

User-friendly Testing Analysis with SAS® Software Products

Mark L. Schneider, SAS Institute Inc., Cary, NC

ABSTRACT

The Quality Assurance (QA) department at SAS Institute Inc. is responsible for testing the numerous features of the SAS® System on a variety of hardware platforms. Much of the testing is conducted in an automated fashion, thus producing reams of output that must be scrutinized by QA analysts to weed out unexpected, inconsistent, or incorrect behavior. In response to this challenge, the QA tooling group has developed a centralized analytical system by which bugs can be quickly identified. The goal of the system is to reduce the time spent on problem identification, in effect maximizing time available for interactive testing and new testware development. The system was developed using SAS software products for two reasons: the ease with which user-friendly applications can be developed, and the opportunity for evaluating SAS software products in a real-world application.

INTRODUCTION

The QA department at the Institute maintains an ever-expanding library of testware currently composed of over 50,000 programs. Most of these programs run under each of the many operating systems supported by the SAS System, including MVS, CMS, VMS, UNIX, Windows, OS/2®, and VSE. A standard set of testing tools is maintained on each of the target hosts. These tools are used to execute the programs using the latest version of SAS software, compare the results to known benchmarks, and report the test results in a host-specific format. These test results are then transferred to a common platform (HP 9000), which is the primary system used by QA analysts.

A base SAS® DATA step program is then used to read in these results, converting them to SAS data sets. In this format, a myriad of SAS procedures can be used to format, manipulate, display, and graphically summarize the data. A single analytical tool (QATRACK) is used to monitor test results as well as to perform tasks affecting the resubmission of failed tests. In addition to the extensive data management and reporting capabilities offered by base SAS software, several additional products are employed by QATRACK:

Procedure	Product	Use within the Tool
N/A	SAS/AF® software	Menu-driven interface
DOWNLOAD	SAS/CONNECT™ software	Download SAS data sets from remote platforms

Procedure	Product	Use within the Tool
FSEDT	SAS/FSP® software	Browse/edit data one program per screen
FSVIEW	SAS/FSP software	Browse/edit data one program per line
FSLIST	SAS/FSP software	Browse flat files
GPLOT	SAS/GRAPH® software	Performance trend analysis plot
N/A	SAS/SHARE® software	Simultaneous write access to multi-user data

DATA DESCRIPTION

Due to the volume of results from each of the platforms tested simultaneously, the data are broken down by product or feature and by the operating system under which the tests were run. For example, results for a certain statistical procedure under MVS are kept in a different data library than results for a graphical procedure under VMS. QA analysts are assigned to separate products and features, thus contention for the same data is greatly reduced. Although different data sets are maintained for each product-system pair, analysts and managers can generate reports that merge the data from separate data sets, allowing a global view of testing progress.

QATRACK maintains two classes of data for each product-system pair: latest and cumulative. The first corresponds to the most recent run for each of the test programs. The second contains an audit trail of all past runs. While the first data class is generally used for snapshot information on the health of an image under test, the second class is used to evaluate trends in system performance as well as to determine rerun and regression ratios.

Software tests are organized into groups of programs that are run together, known as test tables or test suites. In order to provide consistent information across systems, the following information is passed to QATRACK from the system-specific testing tools for each test case:

Program Test name

Table Test suite to which the program belongs

Dvd	Functional area to which the table belongs (stands for Development Directory)
SAS RC	Return code from the execution of the SAS image
Version	Internal version ID for the image tested
Date Run	Date and time the program was run
Result1-Result4	Comparison result of each output type to its benchmark

Note that each test program may produce up to four output types that are compared with their respective expected benchmarks. The Result fields are used to report whether the result and benchmark were found, and if so, the number of differences detected between the two.

