

VIEWS

NEWS¹⁵

3RD QUARTER 2001

NEWS, HINTS, TIPS AND INFORMATION FOR UK SAS USERS

This year's main conference is just a few weeks away (September 25th and 26th). VIEWS' mission is to encourage the sharing of knowledge and experience on the usage of SAS. That's the mission of the conference too; and with three streams on each of two days, there's every reason to believe that you will find papers that interest you throughout the two days.

Conferences such as VIEWS can also be looked upon as an excellent opportunity to broaden your knowledge as well as deepening it. And the mixer on the evening of the first day will give everybody a chance to discuss topics of interest in an informal atmosphere – with fellow customers, with professional consultants, and with SAS staff.

If you haven't already, I suggest you complete the enclosed Registration Form and return it straightaway. Return it before the early-bird deadline and get two days for the price of one!

Acting as Chairman of VIEWS and producing my own email hints & tips newsletter (see *News*) is time-consuming, so I shall be producing subsequent issues of *VIEWS News* with the help of a co-editor. As ever, *VIEWS News* will continue to provide short, punchy hints and tips on all aspects of SAS software. If you would like to contribute an article, or discuss the possibility of doing so, please email me at newsletter@views-uk.org.

Andrew Ratcliffe

THE CONSULTANT

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

Decoding WinAPIs

Q: I've always found the MODULEx functions and SASCBTBL to be a complete mystery. I'm trying to use the MessageBoxA routine. Can you please give me some background on the subject?

A: I'll try to explain what the following SASCBTBL entry is about:

```
routine MessageBoxA
module=USER32
minarg=4
maxarg=4
stackpop=called
returns=short
;
arg 1 input num format=pib4. byvalue;
arg 2 input char format=$cstr200.;
arg 3 input char format=$cstr200.;
arg 4 input num format=pib4. byvalue;
```

The MODULE= option indicates what DLL the routine is coming from. In this case, MessageBoxA will come from USER32.DLL. The MINARG= and MAXARG= options indicate the min/max number of arguments to pass. Most often those will be the same. STACKPOP=CALLED is used for the routines with the __system attribute (or __pascal), which most KERNEL32 and USER32 DLL entries are. (The called routine pops the stack before returning, vs. the calling routine popping the stack upon return). The RETURNS= option indicates what type of value is being returned.

The ARG statements correspond to the individual arguments to the MODULE/MODULEN/MODULEC function. Typically they map one-to-one with the actual arguments to the desired function, unless you are aggregating several arguments into a structure. (In the case of MessageBoxA, you're not).

MessageBoxA requires 4 arguments. I didn't see MessageBoxA

explicitly described in my MSDN entry, but I believe it's the same as MessageBox, whose description says:

```
int MessageBox(
    HWND hWnd,
    LPCTSTR lpText,
    LPCTSTR lpCaption,
    UINT uType);
```

The first argument is a handle, which is a long integer. Making it unsigned means that you use the PIB format, and PIB4. ensures that it's the right length. The second and third arguments are null-terminated strings. Since SAS doesn't typically deal with null-terminated strings, you use the \$CSTR formats in the SASCBTBL entry to ensure that null-terminated strings are handled. The \$CSTR format will trim trailing blanks and insert a null-terminator after the last blank. The informat will find the null terminator and replace it, and all subsequent characters up to the specified length, with blanks. The fourth argument, like the first, is a long integer so PIB is best to use.

The first and fourth arguments have the BYVALUE attribute. This is because MODULE uses by-address as the default. But MessageBoxA is expecting arguments 1 and 4 to be passed by value, so you need to indicate this in the appropriate ARG statements.

Amadeus Software Consultancy & SAS Development Teams

PharmaSUG 2001

I had the pleasure of attending this year's PharmaSUG in Boston, USA in May. Here are my notes from my visit.

The FDA compliance sessions were well attended where SAS users got to see how different companies approach the DEFINE.PDF required in e-sub. The session on CMM (Capability Maturity Model) outlined nicely a natural framework for SAS program development and maintenance. Production of web-ready outputs without using SAS/IntrNet left many users with ideas on re-assessing their current working practices.

Overall, the breadth of presentations allowed attendees to contrast their own company's approach to solve problems common to virtually all of the pharmaceutical industry. Sure, you can pick up coding tips but, in my opinion, it's improvements to processes that have the potential to save the most time and resources.

