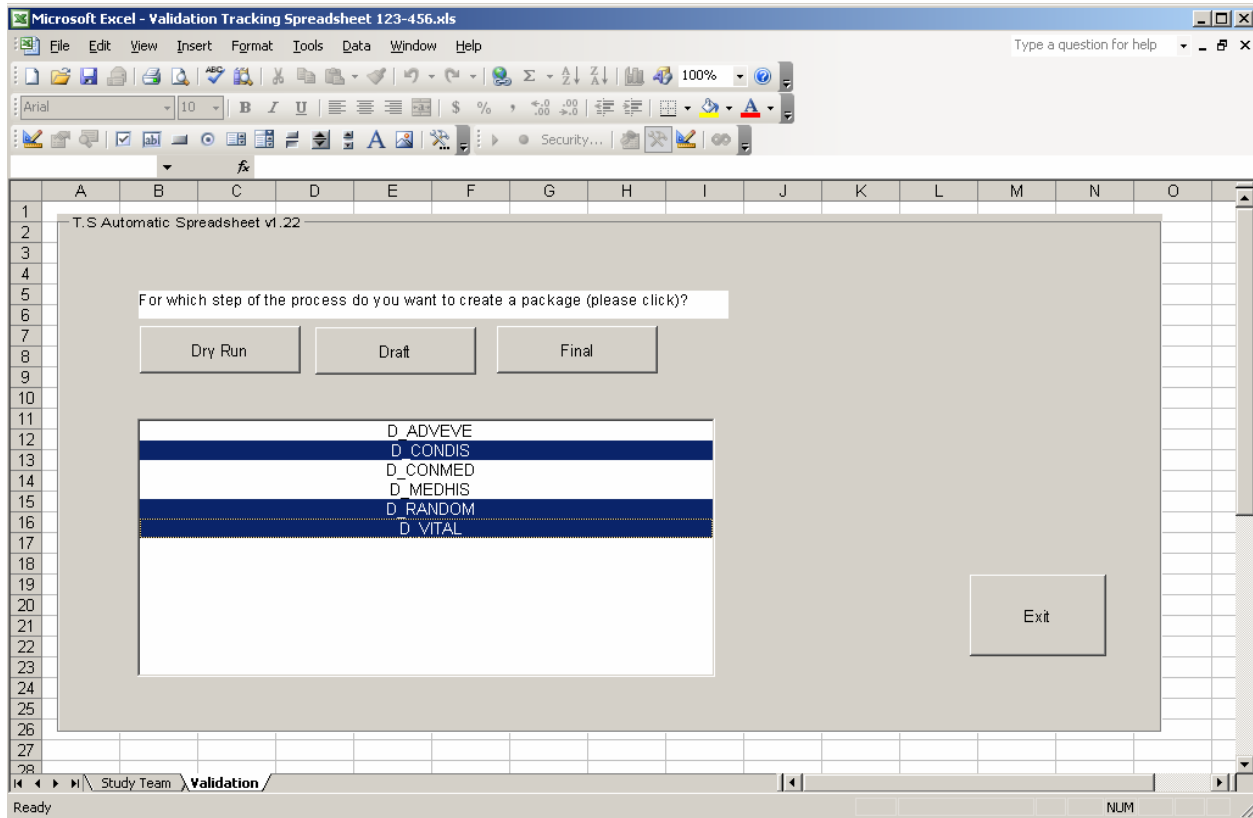
This is done with the Visual Basic Code identifying the section of the Validation Tab corresponding to the step selected. Then the code looks into the Validation Status column for all programs that are recorded as “Complete”.



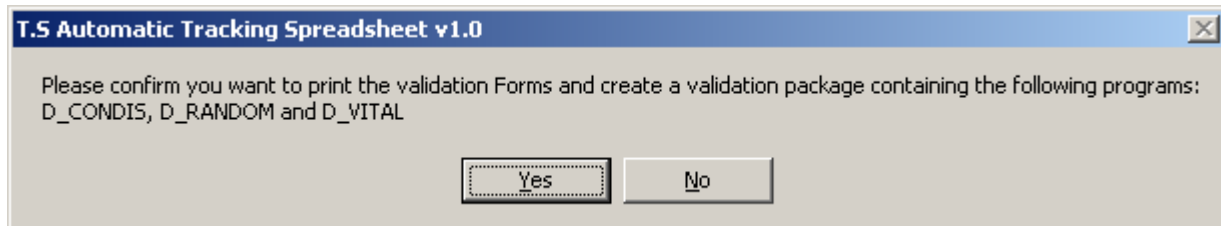
A selection is possible as shown hereunder (whether single or multiple):