Consolidating Results

If the system-specific testing tool outputs the previous fields to a flat file, FTP is used to pass the file to the HP 9000 platform. From there, a simple DATA step is used to input the file to a SAS data set. If the system-specific testing tool is implemented such that results are already stored in SAS data set format, SAS/CONNECT software is used to make a connection to the remote host and download the test results to the HP 9000 platform. The following DATA step program is used to download the results data set from our MVS system:

```

/*****
* Log on to the remote MVS system using a
* telnet script supplied with SAS/CONNECT.
*****/
options comamid=telnet remote=mvscpu;
signon "telso.scr";

rsubmit sdmvs;
/*****
* Set up a libref to the MVS data library
* using the SAS/SHARE engine.
*****/
options comamid=xms0;
%shrmacs(user,nomsg,applsys=mvs);
%libdef(MVSLIB, MVSTEST.R608.TRACK);

/*****
* Download the data set to a local,
* temporary data set.
*****/
proc download data=MVSLIB.qarslt
              out=download;

run;
endrsubmit;

signoff "&QATROOT/sasrc/telso.scr";

```

Since test suites are run asynchronously, results can be posted to the MVS data set at any time during the day. In order to prevent collisions, the SAS/SHARE server is used to handle multi-user write access to the results data set. The SHRMACS and LIBDEF functions are used to set up access to the appropriate share server and to point to the physical MVS data set holding the SAS data library.

Latest Data

As discussed previously, the first data class maintained by QATRACK is a snapshot of the most recent results for each test program. In addition to the standard fields provided by the system-specific testing tools, a current status field is computed by QATRACK that gives the analyst a quick reference to the test's success. Valid statuses include the following:

CLEAN	All results compared cleanly to their corresponding benchmarks
BADRC	An invalid return code was generated by the SAS image
DIFFS	Differences were detected between the results and their benchmarks
NORESULT	No result(s) found
NOBENCH	No benchmark(s) found
MULTIPLE	More than one type of error detected

A number of editable fields are also provided to help the tool developer add descriptive information concerning tests that were not marked CLEAN. This information is particularly useful in tracking the resolution of outstanding bugs. These fields are discussed in greater detail in the FSEDIT section.

Cumulative Data

The second data class maintained by QATRACK is the audit trail of all past runs for each test program. Since there is no limit to the number of observations (records) that can be appended to the audit trail, the data set is stored in compressed format using the COMPRESS= data set option. This reduces its size by compacting repeated consecutive characters or numbers to 2-byte or 3-byte representations.

Since the cumulative audit trail is accessed only for reports, QATRACK can limit the user to read-only access. Obviously, having write access to a testing audit trail could have a hazardous impact on evaluating testing progress. If such access were granted, the user would be able to remove or modify entries for a particular test case, thus eliminating important regression run information.

Some system-specific testing tools supply performance statistics along with the standard fields previously listed. These would include on CPU utilization, memory usage, and I/O operations. These data are stored in the audit trail so that trend analysis can be performed for representative test programs.

MENU SYSTEM

SAS/AF software is used for QATRACK's menu-driven interface. The QATRACK menus take advantage of several of the screen-development features offered by this product, including pull-down menus, selection lists, push buttons, radio buttons, and fill-in-the-blank fields. Figure 1 shows the

QATRACK main menu.

CROSS-PRODUCT report can be found in the Reports section later in this paper.

FSEDIT VIEW OF THE LATEST DATA

The FSEDIT procedure is a facility provided by SAS/FSP software that allows the user to view data one observation (or record) at a time. PROC FSEDIT provides many formatting features to the application developer, allowing a more user-friendly presentation of the data through screen positioning, color, highlighting, and I/O validation.

The PROC FSEDIT portion of QATRACK provides the analyst with the greatest power and flexibility in performing problem resolution. Figure 3 shows the FSEDIT view of the latest data.

QATRACK Main Menu

End Family

* QATRACK *

Product/Feature:

DEV608 DEV

Choose one of the following functions:

☒ FSEDIT FSVIEW REPORTS
(per product/system pairs)

CROSS-SYSTEM REPORTS
(per product)

CROSS-PRODUCT REPORTS

☐ QUIT

Figure 1: QATRACK Main Menu

From the main menu, the user can perform any of the following functions:

- Select the system to review via the Family pull-down menu.
- Select the product to review via the fill-in-the-blank field.
- Edit the latest data via either the FSEDIT or FSVIEW radio buttons.
- Run reports on both the latest and cumulative data via the various reports buttons.

