

VIEWS

NEWS

Issue 8 Autumn 1999

News, Hints, Tips, and information for UK SAS Users

The deadlines for production of VIEWS News always seem to clash with the dates of key VIEWS events, so I'm unable to bring fresh news about these events. The last issue was too close to VIEWS 5 to do anything other than squeeze in a bullet celebrating Andy Wilcox's award of Best Paper. This issue has to go to press shortly before the northern event has taken place. I feel sure it will be a great success, but forgive me for being unable to report from the event.

This is the last edition of VIEWS News before the turn of the century. Have you listened to SAS Institute? Have you set your YEARCUTOFF to an appropriate value? If you're not sure, contact Technical Support and ask for advice. See our News item for further support information.

Andrew Ratcliffe (Editor)

Did You Know?

If you have a useful hint or tip, send it to the Editor and share it with the VIEWS membership.

Interrupting a SAS/AF® Application

Many times in a SAS/AF® application you have a long-running task that you'd like the user to be able to interrupt. In these circumstances you might like to try the EVENT() function. This function tells you whether the user has pressed a function key, the ENTER key, or a mouse button.

To demonstrate what it can do, create a frame with two buttons named and labelled 'Run' and 'Event'. Use the following SCL for the frame.

```
INIT:
  Control break ATTN;
RETURN;

MAIN:
  put 'The MAIN section was entered';
RETURN;

RUN:
  do until(event());
  end;
RETURN;

EVENT:
  put 'An event was detected';
RETURN;

ATTN:
  Put 'Attention was detected';
RETURN;
```

You will see that the frame goes into an "infinite" loop when you click the Run button. However, a mouse click or button press breaks the loop. If you break the loop by clicking on the Event button, the EVENT section's code will run; in all cases, the MAIN section's code will run.

The EVENT() function gives programmers a way to permit users to interrupt SCL activity. It does not interrupt submitted base SAS® code. EVENT() does not work on MVS and CMS and ASCII DRIVER machines. On these systems you should use the attention handler exit provided in SCL, i.e. CONTROL BREAK.

The CONTROL BREAK statement takes the name of a label as a parameter. If the user causes an interrupt whilst an SCL statement is executing, control will be passed to the labelled routine. The preceding code shows the label ATTN defined for this purpose.

Interrupts are created in a different manner on each operating system. On Windows the user should press Ctrl-Break; on MVS the user should press Attn.

Whilst the EVENT() function and the CONTROL BREAK statement permit similar functionality, the EVENT() function is more flexible and

user-friendly because it permits a wider range of events to cause the interruption - users will not intuitively use the Ctrl-Break or Attn key.

Andrew Ratcliffe

Accessing Sample Code

There are numerous online sample programs and code available free to SAS users. From the Technical Support web site (at http://www.sas.com/service/techsup/sample_library.html), additional samples and the Version 6.12 Samples Library for Windows are available. These samples demonstrate different features and how to solve application problems.

The Publications web page provides online samples from many SAS books. View this at service/doc/code.samples.html, or download by using Anonymous FTP. SASDOC-L is a listserv maintained by the Publications Division. As a subscriber, you can request ASCII files that contain sample programs. You also receive notification when sample programs from a new book become available. For more information about subscribing to SASDOC-L, send e-mail to pubs@sas.com.

NEWDOCNEWS-L and Andrew Ratcliffe

Splitting Data

We all know that SAS software has the uncanny ability to join data sets together at the DATA step level. Indeed, someone can write:

```
data big;
  set small1 small2 small3 ... smallN;
run;
```

There comes a time that the opposite is needed, that is to split a huge data set into smaller data sets satisfying certain criteria. Here is a recent posting by Michael Major on SAS-L:

'I have a master file that contains the STATE code (sometimes it contains all the US States and sometimes not) and I would like to generate a report (via PROC REPORT or PROC PRINT) that is saved to its own file when the BY value changes. That way I would have one file that contains the MD Report, a second file that contains the VA Report, etc. This master file is quite large so I would only like to make 1 pass through the data.'

