



The Next Generation Smart Program Repository

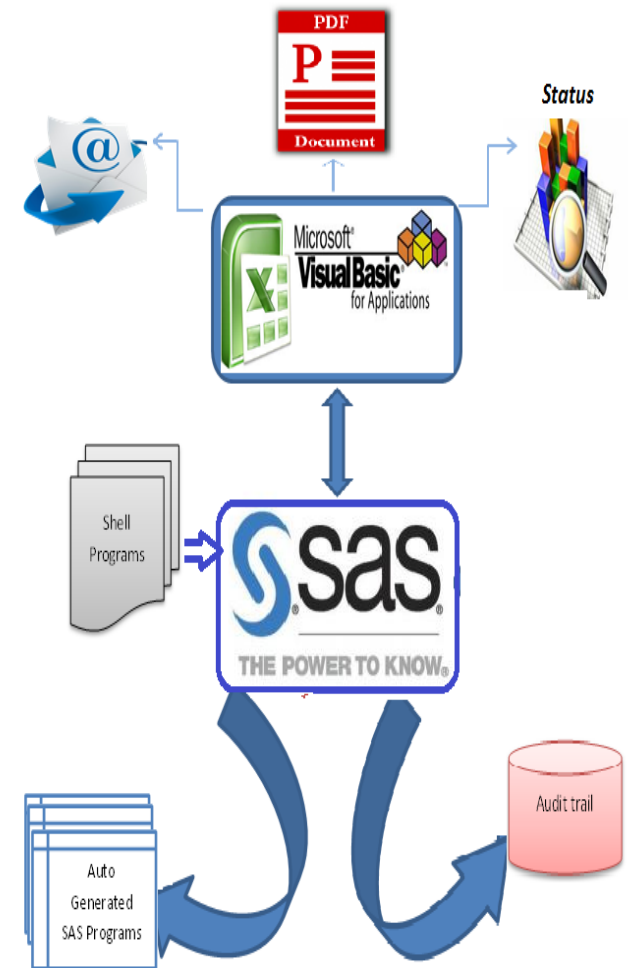
Hrideep Antony, Syneos Health
Aman Bahl, Syneos Health

Introduction

- Streamline and improve the efficiency of the program developmental cycle
- Live tracking of overall project progress
- Increase programming compliance by automated program shell generation
- Centralized control of overall program creation and project assignment

Process Flow

- Excel VBA front-end screen acts as an interface and controls the overall repository
- Status summary screen provides instant and customizable summary reports of project progress
- SAS® utility macros, generates programs and autopopulate program headers, based on information provided in the VBA screen



Excel VBA front-end screen

Add Records

Submit

Status

Email Status

Sr_no	Index_no	Programmer	Validator	Status	Folder_location	Program_name	description
1	INDX-1	Programmer2	Programmer1	Ready for PS Review	/programs/macros	test.sas	Table to create X
2	INDX-2	Programmer1	Programmer2	In Validation	/programs/macros	test22.sas	Table to create X
3	INDX-3	Programmer1	Hrideep	In Validation	/programs/macros	test23.sas	Table to create X
4	INDX-4	Programmer1	Programmer2	In Production	/programs/macros	test24.sas	Table to create X
5	INDX-5	Programmer1	Programmer2	In Validation	/programs/macros	test28.sas	Table to create X
6	INDX-6	Programmer2	Programmer2	not Started	/programs/macros	test30.sas	Table to create X
7	INDX-7	Programmer2	Programmer1	Ready for PS Review	/programs/macros	test26.sas	Table to create X
8	INDX-8	Programmer3	Programmer1	Ready for PS Review	/programs/macros	test29.sas	Table to create X
9	INDX-9	Programmer1	Hrideep	In Validation	/programs/macros	test31.sas	Table to create X
10	INDX-10	Programmer1	Programmer2	In Validation	/programs/macros	test2.sas	Table to create X
11	INDX-11	Programmer1	Programmer2	not Started	/programs/macros	test33.sas	Table to create X
12	INDEX-12	Programmer1	Programmer2	In Production	/programs/macros	test5.sas	Table to create X
13	INDEX-13	Programmer1	Programmer2	not Started	/programs/macros	test36.sas	Table to create X
14	INDEX-14	Programmer2	Programmer2	not Started	/programs/macros	test38.sas	Table to create X
15	INDEX-15	Programmer1	Programmer2	not Started	/programs/macros	test.sas	Table to create X
16	INDEX-16	Programmer1	Programmer2	In Validation	/programs/macros	test3.sas	Table to create X