In the case of the reports buttons, sub-menus are displayed that prompt the analyst for more information as to the content of the desired report. For example, if the CROSS-PRODUCT REPORTS button is pressed, the menu shown in Figure 2 is displayed:

Global QA Group Reports

END

CROSS-PRODUCT REPORTS FOR FM3607

QA Group:

From Date: To Date:
(Blank from/to dates imply all dates)

Select one of the following reports:

☒ FREQUENCY COUNT PER PRODUCT/SYSTEM

☐ LIST OF ALL PROGRAMS (long report)

(Results will appear in the SAS:OUTPUT window)

Figure 2: CROSS-PRODUCT REPORTS Menu

Many of the same SAS/AF screen development features are copied from the main menu in order to provide a consistent look and feel. A more thorough explanation of the

FSEDIT Screen

File Edit Search View Locals Globals Help

Program: abghl17 Table: afmast af / DEV DVD: af Output: _

Status : DIFFS MULTIPLE Problem:

Reviewed: Date Initials

Verified: Date Initials

Machine Result Rebench Date Run: 31MAR92:23:00:00

LOG 2 DIFF(s)

LST

BDO 1 DIFF(s)

TXI

SAS RC: NORMAL Version: 6.07.0400D123199

CDBUGS:

Comments:

SAS Options:

Other Info:

Figure 3: FSEDIT Screen

The most recent data for a single test program are displayed using formats for certain fields to make them more human consumable. For example, the raw data corresponding to the four results are changed from numeric and keyword values to a fully descriptive text string. A '1' is changed to '1 Diff(s)', and an 'NB' is changed to 'No Benchmark'. In addition to the fields supplied by the system-specific testing tools, the following documentation fields are provided to help the analyst track bugs:

Problem	Reason for the test failing (for example, precision differences)
Reviewed	Signoff for first review of the problem
Verified	Signoff for resolution of the problem
Rebench	Mechanism to identify the need for a new benchmark
CDBUGS	Cross-reference to bug-tracking system
Comments	Miscellaneous information
SAS Options	Special options used in running the test
Other Info	Catch-all for information not found in the other fields

Scrolling and Searching

The analyst can scroll from one test program to the next using either the pull-down menu or user-defined function keys. However, a quicker method of locating particular programs is provided using the powerful WHERE clause capabilities offered by SAS/FSP software. WHERE clauses allow the data to be subsetted by selected values for data fields. The analyst may only be interested in programs that did not generate a status of CLEAN. Using a WHERE clause of the form "WHERE STATUS1 NE 'CLEAN'" will restrict the viewable test programs to those that suffered problems at run time or did not compare cleanly with their benchmarks. Similarly, a clause of the form "WHERE TABLE = 'suite1'" will limit the viewable test programs to those belonging to the logical group of programs named 'suite1'. Since subsetting by program name, table, and dvd are frequently desired, short-cuts have been implemented to construct these clauses. Simply typing over any of these three fields triggers the generation of an automatic WHERE clause using the entered value.

Rebenching

Since the SAS System is hosted on disparate platforms, differences frequently occur due to varying levels of host precision for floating-point values. In these situations, the analyst verifies the accuracy of the output and saves the output on the target platform as a temporary benchmark. Differences reported on subsequent runs will be restricted to regressive behavior as is desired. This process is known to QA as hostbenching, and it is made simple from within QATRACK through the Rebench fields. In order to make a copy of the output as a benchmark on the remote host, the analyst simply types NEEDHOST in the corresponding Rebench field. QATRACK takes advantage of networking tools to accomplish the remote hostbench. If successful (that is, no network errors are encountered), the Rebench value is changed to HOST. Henceforth, if differences are reported for this output type, they are easily identified as regressions.

Viewing of Remote Results

As previously mentioned, most QA analysts at SAS Institute use the HP 9000 workstation as their primary computing device. In order to centralize most of the analyst's work to this platform, QATRACK provides the capability to remotely access test input, output, benchmarks, and differences for the most recent runs. These are available through the Output field on the FSEDIT screen. Placing an X in this field invokes another SAS/AF menu (Figure 4) of pushbuttons corresponding to the various testware files stored on the remote system.

