

Paper CT-19

Minimum Level of Documentation for Ad Hoc Report Programming

Robert McCurdy, Rockmart, Georgia

ABSTRACT

In an ad hoc reporting group, unfortunately documentation tends to slide. When under pressure to get a new report out, it is advantageous to be able to quickly identify and locate previously written programs which have similar features. This paper explores minimal documentation which facilitates this process. A work sheet for each program specifying databases, tables, variables, algorithms and called routines not only helps in program development but also provides the basis for the development of an external matrix showing features versus programs. This matrix can help quickly identify which programs have the needed features when beginning a new project. Minimum level of documentation inside the program including the flower box is explored.

INTRODUCTION

The classical program development cycle includes needs analysis continuing with systems analysis, design, coding and testing. Ad hoc program development rarely follows this life cycle development. The proposed method of documentation for ad hoc program development is to create a spec sheet where classical program development methods can be incorporated in an abbreviated form. Referencing each item in the spec sheet to a specific location in the program is important as it helps facilitate program review and possible future modifications.

Using this ad hoc documentation method on several programs will eventually enable the programmer to develop a features matrix connecting the programmer's arsenal of programs and important features contained in the programs. Not only will this be a rapid reference tool but it may dazzle coworkers impressed with your familiarity with your collection of programs.

THE REQUEST

A developer receives a request through email containing selection criteria and other pertinent information. Here is an example:

The Marketing Group is planning to do a telephone survey to determine the interest level for using solar as part of an individual household electrical supply. Needed is a sample of 10,000 customers meeting for following qualifications:

- 1) Residential
- 2) Household income of at least \$75,000
- 3) Average monthly power usage of at least 2500 Kwh
- 4) Live in a metro area

For each person in the sample I need the following information:

- 5) Name
- 6) Meter address
- 7) Telephone number
- 8) Age
- 9) Solar user (Y/N)
- 10) On flat billing

Figure 1 Sample email request

ANALYSIS AND PROGRAM DESIGN

Note that in the items in the request fall into two categories:

- A) Selection criteria (items 1 - 4)
- B) Needed information (items 5 – 9)

The first step is to develop a specification sheet (spec sheet) with three columns (more columns will be added later).

The first column will be a sequential item number (for purposes of this paper they will match the item numbers in the email).

The second column will be a flag indicating whether the item is

- Selection criteria (denoted with “S”) or
- Information (denoted with “I”).

The third column will contain the requested items as specified in the email.

After filling in the first 3 columns the spec sheet will appear as in figure 2.

Spec Sheet			
Item	Flag	User request	
1	S	Residential	
2	S	Income level	
3	S	Power usage	
4	S	Metro	
5	I	Name	
6	I	Address	
7	I	Phone Num	
8	I	Age	
9	I	Solar user	
10	I	Flat bill	

Figure 2 Spec sheet

As programmers and analysts know, now the work begins - finding the needed items to implement the solution. Identification of data includes: databases, tables, variables, values for the variables, and specialized routines if needed. To accommodate these, more columns are needed in the spec sheet. The concept of the spec sheet is to have a simple reference. In the interest of simplicity abbreviations are recommended as shown in parenthesis.

- Block (BK)
- Database (DB)
- Table
- Variable (Var)
- Values
- Footnote (FN)

Figure 3 shows the new heading with the columns to be added:

Spec Sheet								
Item	Flag	User Request	BK	DB	Table	Var	Values	FN

Figure 3 Spec sheet

Data discovery time is a luxury that programmers don't always have when starting a new project. Large organizations have a lot of databases and tables to explore. Additionally, a neophyte programmer or even a seasoned programmer new on the job can be challenged to find source locations for all of the data required on a new report. Imagine walking through the office trying to network with others about where to find data. This is one scenario where having a printed copy of a spec sheet will be helpful to use as a visual aid during conversations.

The spec sheet can also help to fill in gaps that the requester may have overlooked. As an example - item 11 in the spec sheet - the "do not contact list" was not mentioned in the original email request.

The more detail included on the spec sheet, the more useful a tool it will become. When there is not enough room in the variable (var) column and/or in the values column for adequate explanation, place the explanation in a footnote. Footnotes should be referred to in the FN column. Called programs, special procedures, or special code can be referred to in footnotes. If the data resides in other than a SAS® table show the path to the data in a footnote and specify any PROC's or other special code needed to access the data. The column labeled (BK) will be explained in a later section. A sample of a completed spec sheet (except for the BK column) is shown in figure 4.