Here's an idea. Assume that data set PERM.STATES has a variable called STATE.

```
proc sql noprint;
  select distinct state into :statalst
    separated by ' '
  from perm.states;
quit;

%put STATALST has been set to &statalst;

data _null_;
  call execute("data &statalst;");
  call execute('set perm.states;');
  call execute('select(state);');

  statelst="&statalst";
  count=1;
  state=scan(statalst,count);
  do while(state ne '');
    call execute('when (" ' || compress(state)
      !!') output '
      !!compress(state)
      !!';');
  );
  count=count+1;
  state=scan(statalst,count);
```

```

end;

call execute('end;');
call execute('run;');
run;

```

This very idea can be used in a number of ways to split data. It reads the data set twice: firstly in the SQL to generate a macro variable to contain the list of states (separated by spaces), secondly in the DATA step generated by the CALL EXECUTE statements. The abbreviated log is shown below.

```

501 proc sql noprint;
<snip>
506
507 %put STATELIST has been set to &statalst;
STATELIST has been set to AA MM ZZ
508
509 data _null_;
<snip>
525 run;
NOTE: The DATA statement used 0.2 seconds.

NOTE: CALL EXECUTE generated line.
1 +data AA MM ZZ;
NOTE: CALL EXECUTE generated line.
2 +set perm.states;
NOTE: CALL EXECUTE generated line.
3 +select(state);
NOTE: CALL EXECUTE generated line.
4 +when ("AA") output AA;
NOTE: CALL EXECUTE generated line.
5 +when ("MM") output MM;
NOTE: CALL EXECUTE generated line.
6 +when ("ZZ") output ZZ;
NOTE: CALL EXECUTE generated line.
7 +end;
NOTE: CALL EXECUTE generated line.
8 +run;

NOTE: The data set WORK.AA has 2 observations and 2 variables.
NOTE: The data set WORK.MM has 1 observations and 2 variables.
NOTE: The data set WORK.ZZ has 2 observations and 2 variables.
NOTE: The DATA statement used 0.45 seconds.

```

Note that the DATA step will open a number of data sets concurrently - one for each value of STATE. If you have a limitation on the number of concurrently open files, or the number of STATES is very large, you need to adapt the program somewhat.

Paul Kairis and Andrew Ratcliffe

Importing Made Easy

If you have flat file data to read into your SAS/AF® application, you may find that the IMPORT function (only available in SCL) is useful. Give it the name of the file you wish to read and the name of the data set you wish to create from the file; the user will be prompted with a screen that allows them to specify the start and end columns of the fields and much more besides.

It doesn't appear to offer support for delimited files (with embedded quotes), and maybe it gives the user too much flexibility for many applications, but it's worth a look.

Another option is to invoke the SAS Import Wizard. This is yet more flexible **and** permits the importing of delimited data (with embedded quotes). The Import Wizard can be invoked with the DIMPORT command or with a CALL DISPLAY to SASHELP.WIZARD. WIZMAIN.FRAME (but that's an undocumented way of invoking it).

Andrew Ratcliffe

Left-Handed Assignments

An often overlooked aspect of SAS syntax is that the SUBSTR() function can be used on the left-hand side of the equals sign in assignment statements. It doesn't allow you to do anything you couldn't otherwise do with SUBSTR() and concatenation on the right-hand side of an assignment, but it can make the statement more readable. A comparative example follows.

```

length string $200;
string = 'Hello Thomas';
put string;
substr(string,7) = 'Fat Controller';
put string;

length rope $200;
rope = 'Hello Thomas';
put rope;
rope = substr(rope,1,6) || 'Fat Controller';
put rope;

```

Both examples produce the following output.

```

Hello Thomas
Hello Fat Controller

```

Aside from variable names, the other thing you can use on the left-side of an assignment statement is an array reference. A simple example follows.

```

25 data _null_;
26 array railway{*} thomas gordon henry;
27 do ptr = 1 to dim(railway);
28 railway(ptr) = ranuni(0);
29 end;
30 put _all_;
31 run;

THOMAS=0.8965 GORDON=0.3140 HENRY=0.4552 PTR=4 _ERROR_=0 _N_=1
NOTE: The DATA statement used 0.33 seconds.

```

Andrew Ratcliffe

The Consultant

This part of VIEWS is where you can get your technical questions answered. Send your questions to the Editor.

What Happened?!