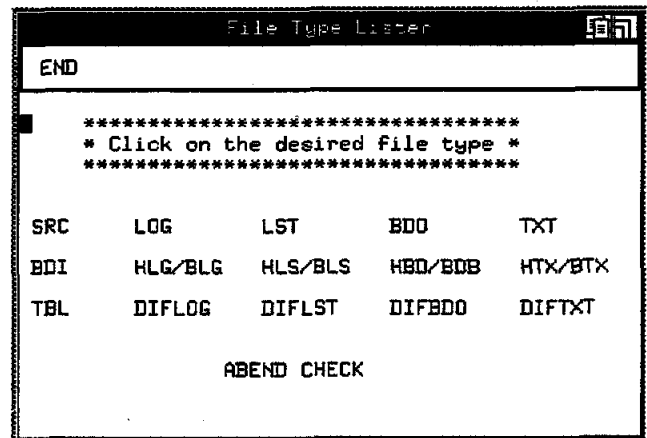


Figure 4: Remote File Lister Menu

Clicking the mouse causes QATRACK to retrieve the associated file type from the remote system, displaying it in a read-only window using the FSLIST procedure. Multiple file types can be viewed simultaneously. For example, the analyst may first bring up the difference listing accompanied by the associated output file and benchmark. After viewing the differences, a bug may be suspected in the test program itself. This can be verified quickly by clicking on the SRC button and examining the input program as it appears on the remote host. If a rerun is warranted, the analyst simply exits the menu and places a RERUN value in the STATUS field. Other QA employees, namely processors, then can use the WHERE clause capability to list all test programs that have been so marked. Since processors are responsible for the execution (and re-execution) of test suites on the remote systems, analysts never have to leave QATRACK to perform problem resolution.

FSVIEW VIEW OF THE LATEST DATA

The FSVIEW procedure is another facility provided by SAS/FSP software that allows the user to view multiple observations (records) at the same time. Like PROC FSEDIT, PROC FSVIEW provides many formatting features to the application developer; however, in order to fit multiple observations on a screen, certain documentation sacrifices must be made. Fields are described using developer-specified column headings, with a width (along with that of the data) that governs the number of fields visible at a time. In order to fit the maximum number of fields on a screen, these descriptions tend to be more terse than their FSEDIT counterparts.

QATRACK's FSVIEW option allows the analyst to view and modify the data for groups of test programs all on the same screen. This is extremely useful when a modification is to be made to the same field of a number of programs. The new value simply is placed in the same column of each corresponding row. Since WHERE clauses are also valid in PROC FSVIEW, a subset of test cases can be established easily. Figure 5 shows the FSVIEW perspective of the latest data.

PROGRAMME	TABLE	STATUS	PROBCH	RESULT1	RESULTS	RESULT2
abg1hk08	afmast	CLEAN		0		0
abg1hk16	afmast	CLEAN		0		0
abg1hk17	afmast	DIFFS		2		0
abg1hi08	afmast	CLEAN		0		0
abg1hi16	afmast	CLEAN		0		0
abg1hi17	afmast	DIFFS		2		1
abg1hm08	afmast	CLEAN		0		0
abg1hm16	afmast	CLEAN		0		0
abg1hm17	afmast	DIFFS		2		0
abg1hr08	afmast	CLEAN		0		0
abg1hr16	afmast	CLEAN		0		0
abg1hr17	afmast	DIFFS		2		1
abg1hc08	afmast	CLEAN		0		0
abg1hc16	afmast	CLEAN		0		0
abg1hc17	afmast	DIFFS		2		0
abg1hp08	afmast	CLEAN		0		0
abg1hp16	afmast	CLEAN		0		0
abg1hp17	afmast	DIFFS		2		1
abg1hz08	afmast	CLEAN		0		0
abg1hz16	afmast	CLEAN		0		0
abg1hz17	afmast	SDRC				
abg1ra03	afmast	CLEAN		0		0
abg1rb03	afmast	CLEAN		0		0
abg1rc03	afmast	DIFFS		0		S1
abg1rd03	afmast	DIFFS		0		S1
abg1re03	afmast	DIFFS		0		S1
abg1rf03	afmast	DIFFS		0		S1
abg1rg03	afmast	DIFFS		0		S1
abg1rh03	afmast	DIFFS		0		S1
abg1ri03	afmast	DIFFS		0		S1