Spec Sheet								
Item	Flag	User requirement	BK	DB	Table	Var	Values	FN
1	S	Residential		CQ	bill	tartyp	1	
2	S	Income level		MON	monthly	income	7 or above	
3	S	Power usage		Use	usage	kwh_nn	ge 2500	1
4	S	Metro		CNTY	cnty_cd	county		2
5	I	Name		CQ	bill	name	as shown	
6	I	Address		CQ	bill	serv_ad		3
7	I	Phone num		CQ	bill	phone		4
8	I	Age		MON	monthly	age		
9	I	Solar user		CQ	bldg	tariff	u1,v1	
10	I	Flat bill		CQ	bill	Flat_flag	y	
11	S	NO CONTACT		CNT	contact	nocontact	nc,nmnc	
Footnote 1: average the past 12 months where nn=01...12								
Footnote 2: select only metro area counties								
Footnote 3: %inc 'parseAD'								
Footnote 4: %inc 'parsePH'								

Figure 4 Completed sample spec sheet

CROSS REFERENCE BETWEEN SPEC SHEET AND PROGRAM

In order to link the items in the spec sheet to a location in the program, use the program block structure that is inherent to SAS programming. SAS is composed of PROCs and DATA steps. A block can be as short as one PROC or one DATA step or a block can consist of several PROCs and/or DATA steps. Each block should have a specific function (or a limited number of functions with similar characteristics). The blocks should be numbered in sequential order as shown:

```
*Block 1;
*Comment stating the purpose of the block;
Lines of SAS code
*End of Block 1; (optional)
```

```
*Block 2;
*Comment stating the purpose of the block;
```

```
Lines of SAS code
*End of Block 2; (optional)
```

The BK column provides for the referencing of blocks of SAS code in the spec sheet. The number of the block which satisfies a requirement in the spec sheet (or contains the solution to an item in the spec sheet) can be placed in the block column in the spec sheet. For example if the SAS code within block 1 satisfies the residential requirement then the number 1 is placed in the BK column beside the user requirement for residential.

When this process is complete, each item in the spec sheet refers to a block of SAS code in the program. This is very useful in that it provides a guide for program review. Additionally, this method of documentation acts as an index into the program code elements, enabling any programmer to quickly familiarize with the existing code and work on program modifications.

An alternative to using blocks is to use line numbers (the enhanced editor provides line numbers). The disadvantage of using line numbers is that when making modifications to the program (adding or deleting lines) the line numbers have to be remapped into the spec sheet. The advantages of using block structure referencing seem to outweigh the use of line number referencing.

FEATURES MATRIX

Eventually, a programmer will have a collection of ad hoc programs. It is advantageous to be able to quickly look up old programs that have the same or similar features which can be achieved through a features matrix. This matrix can be built as a two dimensional matrix: program features versus the program titles. Suggested features include databases, procs, specialized SAS code and macro included programs. This format is not meant to provide detailed information such as lists of tables or variables. However, if a table or variable is seldom used and/or its use is very specialized, an exception should be considered.

Features can be listed in the first column and the names of the programs across the top. For each program that uses a feature place a dot or some symbol at the intersection of the row and column. An example is shown in figure 5.

	S O L A R	B U I N E S S A L	I N S T R A G	L I H T I N G	D E A L E R S	F A M I L Y H	W A T E R	P A R L E S S	
CQ	X	X	X			X	X	X	
USE	X	X	X			X			
CNTY	X	X	X	X	X		X	X	
CNT	X	X		X		X	X	X	
%inc 'parseAD'	X	X	X		X			X	
No contact code	X	X	X	X	X		X		
Ave power code	X	X						X	
PROC IMPORT	X						X		
PROC EXPORT	X	X	X	X	X	X	X	X	

Figure 5 Features matrix

FLOWER BOX

An effective flower box at the top of the program, will provide the user of the program with an orientation as to what the program does, lists any possible changes that may need to be made to the program prior to its execution and any preparation that need to be made to the data. The flower box should contain the following items:

- Name of the program
- Brief statement of objective
- Path where the program is stored
- Path where the data is stored
- Input
- Output
- Prerequisite programs which need to be executed (if any)
- Called programs (if any)
- Data preparation (if any)
- Hard coded dates or other parameters which may need to be changed (show block numbers)
- Author
- Date written

An example flower box is shown as figure 6 (not all suggested item are included):

```

/*****
/*
/*          Program:  Solar and Green                      */
/*
/*          Objective:  Produce a survey list of 1000      */
/*                      qualifying for solar energy use      */
/*
/*          Program Path:  S:\ad hoc programs\Solar and Green\Programs */
/*
/*          Data Path:  S:\ad hoc programs\Solar and Green\Data      */
/*
/*          Input:  CQ.bill
/*                 MON.monthly
/*                 Use.usage
/*                 CNTY.cnty
/*
/*          Output:  S:\ad hoc programs\Solar and Green\
/*                  Data\Solar Survey.xls
/*
/*          Called Programs:  parseAD
/*                           parsePH
/*
/*          Author:  Robert McCurdy
/*
/*          Date:  13AUG2012
/*
*****/

```

Figure 6 Sample flower box

CONCLUSION

While it may seem like extra work, be assured that the time spent over the entire development cycle of a project will usually be shortened. By taking a few minutes to complete the spec sheet, you will have the needed information documented so that you can develop the ad hoc program in an organized manner; which will usually save time in the long run. Spending a few minutes after the completion of each program to place a column in the features matrix will pay dividends when starting a new project. It will make it possible for you to answer that frequent question "Where have I done this before?"

CONTACT INFORMATION

Robert McCurdy

Cell: 717-830-0891

Home: 770-684-9863

Email: RNMCCURDY@aol.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 registered trademarks or trademarks of their respective companies.