Q: *A colleague of mine has been working on bug fixes for version 1.5 of our SAS/AF® software while I've been working on the new version 1.6. She has had to change some of the same SCL entries as I have. She thought she had kept track of her changes so that her bug fixes could be fitted to my changed modules, but it turns-out that she's lost her notes. Is there any way of comparing the original SCL code with her changed copy so that we can see what changes she has made?*

A: SAS/AF software does not provide very many programmer productivity tools. Source Control Manager helps with some elements or change control but would not have helped in your instance.

The most straightforward means of comparing your two copies of the same module is to use PROC BUILD to write them out to a flat file (e.g. ASCII) and then use one of your operating system's tools to compare them. UNIX has a command named DIFF that can do this; MS-Windows users can use the FC command. The following PROC BUILD code will write-out an SCL entry to a flat file.

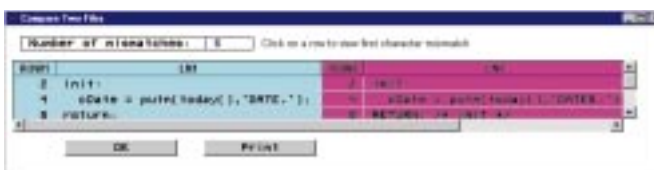
```

proc build c=ace_prog.v15 batch;
print source ptfille='c:\temp\v15scl.txt'
select=mixup et=scl ;
quit;

```

Alternatively, a freeware SAS utility named Cataloger is available from US Quality Partners Qualex. A free download from the Qualex web site, Cataloger is great for generating documentation of your SAS/AF applications (typically directed to MS-Word) but it also provides a comparator facility.

After downloading and unpacking Cataloger, assign the libref CATALOGR to the install directory. Run Cataloger and select ACE_PROG.V15 in the left-hand pane and ACE_PROG.V15B in the right-hand pane. Select MIXUP.SCL in both listings. You can now scroll up and down the two entries in synchronisation by using the central scrollbar. Better still, move to the Comparator window by clicking the Comparator icon and then click the Compare button. The resulting screen might look like the following example.



The screen shows V15 in the left column and V15B in the right column. It shows lines, and collections of lines, that have changed. You can see that your colleague made sure the label and RETURN statements complied with your site's Coding Standards, and you can see that she changed the length of the format being used in the PUTN() function to be sure the result was Y2K-compliant.

Cataloger does a lot more too, but I'll leave you to read the manual supplied with it and find out for yourself.

Andrew Ratcliffe

What is SAS-L?

Q: *I see occasional mentions in VIEWS News (and other SAS-related publications) to SAS-L. Can you tell me what it is and how I can get access to it please?*

A: SAS-L is an electronic bulletin board based on ListServ e-mail forwarding software. SAS software users around the globe use SAS-

L to post and respond to comments, general questions, and problems about SAS software. To post a message on the bulletin board you send a message to a designated email address; as others do the same, you receive emails.

To subscribe to SAS-L, send an e-mail message to listserv@uga.bitnet or listserv@cc.uga.edu containing the following one line of text: `subscribe sas-l (your_full_name)`

E-mail traffic on this list is heavy and the list is peered (mirrored) among several Listservs. When you subscribe to SAS-L, your e-mail address will be checked and your subscription will be automatically transferred to the server nearest you.

The Usenet Newsgroup `news:comp.soft-sys.sas` mirrors all SAS-L traffic and is provided for those who prefer to read messages as net news. Services such as Deja News also provide access to the group.

SAS-L is not the only SAS-related Listserv service. Take a look at the CataList page on the L-Soft International web site at www.lsoft.com/lists/listref.html. That page will help you search for lists of interest to you. If you search for host names containing 'vm.sas' you'll find a large list of services supported by SAS Institute.

Andrew Ratcliffe

Formats, Options, and Functions

Introduced with version 6.12 of SAS software, the **DATEAMPW.d format** writes datetime values with an AM or PM suffix. The **TIMEAMPW.d format** writes times with an AM or PM suffix.

The **SYSPRINT** and **SYSPRINTFONT** options set the printer and font to be used when printing.

The **N()** function returns the number of non-missing values. A varying number of numeric arguments are permitted. The arguments may use the OF-style (see *In Brief*).