Programmer and validator cannot be same at: INDX-6

Add Records button creates new record with Default Status as "Not Started"

Submit Button activates the First line of validation checks on the contents and does colour assignments based on the status column

Dynamic Colour assignments based on Status

Status button takes to the status summary pages

Excel VBA Status summary screen

Generate report

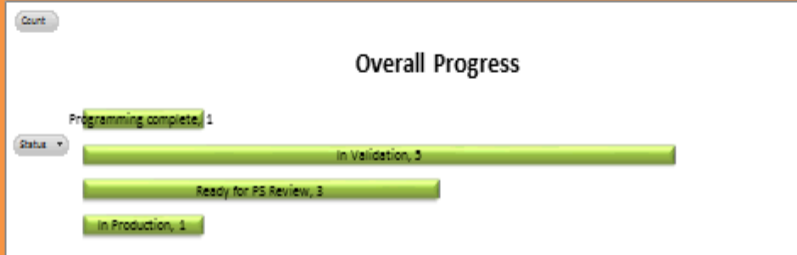
Return

Email Status

Status as 11/24/2017 17:41

Programming	Status						Validation	Status					Overall Status	
Name	not Started	Programming complete	Missing	Ready for PS Review	In Production	In Validation	Name	In Production	Ready for PS Review	In Validation	Programming complete	Status	Count	
Programmer1	3				2	6	Programmer2	1		4		In Production	1	
Missing							Programmer1		3	1		Ready for PS Review	3	
Programmer2	1	1		2			Grand Total	1	3	5	1	In Validation	5	
Programmer3				1									Programming complete	1
Grand Total	4	1		3	2	6							Grand Total	

Overall Progress



Index_no

INDX-1

INDX-10

INDX-2

INDX-3

INDX-4

INDX-5

Programmer

Programmer1

Programmer2

Programmer3

Hrideep

Status

In Production

In Validation

Programmin...

Ready for P...