From the Tracking Spreadsheet (see screenshot here above), 6 datasets are currently validated: 3 have been validated on the 27FEB08 and 3 others on the 06MAR08). If we assume that a validation package has already been created for the 3 programs validated on the 27FEB08, the user can choose to create a package with only the 3 programs validated on the 06MAR08:

PhUSE 2008



The user is now be able to select the programs he is interested in creating a package for. The Visual Basic code created detects if a selection has been done or not (and would appropriately display a message to the user if no selection is performed), and if at least one program is selected, prompts a message for the user to confirm his selection.



Upon confirmation, the program will search into the directory the electronic files that have to be saved thanks to the information it reads from the "Team" and "Validation" Tabs of the Excel file. All SAS programs, logs and outputs from both the Development and Validation areas are therefore zipped together and automatically sent per e-mail to the electronic "storeroom" corresponding to the study. This is possible since every project has a corresponding e-Room and each one of these e-Rooms has an e-mail address (which should be recorded on the "Team" Tab, so that the program is able to automatically send the e-mail).

Once the "package" is electronically stored, the last step is the documentation of the validation, hence a signed Validation Form.

The Templates of the Validation Forms we used are not study specific and therefore are stored in one single location for all projects. If there is a need for study-specific Templates, their location could be recorded in the "Team" Tab as the other data specific to the project and would be read by the program as all these other parameters.

As per ICON's current procedures, both the validation dates and the signature of both development and validation programmers have to be handwritten. Therefore, the program outputs in the Templates the name of the program for which the validation Form is created, its location and the study Number and automatically prints the created form. Once printed, the new Word document created is electronically stored in the e-Room.

PhUSE 2008

The only thing left to do is for both programmers to sign and date the printed form. The signed form can then be scanned and sent to the e-Room as well.

Here are the setup that needs to be done on the Validation Form Templates:

WORD VALIDATION FORMS TEMPLATE

The Templates to be used are Microsoft Word documents. The example described in this paper uses a Template for datasets. Obviously, this can be adapted and modified to suit other criteria, as required.

MANDATORY FIELDS

In order to properly setup the document, Text boxes need to be inserted in the Template document used as the Validation Form. The mandatory Text boxes correspond to the mandatory fields listed in the Tracking Spreadsheet: Study Number, Program Name, Program Location, Program Developer's Name, Program Validator's Name and Date of Validation.

In order to insert TextBoxes, the Visual Basic toolbar has to be displayed in Word. Then, by clicking on the TextBox button in this toolbar, each TextBox can be inserted where appropriate. Once they are included, their name can be also changed appropriately. I simply named them: Study_Number, Program_Name, Program_Location, Developer_Name and Validator_Name. The Validation Date is only for information, to be displayed in the name of the file and the name of the zip package that is created, as it should be handwritten on the Form per ICON's current procedures.

Another field that will not be automatically filled is the Comments field. Indeed, prior to being saved the Form created is displayed for the user to enter his comment electronically if appropriate. This comment can be various, but here are a few examples:

- "There is currently a data issue with patient 001 which should be resolved when the raw dataset is updated and transferred again."
- "The programming is based on Draft 1.0 of the Specifications, and will be re-validated upon the release of a final version."
- "The validation program was written by JD but the outputs were compared by MD."

If the user has comment to enter on the form, he can therefore do it while the form is displayed on the screen, or not if not applicable. This step can also help to ensure the accuracy of what has been automatically read from the Tracking Spreadsheet.

Once checked and/or updated, the user can save it. Clicking on the "Save" button of Word will automatically propose the user the proper location and name for the file.

From the information stored by the program, the default name for any validation Form will be: "Type of Output Name of the program – Step of the process - Validation_date – Initials of the Developer – Initials of the Validator.doc". For example, the Validation form for the D_ADVEVE shown previously would be assigned the following name:

"Dataset D_ADVEVE – Dry Run – 20AUG2008 – TS – JD.doc".

