

Interviewing and Assessing SAS Programmers

Neil Howard, Independent Consultant, Charlottesville, VA

Abstract

You are interviewing a candidate with a sharp navy suit, an impressive resume, ten years SAS experience, who knows his/her strengths and weaknesses and where he/she wants to be in 5 years. But how do you tell if this ten years of experience is really ten years, or the same year over and over again? If, indeed, past behavior (and/or past job performance) is the best indicator of future performance, how are we, the SAS community, judging and being judged as job applicants. What are assessment criteria?

The author developed a survey tool administered to selected SAS programmers and managers to see if the tricks of our trade can be determined. What techniques are being used to interview and assess SAS programmers and their productivity? This paper will explore the results of this survey, looking at details in the following areas: assessing your needs as a hiring entity, developing probing technical SAS questions, identifying resume red flags, conducting useful phone interviews, proficiency testing, code walkthroughs, metrics, standards, and quality control.

The Survey Questionnaire

Disclaimer: The author is not an expert in survey design or analysis.

The topic was driven purely by curiosity and a desire to gather information on effective interviewing in the SAS community. I have conducted hundreds of interviews and been interviewed many times for jobs or by clients. And having been involved in the SAS community for twenty years, I noticed the concern over finding good people and evaluating technical experience. SAS programmers are at a premium, but how do you know you are getting the talent you need? There have been complaints from managers regarding the lack of information available to aid them with the SAS component of an interview. But, most interestingly, there have been comments regarding an overall ineffectiveness in assessing skill level.

The survey questions and categories were derived from: 1) literature on hiring and interviewing, and 2) practical experience (mine and others). The section on SAS specifics was based on the input of experienced SAS programmers, managers and instructors.

The survey tool was critiqued by several experienced SAS users and/or managers, and the first thing that was clear was the bias I injected into the survey, despite my goal of objectivity. The SAS questions, in particular, reflected the importance I place on understanding the internals of SAS (compile, execute, PDV), debugging and testing, and DATA step processing. The needs of hiring statisticians, heavy macro programmers, users of AF, FSP, or graphics, and other products, would not be met completely. The survey did not address operating systems and interface issues. Not enough attention was given to open-ended questions that probed for problem-solving skills and style of approaching tasks.

The survey was sufficient enough to elicit good feedback and generate discussion. The results indicated how differently

everyone approaches the interview process and defines a successful hire or productive programmer. Most reviewers felt the sections were well covered.

Additional questions on metrics and coding standards were added tangentially just to test the waters. The general flavor of the feedback given warrants presentation. And the overall lack of feedback indicates these are touchy, and often untouched, areas.

The Resume

Resumes are like mirrors in a fun house and learning to decipher them is an art. "Providing support to Apollo scientists" may mean "coffee gopher". Ten years programming in SAS for one company may mean the same year over and over again or represent tremendous growth from running ad hoc reports to large systems design and data warehousing.

Technical inaccuracies were red flags for respondents: misspellings of hardware or software products, misuse of jargon, and misstatements in the description of applications indicate misunderstanding of usage and capabilities. Other red flag issues for respondents included gaps in employment history, fluff descriptions instead of concrete detail, and the inability to substantiate the experience claimed.

If the job requires client management where communications and presentation skills are paramount, the actual appearance and format of the resume is important to our interviewers. They look for good organization and effective explanation of technical issues. This was also important when documentation would be a large component of the job.

In general, the respondents expect a neat, accurate resume that demonstrates technical growth, career advancement, evidence of accomplishments, appropriate descriptions of technical material, and a clear understanding of technical issues (in gradients, depending on the nature of the position).

The Phone Interview

Most respondents reported using a telephone interview administered by technical staff to pre-screen applicants. They are looking for number of years experience, attitude, and communications skill (telephone manner). Though stating it in many different ways, the respondents were looking for "chemistry" and to get a "sense of the person". Turn-offs in a telephone interview are lack of enthusiasm, bad mouthing former employers, blaming and excusing.

The General Interview

The form and content of an interview varies among the respondents. In general, we are usually working from a specific job description; several members of the technical team get a chance to meet the candidate. The team leads or managers do the reference checks, not non-technical human resource personnel. Most respondents report they have not

been trained in interviewing techniques or technical interviewing.