Figure 5: FSVIEW Screen

Obviously, both the FSEDIT and FSVIEW procedures have distinct functional advantages, and it is up to the user to determine which of the two best suits the operation to be performed. Novice users of QATRACK are encouraged to use PROC FSEDIT first until they achieve a basic familiarity with the data. Experienced users tend toward the more global window of data presented by PROC FSVIEW. In either case, it should be noted that the two procedures operate on the exact same set of data. Changes in one view are instantly reflected in the other.

REPORTS

Whenever data are captured, reports invariably follow. The SAS System offers a multitude of procedures related to the production of user-friendly reports. QATRACK makes explicit use of these procedures to handle sorting, printing, plotting, and computing frequency statistics on the test results.

Figure 6 shows a frequency count of each of the test programs for the BASENCL product and the OS/2 system, broken down by test suite and resolution status.

```

File Edit View Globals Help

13:54 Wednesday, August 19, 1992 74
Frequencies for the STATUS1 variable
basenc1 / DS2

TABLE OF TABLE BY STATUS1

TABLE      STATUS1
Frequency|CLEAN |DIFFS |INDRESULT| Total
-----|-----|-----|-----|-----
bFreq | 133 | 6 | 0 | 139
chart2 | 66 | 0 | 0 | 66
Forms | 84 | 1 | 45 | 130
means | 106 | 0 | 0 | 106
reperr | 3 | 0 | 0 | 3
report | 93 | 3 | 1 | 97
tabulat | 197 | 1 | 0 | 198
Total | 682 | 11 | 46 | 739

```

Figure 6: Frequency Count for Each Table and Status

As previously mentioned, managers frequently want to look at information that spans multiple systems or products. Figure 7 is an example of the latter. Frequency counts summarizing the number of tests marked for rerun (STATUS1 = 'RERUN') are broken down by functional area (dvd) and system.

```

File Edit View Globals Help

                                08:35 Friday, August 28, 1992  3
                                TABLE 3 OF DVD BY SYSTEM
                                CONTROLLING FOR STATUS1=DIFF5

DVD      SYSTEM
Frequency  LOS2      IVSE      IWIN      Total
-----
ets       | 43 | 10 | 55 | 108
-----
etsbug    | 20 | 4 | 20 | 44
-----
expand    | 2 | 66 | 0 | 68
-----
Foresee   | 1 | 102 | 9 | 111
-----
loan      | 8 | 4 | 0 | 12
-----
mortgage  | 0 | 17 | 0 | 17
-----
tsars     | 0 | 18 | 12 | 30
-----
Total     | 74 | 220 | 96 | 390

```

Figure 7: Frequency Count for Each Dvd and System

Figure 8 shows the number of programs run for a user-specified date range, broken down by product and system. This is helpful to QA managers who are trying to stay abreast of their overall progress in a given week, quarter, or year.

```

File Edit View Globals Help

```

HOST Frequency Count of FM3607 Programs					115
Run Between 01JAN91 and 31DEC91					
13:54 Wednesday, August 19, 1992					
TABLE OF PRODUCT BY SYSTEM					
PRODUCT	SYSTEM				
Frequency	IMS	IMVS	IVMS	I	Total
hasbug	32	0	0	0	32
hasio	0	54	204		258
hasling	16	16	15		48
hasopt	435	1	0		437
hasprc	45	5	29		79
hasrt	255	85	1358		1700
setint	456	404	405		1266
test	197	2	60		259
Total	1439	568	2073		4079

Figure 8: Frequency Count for Each Product and System Bounded by a Date Range

SESUG '93 Proceedings

As previously mentioned, performance statistics are captured for each run of a test program. Figure 9 shows the trend of three such statistics over time for a representative test program. USERTIME and SYSTIME show CPU utilization, while SIGSRCVD measures interprocess communication. Each plotted point represents a single run of the test program.