News

IEWS in the North

IEWS' first event outside of the South East of England takes place on Thursday 25th November at an attractive venue: Old Trafford cricket ground, Manchester. The day's agenda is varied and includes four award-winning papers, including Best Paper from this year's SEUGI - Paul Nicholson from University of Leeds.

Registration forms were mailed in early October and interest is high, so if you've not already returned your form you should do so immediately. See the pages on the IEWS web site for the most up-to-date details.

sas.com interactive Is No More

SAS Institute's on-line magazine has ceased publishing. No replacement has yet been offered, but the SAS Institute web site is frequently updated with new and updated information.

If you used sas.com interactive as a reference tool, don't worry. The content of each issue has been archived and is accessible via the Hot Topics listed on the SAS Institute home page: <http://www.sas.com>

For instance, the stories and Webcast about Customer Relationship Management are available via the CRM software pages: <http://www.sas.com/software/app/crm.html>

PrintAssist

StatSphere, Inc. announced the release of PrintAssist 1.0 in August this year. PrintAssist helps you to print or publish PDF from saved SAS software output. You don't have to worry about those page formatting and cross platform issues any more.

For more information, contact StatSphere at the address given in Contacts. A demo version of PrintAssist 1.0 can be downloaded from their web site.

Year 2000 Support

Earlier this year, SAS Institute in the UK and Ireland wrote to all of its customers advising them of its plans for technical support over the New Year period. SAS Institute has arranged for technical support to be available directly from its headquarters in Cary. Technical support staff will be manning the phones in Cary from 14:00 GMT on 31st December 1999 continuously until 01:00 GMT on Tuesday 4th January 2000. A

number of development staff will also be available during that period. In the unlikely event of a failure in your SAS software, the number to call is 00 1 919 677 8008. Normal UK technical support will resume from 09:00 GMT on Tuesday 4th January 2000.

If your use of SAS software is critical enough you may be considering on-site support. SAS Institute will not provide such support, but some of the Quality Partners and Registered Partners are willing to offer an on-site service. The partners are listed at <http://www.sas.com/offices/europe/uk/partners/index.html>.

And finally, if you haven't looked recently, take a look at the tools and the list of known Year 2000 issues on SAS Institute's web site at <http://www.sas.com/service/y2k>.

If you are in any doubt, ring technical support before the event!

In Brief

- Use `PROC OPTIONS INTERNAL; RUN;` to see undocumented options in addition to the standard ones
- `%PUT _LOCAL_;` will show you all your local macro variables' values. Also try `_global_` and `_all_`
- For functions permitted a varying number of arguments, such as the `SUM()` and `N()` functions, the arguments may be specified using the OF-style. The OF-style is a list of variables preceded by OF, e.g. `sum(of x1-x4)` is equivalent to `sum(x1,x2,x3,x4)`

Diary

Are you organising an event that would be of interest to the IEWS readership? Let us know, as we are interested in all non profit making events related to SAS software.

November 1999

25th Northern conference, Old Trafford cricket ground, Manchester

February 2000

1st Ninth edition of IEWS News
Business sub-group (provisional)

April 2000

9-12 SUGI, Indianapolis, USA
Pharma SIG (provisional)

June 2000

20-23 SEUGI, Dublin, Ireland

September 2000

IEWS#6 conference (provisional)

Contacts

Contributors and Contacts

Paul Kairis Tel: +1 480 963 7967 Fax:
Quality partner in USA Email: kairis@email.com

Qualex Consulting Services, Inc. Tel: +1 919 381 8284 Fax:
Quality partner in USA Email:
Web: www.qlx.com

StatSphere, Inc. Tel: +1 508 624 6454 Fax: +1 508 624 0848
Quality partner in USA Email: info@statsphere.com
Web: www.statsphere.com

IEWS

Web Site www.views-uk.demon.co.uk

Phil Mason Tel/Fax: 01491-834615
Chairman Email: phil_mason@email.com
16 Wood Street, Wallingford,
Oxfordshire. OX10 0AY

Bruce Bovill Tel: 01628-404347 Fax: 01628-404303
SAS Liaison Email: sukbnb@suk.sas.com
Web: www.assisi.demon.co.uk

Andrew Ratcliffe Tel/Fax: 01322-525672
Newsletter Editor Email: andrew@ratcliffe.demon.co.uk
Web: www.ratcliffe.demon.co.uk

