

PhUSE 2008

Paper AD12

Share - A Reporting System for Efficacy Outputs using SAS® Macro

Vinod Kaippillil Mukundaprasad, Roche, Welwyn, UK

ABSTRACT

Share is a suite of validated SAS macros, developed by the programming group at Roche. The macros produce tabular and graphic data displays and range from descriptive analyses to condensations of different statistical procedures. A key feature is the filter system which allows to generate data displays on different subsets of data. All reporting macros follow the same modular design to ensure a consistent appearance and functionality.

The modular design of the macros facilitates further development. To ensure backward compatibility after a change, a suite of regression tests is executed and electronically verified before the release.

Originally this is developed for the standard efficacy reporting, but this has been very successful in some non-standard safety reporting also. An estimated 80-90% of planned efficacy analyses are covered by the share macros. Share concept has been very successful in the area of Oncology and Diabetes. Roche is now looking into extending the share concept into other therapeutic areas.

INTRODUCTION

A few years ago, the Oncology Franchise Network, a company-wide standardization initiative across functions was launched to harmonize the Oncology trial conduction. Key representatives from Clinical Science, Clinical Operations, Medical writing, Data Management, and Biostatistics formed this board, which defines and reviews standards on a regular basis.

The Oncology Franchise Network agreed on standard CRF pages for tumor assessment and a standard Oncology Analysis Plan with 17 textual and 4 graphical data display templates. This agreement was the foundation for the development of a suite of SAS® macros which we called Share.

This paper concentrates on the key features of the Share rather than describing all the modules in detail.

BUSINESS RATIONALE

As the world leader in Oncology drug development, Roche conducts numerous studies in this therapeutic area. Study data used to be captured in different ways which lead to strong customization of data analyses and data displays.

Before the development of the new macros, existing “standard” macros were copied from study to study, most of the time customized, then re-validated.

After the successful implementation of the standard Oncology Analysis Plan, project statisticians copy the applicable parts of the standard Oncology Analysis Plan into their Project Analysis Plan. All 21 data display templates in the standard Oncology Analysis Plan can be produced with the share macros.

DEFINITION OF REQUIREMENTS

The goal for the development team was to program a suite of macros, that

- is able to produce all data displays as defined in the Standard Oncology Analysis Plan.
- uses SAS® within the Roche UNIX environment.
- has sufficient built in flexibility.
- is easy to use in its standard usage.

PhUSE 2008

BUILDING A TEAM

The perhaps most important decision was to be made on the composition of the team. All members needed to work well together, yet have different characters and skills to ensure a sufficient pool of ideas. The team needed to bring in a strong leader, project management skills, drug project experience, technical expertise, and links to other Roche application development teams to capitalize on their experience.

All 10 programmers and statisticians were borrowed part-time from active drug development projects. This was a strength of the team since the drug project experience led to a user-friendly and robust product. At the same time it was a handicap to compete for resources with high priority projects.

The development team was a virtual team, operating in the Europe and US. It was an important investment to hold a face to face meeting at an early stage to develop a good team spirit.

DEFINING BOUNDARIES

Firstly, the boundaries of the share macros needed to be defined. The two key efficacy analysis datasets in Oncology are a time-to-event dataset for survival analyses and a response dataset. Since the Oncology Franchise Network left considerable flexibility to their definition, the share development team decided to not offer macros for the creation of these analyses datasets. Instead, exact specifications are given to the users to ensure that the datasets will be accepted by the reporting macros.

Another boundary decision to be made was how the macros should respond to the two major factors of variability:

- Unexpected data: The team decided to give the responsibility for the data to the user.
- Incorrect macro calls: It was decided to implement a parameter checking system which returns meaningful messages to the log before the standard SAS® error messages occur. Dependent on the severity of the issue, the parameter checking system posts alerts or interrupts SAS®.

TEAM PROCESSES

The development team held monthly teleconferences with minutes and action items. To help visualize ideas, net meeting was used for file sharing. Smaller development subgroups collaborated informally, many times capitalizing on bonds developed during the initial face to face meeting.