PhUSE 2008

Here is an example of how the Template Validation Form looks like:

DATASET VALIDATION FORM			
ICON Study Number			
Program Name			
Program Location			
PROGRAM		√	N/A
Header			
1	The title is correct	☒	
2	The name and location of the file is correct	☒	
3	All the datasets used in the program are listed	☒	
4	The author/modifier and date are entered. Modification details are provided, as appropriate.	☒	
Content			
5	The program is clearly laid out	☒	
6	The program is well commented	☒	
7	The output file (if any) and log are correct	☒	
OUTPUT		√	N/A
Appearance			
8	Are all the variables on the dataset specification present?	☒	☐
9	Are all the variables named in accordance with the specifications?	☒	☐
10	Are all the variables labelled in accordance with the specifications?	☒	☐
11	Do all variables have the correct format/length applied?	☒	☐
Data			
12	Do <i>all</i> (derived, raw) variables in the original derived dataset match the corresponding variables in the duplicate program (including values and attributes)?	☒	
13	Do they have the same number of records?	☒	
14	The log contains no errors or warnings	☒	
Failed Check No	Comment/Resolution	Resolved	N/A
		☐	☐
		☐	☐
Validator			
Signature		Date	
Programmer			
Signature		Date	

In this screenshot, I left the border of each Textboxes in black so that we can see where they are located, but they can be set as White so that they are invisible, and therefore, we only see the values they display when they are printed. This can be done using the "BorderStyle" option of each of these Textboxes.

PhUSE 2008

Once filled, you obtain the following:

DATASET VALIDATION FORM			
ICON Study Number	123/456		
Program Name	D CONDIS		
Program Location	A:\Data\Sponsor\123 456\Development\Dataset\		
PROGRAM		√	N/A
Header			
1	The title is correct	☒	
2	The name and location of the file is correct	☒	
3	All the datasets used in the program are listed	☒	
4	The author/modifier and date are entered. Modification details are provided, as appropriate.	☒	
Content			
5	The program is clearly laid out	☒	
6	The program is well commented	☒	
7	The output file (if any) and log are correct	☒	
OUTPUT		√	N/A
Appearance			
8	Are all the variables on the dataset specification present?	☒	☐
9	Are all the variables named in accordance with the specifications?	☒	☐
10	Are all the variables labelled in accordance with the specifications?	☒	☐
11	Do all variables have the correct format/length applied?	☒	☐
Data			
12	Do <i>all</i> (derived, raw) variables in the original derived dataset match the corresponding variables in the duplicate program (including values and attributes)?	☒	
13	Do they have the same number of records?	☒	
14	The log contains no errors or warnings	☒	
Failed Check No	Comment/Resolution	Resolved	N/A
		☐	☐
		☐	☐
Validator	John Doe		
Signature		Date	
Programmer	Tangi Sanséau		
Signature		Date	

As per ICON's current procedures, the signature of each the Development programmer and the Validation programmer as well as the Development Date and Validation Date have to be hand-written on the Form, therefore, once all fields shown hereunder are automatically filled from the information contained in the Tracking Spreadsheet, the Form is automatically printed and saved in the directory specified per the Spreadsheet.

When a project is over, this whole automated process ensures that all Validation Forms are properly archived, as well as the programs, logs and outputs coming from SAS. It is the responsibility of the Validation programmer to ensure that all scanned Validation Forms are also sent to the e-Room.

Based on the process I described in this paper, the Validation programmer and Development programmer will always be matching the Tracking Spreadsheet information, as well as the dates of validation, since this information will be displayed in the name of the files stored in the e-Room.

PhUSE 2008

The Quality Assurance Department can then easily compare the Tracking Spreadsheet to the content of the e-Room to ensure accuracy and completeness.

From a more technical point of view, here are a few details regarding how the Visual Basic Code works:

VISUAL BASIC CODE

PRINTING THE VALIDATION FORMS

```
If MsgBox("Do you want to print the following Form: " & V_Folder & "\Validation
Form\" & Type_F & "-" & Pgm_Name & " - " & Date_V & ".doc", vbYesNo, "T.S Automatic
Tracking Spreadsheet v1.0") = vbYes Then
    DocWord.SaveAs (V_Folder & "\Validation Form\" & Type_F & "-" & Pgm_Name & " - "
    & Date_V & ".doc")

    'Count how many Forms are created (For user information)
    Count_Forms = Count_Forms + 1

    'PRINT commands
    Word.ActivePrinter = "safecom"
    With Word
        .PrintOut filename:=(V_Folder & "\Validation Form\" & Type_F & "-" &
Pgm_Name & " - " & Date_V & ".doc")
    End With
    'end of PRINT commands
    DocWord.Close
EndIf
```

Using VBA, the code can be used to point at an external application, such as Word.