In Valiation

not Started

Validator

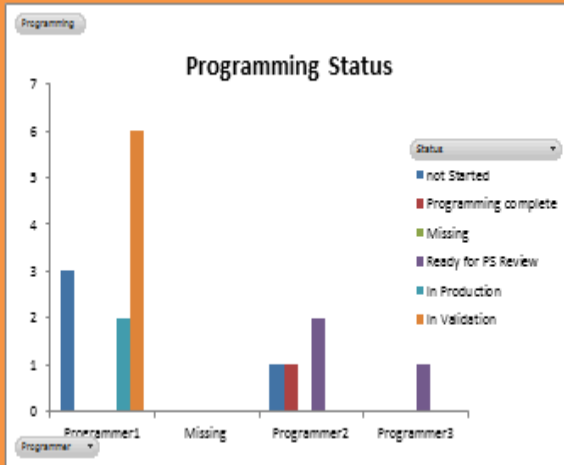
Programmer1

Programmer2

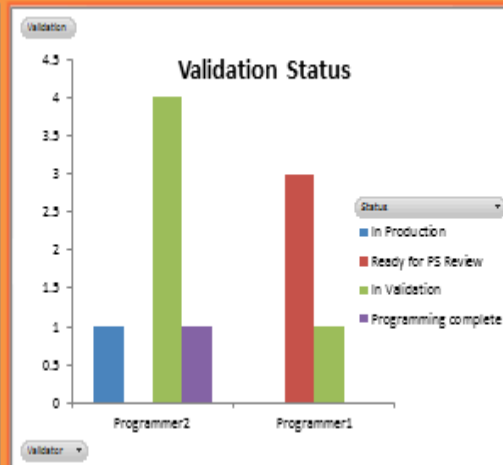
Hrideep

not Started

Programming Status



Validation Status



Submit Button

- Submit Button activates validation checks such as missing program names, and same programmer and validator assignments and assigns a color to each record based on the status.
- Red color is assigned and an error message is displayed when the key values are missing.
- Error message is displayed when both programmer and validator names are same.

```
Private Sub submit_button()  
    'defining and assigning values to the variables  
    Dim i As Long, r1 As Range, r2 As Range, r3 As Range, r4 As Range  
    ' defining the loop and assign cell values to variables to check the validity of data  
    For i = 2 To 1000  
        Set r1 = Range("E" & i)  
        Set r2 = Range("A" & i & ":O" & i)  
        Set r3 = Range("C" & i)  
        Set r4 = Range("D" & i)  
        ' Assign colour codes based on the status of each cell  
        If r1.Value = "In Validation" Then r2.Interior.ColorIndex = 37  
        If r1.Value = "In Production" Then r2.Interior.ColorIndex = 4  
        If r1.Value = "not Started" Then r2.Interior.ColorIndex = 24  
        If r1.Value = "Ready for PS Review" Then r2.Interior.Color = vbYellow  
        If r1.Value = "In Programming" Then r2.Interior.ColorIndex = 16  
        ' if key variables are missing then assign colour code red and prompt a message box to display the error  
        If Range("A" & i).Value > 0 And Range("G" & i).Value = "" Then  
            r2.Interior.Color = vbRed  
            cn = ActiveCell.Column  
            cr = Range("B" & i).Value  
            MsgBox ("Program Name cannot be missing at: " & cr)  
        End If  
  
        If Range("C" & i).Value = Range("D" & i).Value And Range("A" & i).Value > 0 Then  
            crn = Range("B" & i).Value  
            r2.Interior.Color = vbRed  
            MsgBox ("Programmer and validator cannot be same at: " & crn)  
        End If  
    Next i  
  
End Sub
```

Add records Button

- Creates a new record by automatically incrementing the index numbers and setting the status defaults to “Not started”.
- Previous row values are copied over to the new record The programming and validation status is set to ‘not Started’ and program name is made missing

```
Private Sub add_record_Click()  
'Add new records with common information from the previous records  
If WorksheetName = vbNullString Then WorksheetName = ActiveSheet.Name  
With Worksheets(WorksheetName)  
    xllastrow = .Cells.Find("*", .Cells(1), xlFormulas, _  
        xlWhole, xlByRows, xlPrevious).Row  
    Rows(xllastrow).Copy  
    Range("A" & xllastrow + 1).Select  
    ActiveSheet.Paste  
    Application.CutCopyMode = False  
    Range("B" & xllastrow + 1).Value = "INDEX-" & Range("A" & xllastrow).Value + 1  
    Range("A" & xllastrow + 1).Value = Range("A" & xllastrow).Value + 1  
    'Set default values to "not started"  
    Range("D" & xllastrow + 1).Value = "not Started"  
    Range("E" & xllastrow + 1).Value = "not Started"  
    Range("G" & xllastrow + 1).Value = ""  
End With  
End Sub
```

Status Button

- Pivot Tables functionality of Excel is used for graphical summary of the repository
- Status button Initiates a summary status page and refreshes the contents of the graphs based on the VB code shown

