PP19: Industry Standard Good Programming Practices Guidance V1.0 now available

The PhUSE GPP Steering Board

Most companies have individual guidance in place defining best practices for the development of high quality and efficient programming code. The Steering Board for Good Programming Practices in Health and Life Sciences is a voluntary industry group with representatives from a diverse array of health and life sciences organizations working to develop common good programming practices applicable to statistical programming for the analysis and reporting of clinical trial data as well as for data integration or preparation of clinical data for e-submissions. We recently achieved our first goal and released the first version of a consolidated guideline which potentially could replace those documents created and maintained by each organization and could also serve as a set of business rules when working on standard scripts for public use. The GPP PhUSE Wiki remains open for further public comments. The GPP Board facilitated discussion clubs at the PhUSE annual conferences 2012 and 2013 where significant content and process related input to the guideline was gathered.

This is what the GPP Wiki looks like

http://www.phusewiki.org/wiki/index.php?title=Good_Programming_Practice_Guidance

This is also the place to share your ideas and post your contributions.
Please provide your comments!

Contents

1 Published Guidance Document
2 Introduction
3 Getting Started With a New Project
4 Language
5 Program header
5.1 Revision history
6 Comments
7 Naming conventions
8 Coding conventions
8.1 Log File Checking
8.2 Portability
8.3 Hard coding

1 Published Guidance Document

Version 1 of the guidance has been published …
… public comment and comments will be incorporated in periodic revisions so please continue to review and comment on the wiki.

2 Introduction

This document provides guidance for good programming practices (GPP) for data manipulation, analysis and reporting of clinical data in health and life sciences organizations.

GPP addresses the way in which you write code and comments. The following key principles of Good Programming Practice are followed in this document.

- Write code that is clear and easy to read and review
- Write code that is easy to maintain, modify and debug
- Use robust programming to reduce the need for code maintenance
- Develop code that can be reused or easily adapted.
- Write code that uses minimal computer processing resource (Efficient code)
- Write code in such a way to reduce logical and syntax errors

This guidance is primarily aimed at SAS programmers, however the principles of GPP also apply to other languages such as R and Stata. In addition, although this is not produced with SAS macros in mind, the same principles apply to macros too.

We often have to update existing programs to add new rules, copy programs from one study to another, and take over programs written by others. This guideline aims to show how to produce well structured and well documented programs so that they are easy to read and maintain over time. It is meant to be applicable to all programs, and hence all programmers regardless of experience. Specific rules may be of more use to novice programmers, but applying the principles should be in mind for experienced programmers and mentors.

8 Coding conventions

In order to be efficient and streamline the sharing of program code between programmers, with regulatory agencies, and with external partners or vendors, it is vital for code structure to follow standard conventions. SAS code which follows these conventions is much easier to read, modify, maintain, and correct. These conventions are divided into those which should be considered as required, and those which are merely recommendations to be followed as applicable.

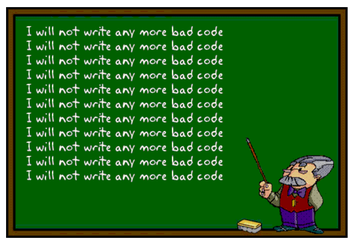
Required conventions

- Do not overwrite existing datasets
- Each organization may have its own standards for using case within programming code but use of all uppercase should be avoided
- Separate data steps and procedures with at least one blank line
- Use 'data=dataset' option in procedure statements so that the dataset being used is explicitly stated to ensure that the statement will work if it is moved to another location
- End data steps and procedures with run or quit to provide a boundary and allow for independent execution
- Split data steps into logical parts
- Put each statement on a separate line
- Left justify global statements and data and procedure statements and their corresponding run and quit statements
- Indent statements belonging to a level by 2 to 5 columns (use the same number of spaces throughout the program), i.e. every nesting level should be visibly indented from the previous level.

Have a look at the Wiki, there is more …

What is next?

We would like the GPP guidance document to continue evolving over the next few years. We want to avoid a document that is so long and detailed, that it is not practical to use, but we welcome suggestions for updates, additions and improvements. For example, the section on defensive and robust programming could be extended.



The PhUSE Wiki will stay continuously open for further recommendations and additions.

During the upcoming months the GPP board will
- Gather GPP V1.0 implementation experiences
- Continue collecting input and recommendations with regard to the existing or new sections of the GPP guidance
- Enhance and maintain the GPP guidance document and create new versions as needed

Get involved!

- ❑ **Add a new wiki page on a GPP topic**
  (Put the text [[Category:Good Programming Practice]] at the end of the article to enable searching for GPP)
- ❑ **Make updates to existing pages**
- ❑ **Add to the talk page for the wiki, as shown in the example below:**

7 Naming Conventions

I think it would be universally accepted not to have space characters in program / output file names.