The SAS skills of the candidate are evaluated primarily through: 1) standard (routinely used by the group) or ad hoc technical questions (driven by the resume), and 2) code samples. Most respondents had never used or received a SAS proficiency test. Even fewer had devised such a skills test for in-house use.

Since past performance is the best indicator of future performance, it is important to get this part right. Clearly, the most popular tool is sample code; and most respondents walk through the code with the applicant to determine if: 1) they wrote the code, 2) they understand it (didn't copy it or use a shell program), 3) they can defend it, 4) they can explain why certain design methods and coding styles were used, and 5) they can explain how they know the program produces the intended results. Coding standards and adherence to SOPs are a plus in sample code.

Not all programmers are willing to bring code samples out of respect for their employer's or client's confidentiality. And there is often reuse code in large shop programs. Some fear their code will be copied. It was agreed that serious candidate may want to write some generic programs that demonstrate his capabilities.

Typical questions posed by the respondents detailed a problem requiring that the candidates demonstrate their approach to problem solving and how they apply analytical skills. The interviewers are not looking for a right or wrong answer, rather a demonstration of how they think and pursue solutions, what happens when they get stuck or find problems with the data.

The SAS Specific Interview

The responding interviewers typically ask questions that demonstrate an understanding of:

- ◇ data step internals (compile, execute, PDV)
- ◇ read/write of SAS and non-SAS data
- ◇ declaratives
- ◇ arrays
- ◇ fuzzy merges (IN= variables, BY groups)
- ◇ data _null_

Fewer asked questions about:

- ◇ modularity
- ◇ macro
- ◇ PROC vs. DATA step solutions
- ◇ innovative/inventive techniques/solutions.

Almost all of the respondents asked candidates to describe in detail a specific programming task they accomplished or explain a complex macro application. The objective is to find out what the programmer knows and if they know how to explain what they know.

Few respondents asked about testing, validation and debugging techniques. Those who did were fierce on the issues. Debugging and testing constitutes as much as 85% of what is considered "programming". Knowledge of syntax is just a small part; usage and understanding of functionality is important; experience is critical, and knowing how to get questions answered is even more important. A programmer should be able to demonstrate that he understands how to gather requirements, program from them – mapping to code,

and validate the results, using special features of the SAS system. Probe for their habits, methods, and attitude regarding deliverables and how they feel about program "checkers" or QC review or team walk-throughs.

Appendix A contains categorized interview questions obtained from SAS-L contributions and various web sites.

Intangibles

Although hiring is not all about "gut feelings", a majority admitted it is a part of the equation. Interviewers are looking for certain personality traits. Not all programmers are computer science majors; many have music, social or physical science backgrounds and pursue challenging methodical, analytical activities like puzzles, games, etc. Companies are looking for professionalism tinged with a sense of humor, curiosity, self-motivation, flexibility, and creativity.

Important, too, is the ability to work in a team environment effectively, handle pressure, work against deadlines, and communicate effectively with clients. Respondents often ask for descriptions of business relationships with peers, clients, bosses. They prefer candidates who aren't judgmental and don't act like know-it-alls. Although some hire for potential, most prefer highly motivated, interested employees. Asking good questions reflects well on an interviewee.

Metrics

A few questions regarding programming metrics and measures of productivity were included in the survey just to test the waters. It is clear from the few answers that it is hard to quantify and qualify what programmers and analysts do. It is also clear from the lack of response that this is a touchy, often untouched, subject. It warrants further industry research and more data.

The managers among the respondents are often asked to devise metrics for their group. Non-technical senior management is used to seeing numbers. "Happy clients" won't satisfy this mindset; but it proves difficult to apply metrics. Productivity measures most often mentioned are timeliness of deliverables, performance against deadlines, end-user feedback, accuracy of status reporting, ability to solve problems.

Respondents also referred to: ability to take feedback, success in teamwork, adherence to specs, self-motivation, knowing when and how to ask for help, repeat business, follow-through, willingness to share and mentor, being innovative in attacking tasks.