With many presentations demonstrating use of the current (USA) regulatory standards of PostScript & PDF, it's not hard to see that the next couple of PharmaSUG conferences will see how the pharmaceutical industry grapples with the next regulatory standard, namely, XML.

With only a handful of UK attendees in Boston, I would recommend more to make the trip to Salt Lake City in 2002, www.pharmasug.org

Nelson Kinnersley

Improved Files With Separators

Q: I think I recall mention of V8 improvements to the DSD feature in DATA step (for handling comma-separated files). Could you please tell me of anything new in this area?

A: The DSD feature on the file statement of data steps has become very neat in V8.

```
data _null_ ; ****look how small this data step has become ;
set &wanted ;
file 'abc.dat' dsd dlm=&dlim lrecl = 32000 ;
if _n_ = 1 then put &heads;
put &names;
run;
```

Clearly I have pre-loaded &heads and &names;

- &names is just the list of variables.
 - &heads is also these names, but quoted and separated by delimiters
- If a column of the data to be exported to .CSV contains a delimiter, then the DSD feature automatically quotes the variable to protect the delimiter that is really data. If that quoted field itself contains a quote, then the rules of quote() are applied (the " becomes "")

It is a serious improvement over the methods of v6, to become just: put &names;

Part 2

The methods for loading those macro variables (&heads and &names) are as simple in sql as before:

```
%let dlm = '09'x ; **** supposing a tab delimiter was wanted;
*Generate the heading line and variable name list together;
Option _last_ = source.data ; /* neat way to resolve one level
names*/
%let wanted =%upcase( &syslast );
proc sql noprint;
select quote( trim( name )) /* for headings*/
, trim( name ) /* for details */
into :heads separated by " &dlm "
, :names separated by ' '
from dictionary.columns
where libname = %scan( &wanted, 1, . ) and
memname = %scan( &wanted, 2, . ) /* that scan function
again!!*/
order by varnum /* may be important ! */
;
quit;
```

Peter Crawford

DID YOU KNOW?

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

OBS= Option

Have you ever wanted to use the obs= option at the same time as a where clause is applied? Of course in V6 a large amount of red writing would appear in the log.

However, in V8 this syntax works perfectly. But what observations are returned? The Amadeus.demog data set has been sub-setted to include only the Males (28 observations in this example). The firstobs= and obs= options have been applied to the temporary data set containing 28 observations resulting in observations numbered 10 to 20 inclusive being returned.

```
libname Amadeus 'c:\amadeus' ;
data obs_option;
set amadeus.demog (firstobs=10 obs=20 where=(gender=' M' ));
run;
```

Amadeus Software Consultancy Team

When Upgrading to Windows 98, You May Need to Change Your DDE Code

When using DDE to access Excel files, one must first open Excel. This may be done manually or one may use a SAS external call to open Excel. In Windows 95, this could be written

```
X C:\PROGRA~1\MICROS~1\OFFICE\EXCEL.EXE C:\your
directory\yourfile.XLS;
```

Note that there are no quotation marks in this statement.

However, Windows 98 apparently requires quotation marks around the entire string that SAS submits externally as well as quotes around the two internal parts.

```
x " 'C:\Program Files\Microsoft Office\Office\EXCEL.EXE'
""C:\yourdirectory\yourfile.XLS; "" " ;
```

In case the print is hard to read, the pattern above is: double single single double double double double double.

Note that you may use the abbreviation for the directory names as in the first case or the explicit names as in the second example. Excel does appear to be very picky about whether it sees a single quote or a double quote, which is why the statement above is written as you see it. I assume that one must use similar quotation marks when using DDE with other software products.

I should add that I am using Excel 97 SR-2.

Nat Wooding

Put Statement and @ | @@

@/@/@ holds the output record for the execution of the next PUT statement even across iterations of the DATA step. The trailing @ or double trailing @ line-hold specifier, if used, must appear at the end of the PUT statement. Note that in the PUT statement trailing @ and double trailing @ produce the same effect.