Re: 7 Naming Conventions – – Mfoxwell (talk) 03:14, 2 January 2014 (CST)

Agree, spaces can cause extra work and confusion.

- ❑ **Get involved with the GPP board - we are welcoming new members**

GPP Steering Board*	
Mark Foxwell, Chair	Cmed, UK
Arthur Collins	Biogen Idec, US
Beate Hientzsch	Accovion, Germany
Shafi Chowdhury	Shafi Consultancy, UK
Dean Grundy	Roche, UK
Jennifer Chin	Eisai, UK
Ralf Gessenich	Grünenthal, Germany
Cindy Song	Sanofi, US
Jean-Marc Ferran	Qualiance, Denmark

*Established in 2008 after the PhUSE annual conference

http://www.phusewiki.org/wiki/index.php?title=Good_Programming_Practice_Steering_Board

GPP Discussion clubs at PhUSE annual conferences

Delegates were invited to an interactive discussion on how we ensure well written, efficient code.
The interest was overwhelming and even though the room was too small, the group showed real flexibility and started an engaged discussion in 3 different sub-teams as presented below:

Macros – a powerful tool to use and abuse

For what tasks do you commonly use macros?
- repetitive tasks, generic macros (dynamic), specific functions (across programs), ensure consistency, replacement of using arrays, reduces maintainability

What criteria do you use when deciding that a task or problem is worth putting into a macro?
- same code to be used in several studies, looping

[…]

GPP applied in daily routine

[…]

Management view on GPP: what we give and will get in return

[…]

- The vast majority of people confirmed that they would be willing to withdraw their company specific guidelines in favor of an industry-wide standard.

General comments and agreements

The majority of attendees did express an interest in applying a standard GPP guidance once available.
As the FDA/PhUSE CSS Working Group 5 is working on "Development of Standard Scripts for Analysis and Programming" it was agreed to liaise with the working group to identify possible overlaps and common topics.

After the encouraging discussion club in Budapest and with the first version of the industry standard GPP guidance available, the discussion club focused on
- Implementation objectives and strategies
- Costs and benefits of using/not using GPP
- Review of the GPP naming convention section

A few highlights can be found below:

Implementing the cross-industry PhUSE GPP

Questions: Would your organization use the GPP industry standard? Use the PhUSE GPP as a minimum standard? What are the advantages/disadvantages of the PhUSE GPP versus own Guidance / Will GPP raise standard?

- Good as general base/minimum standard but still some disagreements, easier for service providers /onboarding, easier to audit, use of GPP ensures/encourages compliance, reach a higher level of standard

Conclusion

- The need for a GPP guidance is commonly agreed
- It took quite some time, but with the help of many the first version of the GPP guidance has been developed
- Unsurprisingly, there are different opinions on some details of the content
- There is room for improvement
- Implementation experiences are expected to support further enhancement
- Working in a cross-industry voluntary group is a great experience and can lead to notable results, but it requires patience

GPP Survey Results (2013)

The respondents represent a range of organizations, including pharmaceutical companies, biotechnology companies, CROs and consultancies. 68% of respondents had a GPP document to follow such as a guideline or SOP and of these, 80% were mandatory.

Here are some highlights of good programming practices which respondents indicated that they used and felt were either important or very important:
- 82%: ending steps with run
- 78%: explicitly referencing input dataset in each step or procedure
- 77%: descriptive comments in each section of the program
- Others include breaking lines in a neat and logical way and placing only one statement per line

Study on Good Programming Practices in Health and Life Sciences
Summary Part 3 - Programming Practices Ranked by Mean Importance
Page 1 of 2

Programming practice	Mean Importance (0=Not Used - 4=Very Important)
Program Layout - Use of standard header	3.4
Program Flow - Consistently ending steps with run and quit where appropriate	3.2
Program Flow - Program is logically organized	3.2
Program Flow - Consistently explicitly referencing input dataset in each step or procedure	3.1
Program Layout - Descriptive comments in each section of the program	3.0
Program Flow - Consistently placing only one statement per line	2.5
Program Layout - Consistent use of appropriate indentation on each line	2.4
Program Flow - Breaking lines in a neat and logical way	2.4
Program Layout - Consistent use of appropriate spacing between statements and sections	2.3
Standard Naming Conventions - Standard naming convention for macros	2.0
Standard Naming Conventions - Standard naming convention for formats	1.4

SOURCE: DRUG/STUDY/SMART1/GPP-SUM3.SAS DATE: 30AUG2013

For the GPP Steering Board: *Beate Hientzsch*
Helfmann-Park 10
65760 Eschborn, Germany
beate.hientzsch@accovion .com

Special thanks to the Statistical Programming Team at Accovion for putting this poster together!