Several mentioned adherence to SOPs and programming standards, begging the issue of how many shops use these tools. Some use strict code standards for readability and maintainability, checklists, guidelines, and keep complete validation folders. Questions on QC and validation revealed that accuracy of results was an important, yet many did not use program checkers or QC reviewers. The measure was client acceptance.

Number of lines of code written was dismissed almost entirely as a useful metric for programmers. Comments indicate it could promote hackerism. Programmers would not be striving for the most efficient solutions and they would not be motivated to document their programs or maintain progress reports.

Programmers often claim the number of hours worked is an effective measure. Managers were looking for working smarter, not longer.

One most interesting comment made on the survey indicated that reliance of metrics is used as a substitute for good technical managers who know how to manage and motivate technical employees.

Crazy Interviews

Interviews can be highly unpredictable and respondents reported some unexpected experiences: the candidate who filed her nails throughout the interview; the one who brought her badly misbehaving nine year old; the one who brought his lunch and munched a Big Mac and french fries; the one who leaned back in his chair and put his feet up on the desk; and the one who burst into song at an informal lunch interview -- it was appropriate to the moment and lent a definite air of camaraderie to the gathering, viewed positively in context.

Another interviewer asked a candidate was the LAG function did, to which the candidate replied: "Is that the way you do things around here? Ask questions you already know the answers to?" Or a candidate's reaction to an interview: "Why are you asking me all these technical questions?"

One respondent was very impressed by a candidate who brought code samples that were color coded. This could be interpreted as good organizational skills, nice presentation, attention to detail, or perhaps too much attention to detail. Another manager always likes to throw in one totally off-the-wall, totally unrelated question, like "If money were no object, what would you like to do?"

As interviewees, the respondents reported mostly disturbing experiences. One endured a manager who made derogatory remarks about the person they'd be replacing and continued by bad mouthing several other current employees. Another had an interviewer make it abundantly clear what kind of person he would and wouldn't work with -- very illegal.

Proficiency Testing

Several respondents reported interviewing from a "standard" list of questions devised in-house, and some have been shared at previous SUGIs or on SAS-L. SAS-L users were recently notified of a proficiency test for SAS programmers at www.tekmetrics.com. The beta testing is strictly multiple choice, and draws conclusions from your score. "The test was somewhat interesting (but slow) to take, but also illustrated some of the problems with competence testing. There were ambiguous questions. There were poorly worded questions. There were questions which, strictly speaking, didn't have any correct answers. There were questions where not enough information was given to determine a complete answer. There were questions which appeared to have typos. There were answers which 99.9% of all SAS users will never care about."

(a SAS-L contributor) Most of the questions were not relevant enough to include in an interview situation. And these kinds of tests only address a fraction of what is important about a candidate's qualifications and fit for a position.

Certification

Certification is a hot topic this year. SAS Institute has developed a questionnaire on their web site on certification. Several entities have popped up on the web offering to certify many different skill sets, including SAS. SAS-L heartily took on the discussion of whether certification has a place in the industry. SUGI 24 is featuring a panel discussion on certification.

Wrap Up

It would not be prudent to draw conclusions from this survey, but the responses were very interesting and instructive. What seems evident is that the needs of hiring entities vary greatly and the hiring process is highly subjective. And people are looking for very different skills sets and personality traits in candidates.

What I am taking away from this exercise is a new perspective and respect for the process, an opinion on proficiency testing and certification (draw your own conclusions), and some resources for improving interview style and content. Good and bad can be found in almost any approach. And, we can learn from one another.

Acknowledgments

Special thanks to Susan Fehrer from BioClin for her perspective, experience and passion for the subject. Thanks to Bob Hamer and Joshua Sharlin for their candor. Many thanks to survey respondents, especially Lee Herman, Cindy Zender, Linda Pickle, Bob Virgile, Greg Barnes Nelson, Andy Kuligowski, Aria Razban, Tracy Cermack, Steven Weinburg, Ian Whitlock, Heidi Markovitz, Cyndie Gareleck, Pete Hellmann, and James Gear. Thanks to the following SAS-L contributors for sharing their interview thoughts and questions with the SAS community: Standish Sibley, Mike Rhodes, Paul Dorfman, Karsten Self, Jack Hamilton, Girish Patel, Timothy Berryhill, Patricia Flickner, Chris Strickland, Ian Whitlock, Andy Karp, Erik Larsen, Bernard Tremblay, and David Nasser.