For more precision, I chose to display the Type of program (either Datasets, Listings, Tables or Figures) in the name of the validation forms, as well as the program name and the date of validation. Indeed, the same program will usually be validated more than once, as there is often several deliveries for one single project.

CREATING THE ZIP FILE WITH ALL ELECTRONIC FILES OF THE VALIDATION PACKAGE

```
'Count how many Forms are created (For user information)
Count_Forms = Count_Forms + 1

'ZIP commands
With WinZip
    .Create_New_archive
    .Include filename:=(D_Folder & "\Programs\" & Type_F & "\" & Pgm_Name & ".sas")
        'SAS Development program
    .Include filename:=(D_Folder & "\Programs\" & Type_F & "\Outputs\" & Output_Name
& ".lst") 'SAS Development output (lst file)
    .Include filename:=(D_Folder & "\Programs\" & Type_F & "\Outputs\" & Output_Name
& ".rtf") 'SAS Development output (rtf file)
    .Include filename:=(D_Folder & "\Programs\" & Type_F & "\Logs\" & Output_Name &
".log") 'SAS Development log

    .Include filename:=(V_Folder & "\Programs\" & Type_F & "\" & Pgm_Name & ".sas")
        'SAS Validation program
    .Include filename:=(V_Folder & "\Programs\" & Type_F & "\Outputs\" & Output_Name
& ".lst") 'SAS Validation output (lst file)
    .Include filename:=(V_Folder & "\Programs\" & Type_F & "\Outputs\" & Output_Name
& ".rtf") 'SAS Validation output (rtf file)
    .Include filename:=(V_Folder & "\Programs\" & Type_F & "\Logs\" & Output_Name &
".log") 'SAS Validation log
End With
'end of ZIP commands
```

The access to Winzip® is very much like accessing Microsoft Word. With the VB code, we can create a new zip file, initially empty, and include as many files in it as needed. In the example here above, only one program is packed in one zip file which will therefore include both the development and validation programs, logs, outputs (in both .lst and .rtf formats).

PhUSE 2008

SENDING THE VALIDATION PACKAGE TO THE DOUMENT ELECTRONIC STOREROOM

```
'Count how many Forms are created (For user information only, it will be later be
displayed when all Forms are printed)
Count_Forms = Count_Forms + 1

'SEND commands
With Outlook
    .Create_New_e-mail
    .Attachments filename:=(V_Folder & "\Validation Form\" & Type_F & "-" & Pgm_Name
& " - " & Date_V & ".zip")
    .SendTo E-Room_address & "@iconplc.com" 'E-Room_address information coming from
the Tracking Spreadsheet
End With
'end of SEND commands
```

As for Microsoft Word and Winzip, Microsoft Outlook® can be accessed with VB Code. In the example above, a new blank e-mail is created, to include as many attachments as needed (only one zip file in the example corresponding to the zip file previously created). The e-mail address that was read from the Tab “Team” of the Tracking Spreadsheet can then be displayed in the “To” field of the e-mail.

This e-mail will result in the zip file automatically being saved into the corresponding e-Room.

CONCLUSION

This paper demonstrates the advantages that can be gained by automating the production of Validation documentation, which though necessary, can be a tedious and time-consuming task. The code presented here can easily be adapted to suit a variety of requirements. Thus, much of the time spent compiling documentation in clinical trials can be eliminated, without compromising the integrity of the documentation itself.

Finally, upon completion of a project, the check of the process is facilitated for the Quality Assurance Department by ensuring that every package is properly located in the electronic fileroom.

REGULATORY REFERENCES

ICH Guidelines E9, “*Statistical Principles for clinical trials*”, dated 5 February 1998.

CONTACT INFORMATION

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

Tangi Sanséau
ICON Clinical Research
South County Business Park
Leopardstown
Dublin 18
Ireland

Work Phone: +353 1 291 2483
Fax: +353 1 291 2723
Email: Tangi.Sanseau@iconplc.com
Web: www.iconclinical.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.