An important tool to mitigate potential delays was the creation and execution of a detailed project plan. It allowed to monitor the progress and inform management early about possible delays. Without a project plan, the timely development of the share system wouldn't have been possible.

INITIAL VALIDATION

The development team learned from the experiences of other, already existing application development teams at Roche. The team agreed to use a blend of existing techniques:

Each reporting macro is tested with 10-25 different macro calls using static clinical trial data. Each of the 10-25 data displays was double programmed externally. In addition, an external consultant tried "to break the macros" to verify that the macros are sufficiently robust.

The data, macro calls and corresponding data displays were stored in a version controlled system as Benchmarks for future use.

CHANGE CONTROL AND REGRESSION TESTING

All program changes are version controlled in a RAZOR® database. Changes to macros are validated and tested in a test environment by a second member of the development team.

To ensure backward compatibility, all benchmark macro calls are re-executed on the original static data. The data displays are electronically compared with the original benchmark data displays. Evidence of testing and the regression testing is kept in the RAZOR® database. The development of a script, that executed all regression testing steps was invaluable.

USER'S VALIDATION

Despite the extensive validation of the macros, the development team recommends the user to verify the correctness of the data displays. This is even more important when the data display contains primary or secondary

PhUSE 2008

endpoints. The task can be achieved very efficiently by focusing on the underlying statistical procedure and verifying the correct parameter estimates, confidence intervals and p-values.

PROGRAM DESIGN

It was critical to take sufficient time for the program design. Every team member brought valuable solutions from past programming experiences which were discussed and evaluated within the team. Sufficient planning time allowed the team to take all these “goodies” and design modular macros which paid off in all subsequent development phases. Finding the best design was an iterative process and was based on convincing the other team members of one's ideas. Controversies were welcome; In the end the development team agreed on – and even more important – stood behind modularly designed macros with classic calling programs.

A good definition of macro interfaces and macro variables helped with the consistent integration of the macro modules.

IMPLEMENTATION PHASE

All SAS[®] programmers in the development team were involved in the macro programming. The collaboration was informal.

In case of a dependency to a macro which was still under development, a dummy macro was used with an identical interface.

The creation of a user documentation was seen as a late development phase activity and was very time consuming. Especially the formatting of the document (MS Word[®]) and adding hyperlinks was challenging. This could have been delegated to a technical writer.

TESTING PHASE

After testing and validating of all macros, two waves of altogether three pilots were conducted. The idea of a pilot was to simulate the usage of the macros by a user in a non-critical environment. The pilot users were asked to produce outputs with the share system using an already reported database. A confined period of time was scheduled which ended in the completion of a questionnaire.

We found it very important, that pilot studies are not under any time pressure. It would put unnecessary stress on the pilot user and the developers and testers. Furthermore, it could lead to pressure to release the macros before they have been fully tested.

Most suggestions for improvements were made for the documentation. This shows the importance to have a very good version of the documentation available at the time of the pilots.

ROLLOUT PHASE

After the macros were moved to the production environment, they were announced electronically and in a rollout presentation at each Roche site.

Based on feedback from the pilot phase, the development team decided to rely on the documentation to train the users.

After the rollout it showed, that good documentation could not replace a face to face user training. Therefore, the share team created training material and gave face to face training at all Roche sites. In particular, the included hands-on part was very well received.

MAINTENANCE PHASE

The share team has now entered the maintenance phase. The initial development phase has ended, most of the users are trained, and the macros are in usage.

USER SUPPORT

For each time zone, the share development team nominated a super user for first line support. Their task is to operate the support email address, help users in one on one meetings and train new programmers.

PhUSE 2008

MACRO MAINTENANCE

Highest priority is bug fixing, although bugs occur rarely due to the extensive testing in the development phase. Another task is to add new functionality to the existing macros as requested by users. The decision whether to implement an enhancement request is made by the share development team. A decision to implement a change requires a wider need in the user community.

To ensure long term maintainability, it is key to be conservative with agreeing to changes. It is equally important to avoid "patching" but to invest time in finding good solutions.

All issues raised by users or the development team are documented in a database including the decision made and the implementation strategy. This ensures the knowledge transfer across different generations of developers.