References

Deems, Richard S.: Hiring: More Than a Gut Feeling. Book-mart Press, 1995.

Messmer, Max: The Fast Forward MBA in Hiring: Finding and Keeping the Best People. John Wiley & Sons, Inc., 1998.

Yate, Martin: Hiring the Best. A Managers Guide to Effective Interviewing. Holbrook, MA: Adams Media Corporation, 1994.

SAS is a registered trademark of SAS Institute Inc., Cary, NC.

Selected Interview Questions from SAS Users and Managers

From contributions to SAS-L, individual suggestions, questions in use, etc.

NOTE:

⇒ the paper author does not vouch for the value of these questions

- ⇒ answers are not supplied
- ⇒ there could be multiple answers to some questions
- ⇒ many questions overlap - pick and choose – use at your own risk
- ⇒ logic and problem solving demonstration can be more telling than a correct answer to a syntax question
- ⇒ administering non-standard proficiency tests could have legal ramifications (please visit your personnel department)
- ⇒ some great programmers are lousy interviewers or test-takers; some great interviewers and test-takers are not good programmers
- ⇒ the list is by no means comprehensive

Very Basic

- ◇ What SAS statement would you code to read in an external raw data file to a DATA step?
- ◇ How do you read in the variables that you need?
- ◇ Are you familiar with special input delimiters? How are they used?
- ◇ If reading a variable length file with fixed input, how would you prevent SAS from reading the next record if the last variable didn't have a value?
- ◇ What is the difference between an informat and a format? Name three informats or formats.
- ◇ Name and describe three SAS functions that you have used, if any?
- ◇ How would you code the criteria to restrict the output to be produced?
- ◇ What is the purpose of the trailing @? The @@? How would you use them?
- ◇ Under what circumstances would you code a SELECT construct instead of IF statements?
- ◇ What statement do you code to tell SAS that it is to write to an external file? What statement do you code to write the record to the file?
- ◇ If reading an external file to produce an external file, what is the shortcut to write that record without coding every single variable on the record?
- ◇ If you're not wanting any SAS output from a data step, how would you code the data statement to prevent SAS from producing a set?
- ◇ What is the one statement to set the criteria of data that can be coded in any step?
- ◇ Have you ever linked SAS code? If so, describe the link and any required statements used to either process the code or the step itself.
- ◇ How would you include common or reuse code to be processed along with your statements?
- ◇ When looking for data contained in a character string of 150 bytes, which function is the best to locate that data: scan, index, or indexc?
- ◇ If you have a data set that contains 100 variables, but you need only five of those, what is the code to force SAS to use only those variable?
- ◇ Code a PROC SORT on a data set containing State, District and County as the primary variables, along with several numeric variables.
- ◇ How would you delete duplicate observations?
- ◇ How would you delete observations with duplicate keys?
- ◇ How would you code a merge that will keep only the observations that have matches from both sets.
- ◇ How would you code a merge that will write the matches of both to one data set, the non-matches from the left-most data set to a second data set, and the non-matches of the right-most data set to a third data set.

Internals

- ◇ What is the Program Data Vector (PDV)? What are its functions?
- ◇ Does SAS 'Translate' (compile) or does it 'Interpret'?
- ◇ At compile time when a SAS data set is read, what items are created?
- ◇ Name statements that are recognized at compile time only?
- ◇ Identify statements whose placement in the DATA step is critical.
- ◇ Name statements that function at both compile and execution time.
- ◇ Name statements that are execution only.
- ◇ In the flow of DATA step processing, what is the first action in a typical DATA Step?
- ◇ What is _n_?