In example 1, we print out the variable name and on the same line (holding the line by using @) we print out any values of weights where it is not missing. Note that after the test to print out the weights, we need to take off the effect of the @. We can do this by using the PUT after the statement:

```
if weight ne . then put @15 weight @;
```

Example 1

```
data a;
input name $ weight;
put name @;
if weight ne . then put @15 weight @;
put;
cards;
John 45
Sarah .
Stephen 65
;
```

Listing 1

```
John 45
Sarah
Stephen 65
```

Example 2

To see the outcome of removing the PUT statement that releases the effect of the @ after

```
if weight ne . then put @15 weight @;
```

The output shows that the result printed in one line as the effect of the @ was not removed.

SUGI/SeUGI 2001

This year's two primary northern hemisphere SAS conferences were in Long Beach (California) and Florence (Italy). Naturally, I had to attend both!

As is traditional, the US conference was technically-focused, and the European conference was solutions-focused. Both have their merits, and I believe that SAS may be rebranding these events in the coming year in order to make the distinction clearer.

There were a number of papers on V9 at SUGI. This project has been named Mercury because one of its key aims is to bring greater speed to SAS. One of the most obvious ways that this has been achieved is in the use of parallel processing. SAS are taking a slow and careful approach to this, but you will see the fruits of their labours in the earliest releases on V9, and more will follow in later releases. The initial release of V9 is likely to hit European shores in the summer of next year.

At SeUGI the talk was of solutions, and one of the newest solution focuses for SAS is Enterprise Performance Management (EPM). This encompasses the concept of applying KPI (Key Performance Indicators) and balanced scorecard principles to the whole of the enterprise, i.e. suppliers, customers, and the organisation itself.

This is necessarily a short overview of the two conferences. Longer reviews were published in issues 4 and 5 of RTSL's newsletter (NOTE:) – see News. However, I'd like to finish by encouraging you to attend either of next year's events if possible. There will be a lot of talk about the new release of SAS and their new features, and I always associate great value with the ability to meet and talk with fellow SAS practitioners.

```
data a;
  input name $ weight;
  put name @;
  if weight ne . then put @15 weight @;
  *put;
  cards;
John      45
Sarah .
Stephen   65
;
```

Listing 2

```
John      65 Sarah Stephen
```

Example 3

As in example 2, but removed the trailing @ at the end of the statement:

```
if weight ne . then put @15 weight @;
```

```
data a;
  input name $ weight;
  put name @;
  if weight ne . then put @15 weight;
  *put;
  cards;
John      45
Sarah .
Stephen   65
;
```

Listing 3

```
John      45
Sarah Stephen 65
```

Safa'a Sadiq

FORMATS, OPTIONS, AND FUNCTIONS

The **\$REVERSw.** format reverses the supplied string. In addition, it left justifies the result. The **\$REVERJw.** Format reverses the string but does not justify the result

The **WORKTERM** system option specifies that files in the work library should be deleted when the SAS session ends normally. The contrary option, **NOWORKTERM**, specifies that the work directory and its contents should be retained at the end of a SAS session. Relatedly, the **WORKINIT** system option specifies that the work library should be cleared at the start of a SAS session; the **NOWORKINIT** option specifies that it should not be cleared

The SAS/AF SCL **WOUTPUT** function allows the programmer to manipulate the Display Manager (DM) output window. Actions include clearing the window, writing the window to an external file, and printing it.

NEWS

Onyx At InterNext

Back in April of this year, InterNext announced the launch of Onyx - an application server that allows SAS applications to use a wide variety of front-ends including web pages (HTML and a scripting language), Java applets, and client-side applications.

When using Onyx with the web, any .sas or .scl file or reference on a web server is executed using the SAS system, without the need to pass parameters to a CGI script explicitly. This creates clean URLs and programs that are easier to understand and develop: with Onyx the parameters and program to execute are passed intuitively.

The range of features and functionality of this new product is too large to cover here, so take a look at the InterNext web site for more details: www.internext-inc.com/software/onyx.

Hints & Tips By Email

If you missed the news article in the last issue you will be glad of this brief reminder that more hints & tips are available on the use of SAS software in the free email newsletter produced by Andrew Ratcliffe (Editor of *VIEWWS News*).

To subscribe, send an email to listmgr@ratcliffe.co.uk and type 'subscribe-note.colon' as the subject (without the quotes).