Major enhancements are combined into one to two rollouts per year. Rollouts are announced in advance with a forecast of new additions. In case a non-backward compatible change can not be avoided, users are given exact specifications of the change and its consequences ahead of time.

In case, a bug is discovered, the change will be implemented immediately and communicated to the affected users electronically.

LESSONS LEARNED

- Get stable requirements supported by all functions.
- A good team culture brings out the best in every member.
- Invest in program design.
- Invest in good documentation.
- Use more than one set of test data.
- Conduct pilots.
- Plan for the maintenance phase.

BUSINESS VALUE

The share macros are now used by all Roche Oncology projects and is estimated to save an average of 50% of resources.

The defined program structure with its utility macros is now also used by project teams in their own programs. This gives project teams access to key features like the filter system and leads to a high level of consistency between outputs.

It enabled management to quickly assemble a team of programmers from different projects and sites to work on a project.

PhUSE 2008

LAYOUT OF A TYPICAL SHARE DATA DISPLAY.

Time to Event Summary by Randomized Treatment

Protocol(s): XX95001

Analysis: All Randomized Patients

	Treatment A (N=586)	Treatment B (N=586)
Patients with event	184 (31.4 %)	166 (28.3 %)
Patients without events*	402 (68.6 %)	420 (71.7 %)
Time to event (days)		
Median#	308.0	351.0
95% CI for Median#	[275;337]	[308;427]
25% and 75%-ile	183;429	206;520
Range##	1 to 534	1 to 560
p-Value (Log-Rank Test)	0.0676	
Hazard Ratio	0.82	
95% CI	[0.67;1.01]	
1 year duration		
Patients remaining at risk	21	28
Event Free Rate	0.38	0.48
95% CI for Rate	[0.30;0.46]	[0.40;0.56]

Time to event (days) (TTEVENT) - Censoring: Event (0=censored, 1=event) (CSEVENT)

* censored

Kaplan-Meier estimate

including censored observations

Program : /home/biostat/muehliga/cdp90501/xx95001/p2ex1.sas

Output : \$HOME/cdp90501/xx95001/reports/p2ex1.lst

31MAY2007 16:18

Page 1 of 1

PhUSE 2008

THE DATA DISPLAY HAS BEEN PRODUCED WITH THIS MACRO CALL:

```
%onco_km_cox ( dsin = ana.event,
               timevar = ttevent,
               censvar = csevent,
               censval = 0 ,
               timepoint = 365 ,
               timelabel = 1 year duration ,
               strata = ,
               covariables = ,
               ties = breslow ,
               decimals = 0 ,
               alpha_med = 0.05 ,
               alpha_km = 0.05 ,
               alpha_cox = 0.05 ,
               rincens = Y ,
               demo = anar.demoext ,
               rx = rnd ,
               groupstyle = rx ,
               rxfmt = ,
               rxordfmt = ,
               rxreference = ,
               byvar = ,
               byvarfmt = ,
               byvarordfmt = ,
               style = LOG-RANK SPANBYVAR KCNUMEV KCMED KCHR KCDUR ,
               titles = contents.txt,
               footnotes = footnotes.txt,
               blockformat = sortform ,
               debug = N ,
               options = ,
               doctype = LST PDF,
               pagesize = SASP8 ,
               savedataset = QA ,
               endpgm = );
```

Please note, that most of the macro parameters have meaningful defaults. The user typically needs to define 10-15 macro parameters.

The macro parameters for other share data displays are identical unless they control a table specific functionality. For example, the parameter rx= always contains the treatment group variable.

PhUSE 2008

HIGH-LEVEL PROGRAM FLOW OF A TYPICAL SHARE MACRO

```
%macro shareexample (dsin= ,  
    .... );  
  
    %util_paramchk;    /* Checks for unexpected parameters */  
    %util_initpgm;     /* Initializes macro variables etc. */  
    %util_initdata;    /* Applies filter system and calculates N */  
    <table specific data manipulation steps>  
    %stats_lifetest;   /* Contains statistical procedure and returns  
                        dataset created with ODS */  
  
    <more table specific data manipulation steps>  
  
    %util_titles;      /* Assigns titles from central file */  
    %util_footnote;    /* Assigns user defined footnotes from central  
                        file */  
  
    %util_paginate;    /* Simulates proc report and flags desired page  
                        breaks in dataset */  
  
    %util_odssetup;    /* Selects output destinations */  
  
    proc report;  
        ....  
  
    run;  
  
    %util_valdataset;  /* Saves dataset for double programming */  
    %util_endpgm;     /* Closes ODS, prints page numbers */  
%mend shareexample;
```

PhUSE 2008

A SELECTION OF COMMON FEATURES OF THE SHARE MACROS:

FILTER SYSTEM.