Base SAS

- ◇ What is the effect of the OPTIONS statement ERRORS=1?
- ◇ What's the difference between VAR A1 - A4 and VAR A1 -- A4 ?
- ◇ What do the SAS log messages "numeric values have been converted to character" mean? What are the implications?
- ◇ Why is a STOP statement needed for the POINT= option on a SET statement?
- ◇ How do you control the number of observations and/or variables read or written?
- ◇ Approximately what date is represented by the SAS date value of 730?
- ◇ How would you remove a format that has been permanently associated with a variable??
- ◇ What does the RUN statement do?
- ◇ Why is SAS considered self-documenting?
- ◇ What areas of SAS are you most interested in?
- ◇ Briefly describe 5 ways to do a "table lookup" in SAS.
- ◇ What versions of SAS (on which platforms) have you used?
- ◇ What are some good SAS programming practices for processing very large datasets?
- ◇ What are some problems you might encounter in processing missing values?
 - * In Data steps? Arithmetic? Comparisons? Functions? Classifying data?
- ◇ How would you create a data set with 1 observation and 30 variables from a data set with 30 observations and 1 variable?
- ◇ What is the different between functions and PROCs that calculate the same simple descriptive statistics?
- ◇ If you were told to create many records from one record, show how you would do this using arrays and with PROC TRANSPOSE?
- ◇ What are _numeric_ and _character_ and what do they do?
- ◇ How would you create multiple observations from a single observation?
- ◇ For what purpose would you use the RETAIN statement?
- ◇ What is a method for assigning first.VAR and last.VAR to the BY group variable on unsorted data?
- ◇ What is the order of application for output data set options, input data set options and SAS statements?
- ◇ What is the order of evaluation of the comparison operators: + - * / ** () ?

Testing, debugging

- ◇ How could you generate test data with no input data?
- ◇ How do you test/ debug your SAS programs?
- ◇ What can you learn from the SAS log when debugging?
- ◇ What is the purpose of `_error_` ?
- ◇ How can you put a "trace" in your program?
- ◇ Are you sensitive to code walk-throughs, peer review, or QC review?
- ◇ Have you ever used the SAS Debugger?
- ◇ What other SAS features do you use for error trapping and data validation?

Missing values

- ◇ How does SAS handle missing values in: assignment statements, functions, a merge, an update, sort order, formats, PROCs?
- ◇ How many missing values are available? When might you use them?
- ◇ How do you test for missing values?
- ◇ How are missing values represented internally?

General

- ◇ Do you observe any coding standards? What is your opinion of them?
- ◇ What percent of your program code is usually original and what percent copied and modified?
- ◇ Have you ever had to follow SOPs or programming guidelines?
- ◇ Name several ways to achieve efficiency in your program. Explain trade-offs.
- ◇ What other SAS products have you used and consider yourself proficient in using?

Functions

- ◇ How do you make use of functions?
- ◇ When looking for contained in a character string of 150 bytes, which function is the best to locate that data: scan, index, or indexc?
- ◇ What is the significance of the 'OF' in `X=SUM(OF a1-a4, a6, a9)`; ?
- ◇ What do the PUT and INPUT functions do?
- ◇ Which date function advances a date, time or date/time value by a given interval?
- ◇ What do the MOD and INT function do?
- ◇ How might you use MOD and INT on numerics to mimic SUBSTR on character strings?
- ◇ In ARRAY processing, what does the DIM function do?
- ◇ How would you determine the number of missing or nonmissing values in computations?
- ◇ What is the difference between: `x=a+b+c+d`; and `x=SUM(a,b,c,d)`; ?
- ◇ There is a field containing a date. It needs to be displayed in the format "ddmmyy" if it's before 1975, "dd mon ccyy" if it's after 1985, and as 'Disco Years' if it's between 1975 and 1985. How would you accomplish this in data step code? Using only PROC FORMAT.
- ◇ In the following DATA step, what is needed for 'fraction' to print to the log? `data _null_; x=1/3; if x=.3333 then put 'fraction'; run;`