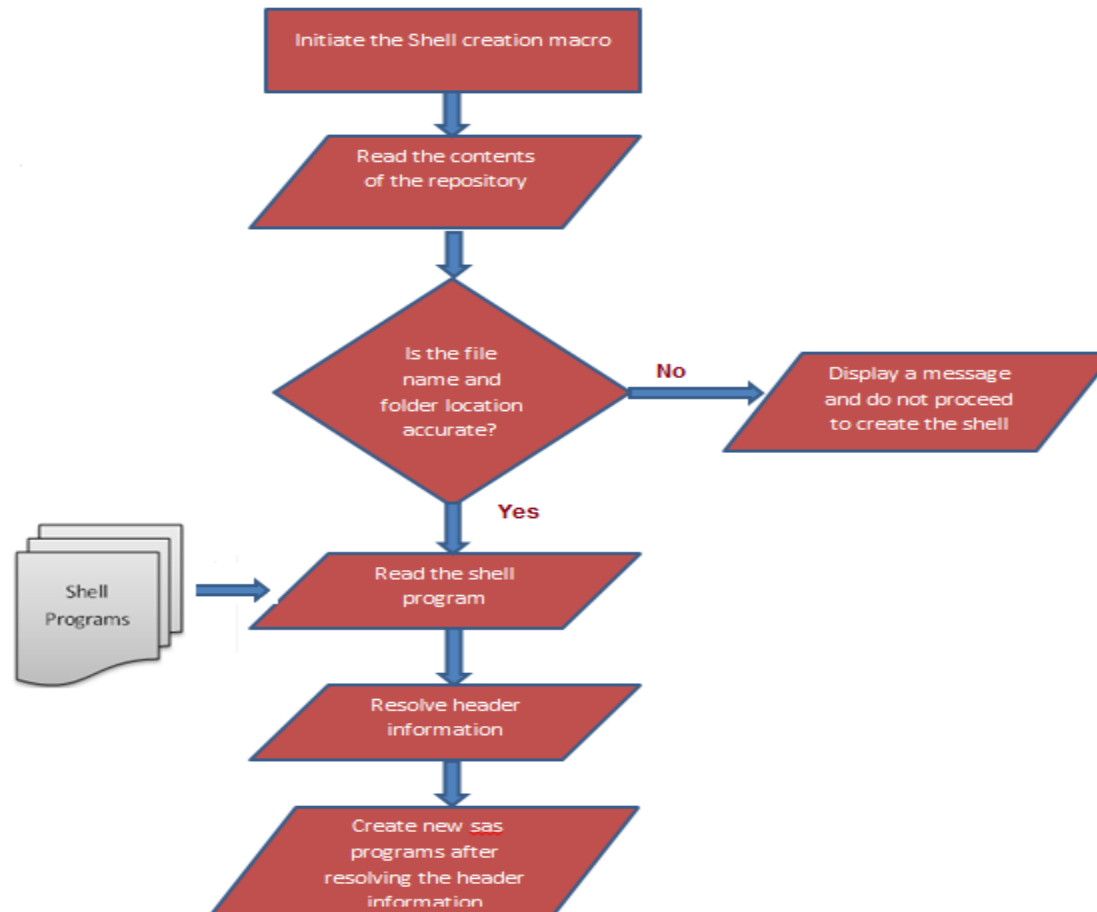
```
Private Sub Status_button()  
'Initiate the "Status" tab and refresh the tables based on the current information in the main sheet  
    ThisWorkbook.Sheets("status").Activate  
    Sheets("status").PivotTables("PivotTable2").PivotCache.Refresh  
    ThisWorkbook.Sheets("status").Activate  
    Sheets("status").PivotTables("PivotTable3").PivotCache.Refresh  
    ThisWorkbook.Sheets("status").Activate  
    Sheets("status").PivotTables("PivotTable4").PivotCache.Refresh  
End Sub
```