All share data displays can be produced on user specified subpopulations (filters). This feature is very useful for exploratory analysis.

The subpopulations are defined in a permanent filter dataset of the following structure:

<i>fltnumb</i>	<i>fltwhere</i>	<i>flttitle</i>
S	WHERE SAFETY='YES'	SAFETY POPULATION
FEM	WHERE SEX='FEMALE'	FEMALE PATIENTS
ELD	WHERE AGE > 60	ELDERLY PATIENTS
...	...	...

Fltnumb: unique identifier of subpopulation.

Fltwhere: Subsetting where clause for subpopulation.

Flttitle: Subtitle describing subpopulation.

In the Roche UNIX system, programs are submitted in batch using the script `rsas`: E.g. `rsas tableprog.sas`. `Rsas` will create a log file `tableprog.log` and an output file `tablelog.out`.

Optionally, a program can be submitted with a filter number passed to `rsas` using the `-suffix` option: `rsas -suffix S tableprog.sas`. The files `tableprog_S.log` and `tableprog_S.out` will be created. The addition of the filter ID to the log and output file ensures, that files are not overwritten when the same program is submitted with different filters.

When the suffix option is used, a utility macro inside the share macro reads the record from the filter dataset that matches the suffix argument. It subsets the demo dataset using the where clause before the merge with the analysis dataset. The filter title is stored in a macro variable to allow dynamic subtitles.

Filters can be combined and are executed sequentially. E.g. `rsas -suffix S_FEM tableprog.sas` will execute the reporting program `tableprog.sas` on female patients who are also in the safety population.

STYLES OF BY PROCESSING

Most of the tabular share macros allow by-processing. A categorical variable can be passed to the macro parameter `byvar=`. To allow a flexible usage of the space on the output, three different layouts are offered to arrange the by-categories:

- Spanned byvar splits the by groups horizontally with spanned headers.
- Block byvar places the by groups vertically.
- Page byvar is similar to block byvar but forces a page break for each by group.

The user defines the preferred by grouping style as a keyword in the macro parameter `style=`.

CENTRAL FILE FOR TITLES AND FOOTNOTES

Titles and user defined footnotes are stored in two central text files. A utility macro inside the share macro picks all records which match the output file name. The asterisk (*) serves as a wildcard to allow the usage of the same title for different outputs.

PhUSE 2008

```
enttar1_01***** &draft Listing of Non-Target Lesions
etarr1b_10***** &draft Summary of Target Lesions at Baseline
*****
***** Protocol(s): XX95001
*****
***** Analysis: &_ufilttitle1
```

Please note that the system allows macro variables.

The macro variable _ufilttitle1 contains the filter title and will be displayed for every output.

The macro variable draft is defined in a local autoexec and resolves to DRAFT during the conduct of the study and is set to blank after database closure.

The footnotes text file looks identical.

OPTIONAL VALIDATION DATASETS

When users choose to double program an output, getting a 100% match by electronically comparing the final outputs can be very time consuming. The share macros offer to make a permanent copy of the final dataset which is used by proc report. This allows the use of proc compare.

CONCLUSION

The development of an application with a long lifespan can be a very satisfying project. By bringing programmers and statisticians with different backgrounds together, the whole can be more than the sum of individuals. The share development team developed a suite of efficacy reporting macros which are – due to their modular design – easy to maintain and is creating widely recognized business value.

CONTACT INFORMATION

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

Vinod Kaippillil Mukundaprasad

Roche Products

6 Falcon Way,

Welwyn Garden City, England, AL7 1TW

vinod.kaippillil_mukundaprasad@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.