IN BRIEF

- Membership of *VIEWWS* is free. Just send an email requesting membership to membership@views-uk.org
- SAS Help suggests that it is possible to use the lag function to obtain the value of a variable up to 100 iterations ago, lag1, lag2...lag100. However the physical limit far exceeds 100. Calculating the 5,000,000th lag value is certainly possible, which should be far more than enough for anyone!!
- Back issues of *VIEWWS News* are available from the *VIEWWS* web site

DIARY OF EVENTS

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

September 2001

25-26 **VIEWWS main conference,**
30-1/6 **DiaMondSUG,**
Chicago

October 2001

9th **V8 Enhancements for**
Applied Statisticians
Free seminar by
Amadeus (See Contacts)

November 2001

1st **VIEWWS News** 16th Edition

April 2002

14-17 **SUGI,** Orlando, Florida,
USA

May 2002

? **PharmaSUG,** Salt Lake
City, USA

June 2002

? **SeUGI,** Paris, France

CONTACTS

Contributors and Contacts

Amadeus Software

Email: info@amadeus.co.uk
Tel: 01993-878287
Web: www.amadeus.co.uk

Peter Crawford

Email: Peter.Crawford@db.com

Nelson Kinnersley Roche Products Ltd.

Email: Nelson.kinnersley@roche.com

Safa'a Sadiq Covance

Email: Safaa.Sadiq@covance.com

Nat Wooding

Email: Nathaniel_Wooding@dom.com

SAS UK

Tel: 01628 486 933
Web: www.sas.com/offices/europe/uk

VIEWWS

Web Site

www.views-uk.org

Andrew Ratcliffe Chairman, Newsletter Editor

Email: newsletter@views-uk.org
Tel: 01322-525672
Fax: 0870-050-9662
Web: www.ratcliffe.co.uk

Steph Penning SAS Liaison

Email: sasliaison@views-uk.org
Tel: 01628-486933
Fax: 01628-404448

Views News is published quarterly by *VIEWWS* - the UK's independent SAS user group. *VIEWWS* is a non-profit organisation. Our Mission Statement and Rules can be viewed on our web site together with a list of current committee members.

Looking for a great future?

Training and Career Development ...

Biostat Consulting is a forward thinking, innovative and fast growing Contract Research Organisation providing data management, programming, statistics and reporting services to the pharmaceutical industry.

Having successfully built a prototype data management system that fully utilises SAS/IntrNet software and also provides the capabilities of Remote Data Capture, Biostat Consulting are looking to further develop this system and increase it's functionality. The intention is for the database management system to provide a cost-effective, efficient and flexible solution to the management of clinical trial data. Though the data is input and stored directly into SAS data sets, the user is within a comfortable and familiar Web Browser environment.

SAS/IntrNet

- Integration of the SAS system and the World Wide Web
- Live reports available direct to the user, whether data managers, investigators, clients or other study personnel
- Direct access to the study data for appropriate personnel
- The users' software and maintenance requirements are significantly reduced
- The user only needs to be able to access the World Wide Web and they can access the study database (with appropriate authorisation)

Pension, private health care, company car all available ...

Flexible Generous Package ...

In-house and Client based Projects ...

We are currently seeking professional and ambitious staff at various levels for the Biostat Consulting Oxfordshire office and client sites in the following roles:

Clinical Data Managers / Co-ordinators

SAS Programmers



If you would like further information please call us on 01932 706701 or email us on jobs@biostat.co.uk. Or visit us at our website – www.biostat.co.uk

If you are interested in one of these exciting roles, please write with a full CV, stating your location requirements and current salary to:

Biostat Consulting Ltd,
East Wing, Auckland House,
New Zealand Avenue,
Walton Upon Thames,
Surrey, KT12 1PL.

biostat^c

Biostat Consulting Ltd is a member of the ASA Group of Companies

Official sponsors of

VIEWS
The Independent
UK SAS User Group

sas@asaltd.co.uk


recruitment & contract consultants

ASA Ltd. First Floor, East Wing, Auckland House, New Zealand Avenue, Walton Upon Thames, Surrey KT12 1PL.
Contact Richard Field at ASA on Tel: 01932 701801 Fax: 01932 701800 Email: sas@asaltd.co.uk

SAS is a registered trademark of SAS Institute Inc. Cary, North Carolina, USA.