Generate report and return button

- The “**Generate report**” button on the status report page will trigger a prompt box, which will allow users to save the status reports in a pdf format.
- The “**Return**” button will allow users to return to the original repository front end page to make changes

```
Private Sub generate_report()  
'This converts the whole status summary sheet into .pdf file format  
Dim Ws As Worksheet  
Dim Filename As String, myShell As Object  
  
Filename = Application.InputBox("Enter the pdf file name", Type:=2)  
  
Sheets("status").Range("A4:q100").ExportAsFixedFormat Type:=xlTypePDF, _  
    Filename:="Z:\folder1\" & Filename  
  
MsgBox "Status Report Saved Successfully!"  
End Sub  
  
Private Sub return_tab()  
'returns control to the main sheet  
ThisWorkbook.Sheets("sheet1").Activate  
  
End Sub
```

SAS shell creation program flow chart



Shell Macro

- “Pipe” device type on a Filename statement creates a directory listing of all files.
- If the path is not accurate then an error message is printed out
- “Pipe” device type also returns all the file names that exist in the directory.
- This information is used to check if the file name already exists to avoid overwriting and to print a warning message

```
/*Using Pipe function check the existence of program location and duplicate program names */
Filename Filename pipe "dir &Macro_Folder_location ";
%let filefound=0;
  Data _file_exist&i ;
  Infile Filename truncover;
  Input filename $800.;
  Put filename=;
  if filename="dir: cannot access" then fileok=0;
  else fileok=1;
  /*If the file path is not correct display message */
  if fileok=0 then Message=" NOTE: Unable to open file. Please check the File path: &Macro_Folder_location " ;
  call symput ("fileok",fileok);
  /*If the file name already exists display a message and assign 1 to macro variable */
  if index(upcase(filename),upcase("&macro_program_name")) >0 then do;
    filefound=1;
    call symput ("filefound",filefound);
    Message="Message : Sorry! File &macro_outputlocation already exists";
  end;
Run;

%if &filefound =1 %then %do;
Proc print data= _file_exist&i(where =(message ne '')) noobs;
var Message;
run;

%put Message : File &macro_outputlocation already exists;

%end;
```

Why turning Smart?

- Overall, industry expectations are much higher as it relates to regulatory submission programming turnover cycle.
- Process automation is inevitable to meet the aggressive timelines.
- Reduces the necessity to have to send frequent timeline and status meetings and utilize that time for program development.
- Reduces the risk of missing the deliverables due to wrong estimation of project progress.
- It frees up time and resources. Ensures even distribution of workload among team members.
- Accuracy – The elimination of issues due to manual intervention
- It allows us opportunity to focus on analyzing the data instead of pulling it.

Future Expansion plans

- Automated header checks to improvise process compliance.
- Further expansion of shell programs for common TFLs to reduce programming efforts.
- Improvised front end capabilities such as providing direct access to programs and outputs.
- Automated Title and footnotes generation with centralized control.

Conclusion

- This dynamic utility will assist users in their ability to achieve a good understanding of the project status while improving the efficiency of the program developmental cycle and programming compliance due to process automation
- This utility is the pilot of the capabilities of the “smart program repository” automation tool, which will be incorporated into another tool being used for tracking projects, Jira. For more details on Jira, please refer to PharmaSUG 2018 Paper LD-02 by Nancy Brucken, “Customized Project Tracking with SAS and Jira.”



Acknowledgment

Thanks to Steve Benjamin (Director, Biostatistics and Statistical Programming, Clinical Solutions, Syneos Health) for his leadership, constant support, encouragement and valuable assistance in reviewing this paper.



Name: Hrideep Antony

Organization: Syneos Health

Address: 1001 Winstead Drive, Suite 200
Cary, NC 27513

Work Phone: +1 919-337-1415

E-mail: hrideep.antony@@syneosheath.com

Name: Aman Bahl

Organization: Syneos Health

Address: 5515 North Service Road, Suite 200
Burlington, Ontario L7L 6G4

Work Phone: +1 289-313-3014

E-mail: aman.bahl@syneosheath.com