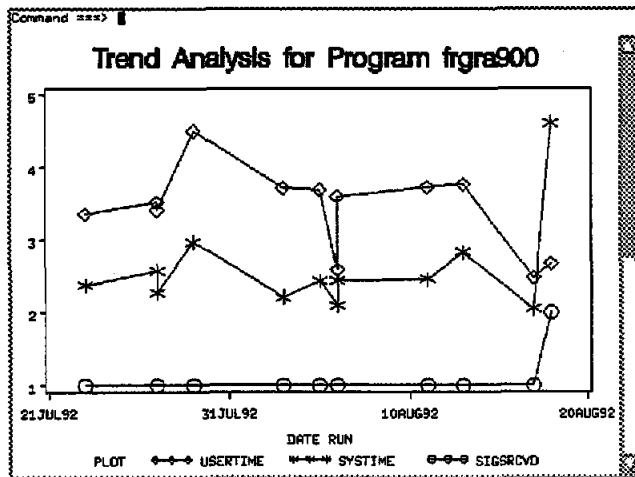


Figure 9: Performance Trend Analysis

In addition to the reports supplied in the QATRACK menus, users may run their own tailored reports outside of the QATRACK tool. All data are stored in public libraries and are readily accessible through either batch or interactive SAS jobs.

PORTABILITY

One of the huge advantages to QATRACK being an application based on SAS software is its ease of portability to any platform currently supported by the SAS System. In fact, the original version of QATRACK was hosted simultaneously under the MVS, VMS, and CMS platforms. These three versions were then consolidated and moved to a MIPS workstation. From there, QATRACK found its eventual home on the HP workstation.

Even though UNIX workstations are graphics-oriented devices, QATRACK could be invoked using an ASCII terminal. The SAS System provides windowing capability even on non bit mapped devices. This allows multi-display functionality, for example in viewing remote results, regardless of the device type. However, the look and feel is greatly enhanced by taking advantage of the bit mapped display offered by the UNIX platform.

FUTURE ENHANCEMENTS

As the SAS System evolves, the QATRACK system is poised to evolve along with it. Plans for the future include taking advantage of SAS/SHARE software when it is released under UNIX. Currently, QATRACK limits the second user of data for a particular product and system to

read-only access. SAS/SHARE will eliminate this restriction.

An obvious functional enhancement is the capability to initiate test suite reruns from within the QATRACK menu structure. Remote jobs could be submitted directly from the tool, taking advantage of the same network tools that are used to retrieve remote results. This will all but eliminate the need for QA processors to get involved in the verification of bug fixes, since QA analysts can launch reruns themselves.

QATRACK data and reports are prime candidates to be included in an Executive Information System (EIS). SAS/EIS™ software allows the creation of a customized information presentation system based on data generated from various sources. Current QA plans include the development of an EIS that will display information from QATRACK in conjunction with data from the Institute's bug-tracking system. The EIS will provide a single point from which managers can graphically monitor progress in the testing cycle.

Finally, the application can be converted to use SAS software FRAME entries that take advantage of graphics display devices and recent developments in graphical user interfaces. An example for using FRAME entries is in enhancing the trend analysis plot. Buttons can be attached to the plot allowing the analyst to select various performance statistics, specify a time range to be plotted, and scroll from one program to the next, all without ever having to leave the plot screen.

CONCLUSION

In taking advantage of the many data presentation and reporting procedures offered by the SAS System, the QATRACK test analysis tool has proven to be a dramatic time-saver in the identification and resolution of software defects. Its centralized approach to the management of test result data from a number of diverse platforms drastically reduces the amount of operating system expertise and electronic legwork required of the QA analyst.

ACKNOWLEDGMENTS

The author wishes to thank Kay Bydalek, Ellen Daniels, Sue McGrath, and Scott Ellis for technical review, and Richard Ragland and Stuart Austin for aid in typesetting the paper.

Base SAS, SAS/AF, SAS/FSP, SAS/GRAPH, SAS/SHARE, SAS/CONNECT, and SAS/EIS are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. OS/2 is a registered trademark or trademark of International Business Machines Corporation.

Other brand and product names are registered trademarks or trademarks of their respective companies.