PROCs

- ◇ Have you ever used "Proc Merge"? (be prepared for surprising answers....)
- ◇ If you were given several SAS datasets you were unfamiliar with, how would you find out the variable names and formats of each dataset?
- ◇ What SAS PROCs have you used and consider yourself proficient in using?
- ◇ How would you keep SAS from overlaying the a SAS set with its sorted version?
- ◇ In PROC PRINT, can you print only variables that begin with the letter "A"?
- ◇ What are some differences between PROC SUMMARY and PROC MEANS?
- ◇ PROC FREQ:
 - * Code the tables statement for a single-level (most common) frequency.
 - * Code the tables statement to produce a multi-level frequency.
 - * Name the option to produce a frequency line items rather than a table.
 - * Produce output from a frequency. Restrict the printing of the table.
- ◇ PROC MEANS:
 - * Code a PROC MEANS that shows both summed and meaned output of the data.
 - * Code the option that will allow MEANS to include missing numeric data to be included in the report.
 - * Code the MEANS to produce output to be used later.
- ◇ Do you use PROC REPORT or PROC TABULATE? Which do you prefer? Explain.

Merging/Updating

- ◇ What happens in a one-on-one merge?
- ◇ How would you combine 3 or more tables with different structures?
- ◇ What is a problem with merging two Data Sets that have variables with the same name but different data?
- ◇ When would you choose to MERGE two data sets together and when would you SET two data sets?
- ◇ Which dataset is the controlling dataset in the MERGE statement?
- ◇ How do the `IN=` variables improve the capability of a MERGE?
- ◇ Explain the message 'MERGE HAS ONE OR MORE DATASETS WITH REPEATS OF BY VARIABLES'.

Simple statistics

- ◇ How would you generate 1000 observations from a normal distribution with a mean of 50 and standard deviation of 20. How would you use PROC CHART to look at the distribution. Describe the shape of the distribution.
- ◇ How do you generate random samples?

Customized Report Writing

- ◇ What is the purpose of the statement DATA _NULL_ ; ?
- ◇ What is the pound sign used for in the data null?
- ◇ What would you use the trailing @ sign for?
- ◇ For what purpose(s) would you use the RETURN statement for?
- ◇ How would you determine how far down on a page you have printed in order to print out footnotes?
- ◇ What is the purpose of using the N=PS option?

Macro

- ◇ What system options would you use to help debug a macro?
- ◇ Describe how you would create a macro variable.
- ◇ How do you identify a macro variable?
- ◇ How do you define the end of a macro?
- ◇ How do you assign a macro variable to a SAS variable?
- ◇ For what purposes have you used SAS macros?
- ◇ Describe the function/utility of the most difficult SAS macro that you have written.
- ◇ What is the difference between %LOCAL and %GLOBAL?
- ◇ How long can a macro variable be? A token?
- ◇ If you use a SYMPUT in a DATA step, when and where can you use the macro variable?
- ◇ What do you code to create a macro? End one?
- ◇ Describe how you would pass data to a macro.
- ◇ You have five data sets that need to be processed identically; how would you simplify that processing with a macro?
- ◇ How would you code a macro statement to produce information on the SAS log? This statement can be coded anywhere.
- ◇ How do you add a number to a macro variable?
- ◇ If you need the value of a variable rather than the variable itself, what would you use to load the value to a macro variable?
- ◇ Can you execute a macro within a macro? Describe.
- ◇ Can you a macro within another macro? If so, how would SAS know where the current macro ended and the new one began?
- ◇ How are parameters passed to a macro?

Pharmaceutical Industry

- ◇ What pharmaceutical companies, CROs, and biotechs have you worked for using SAS and how long were you there?
- ◇ Describe the types of SAS programming tasks that you performed:
 - * Tables? Listings? Graphics? Ad hoc reports? Other?
- ◇ Have you been involved in editing the data or writing data queries?
- ◇ What techniques and/or PROCs do you use for tables?
- ◇ Are you involved in writing the analysis plan and tables specs?

Intangibles (some dealing with contracting)

- ◇ Will it bother you if the guy at the next desk times the frequency and duration of your bathroom/coffee breaks on the grounds that 'you are getting paid twice as much as he is'?
- ◇ How will you react when, while consulting a SAS documentation manual to get an answer to a problem, someone says: 'hey, I thought you were supposed to know all that stuff already, and not have to look it up in a book!'
- ◇ Can you continue to write code while the rest of the people on the floor where you work have a noisy party to which you were not invited?

Problems and open-ended questions

- ◇ Describe a time when you were really stuck on a problem and how you solved it.