

Having completed my apprenticeship with *VIEWS News* 16 my New Year's resolution for 2002 is to continue to provide you with lots of information about SAS programming, with regular Hints and Tips, and news about SAS-related events. In this issue I hope you will be pleased to see that I have restored the Formats, Functions and Options section.

Now that we no longer have the financial support of SAS (see *News* below), it is now even more important that we work together to keep SAS users up-to-date with new ideas and techniques. If you would like to contribute an article, or discuss the possibility of doing so, please email me at newsletter@views-uk.org.

Philip R Holland (Newsletter Editor)

Membership of **VIEWS** is **free** and you can register for notification of all future **VIEWS** events by emailing us at membership@views-uk.org, making certain you include your name, email address and postal address, so that we can email you a free copy of the quarterly *VIEWS News*.

The Consultant

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

Polar Co-ordinates and Charts

Q: Can you please offer some hints and tips on using polar co-ordinates and creating polar graphs?

A:[SAS 6+8] Mention of polar co-ordinates stirred happy memories of school-day mathematics. Trouble is, those school days were a long time ago! However, we couldn't resist having a go.

As far as we are aware, no SAS modules specifically support polar co-ordinates, so you need a roll-your-own solution. It seems that there are two problems to solve:

1. Translating the polar co-ordinates to Cartesian so we can use PROC GPLOT.
2. Displaying reference lines that reflect the polar nature of the data.

The solution to the first step is easy - in fact you've probably already worked that out yourself: just convert using sin/cos...

```
/* Create some sample data - in degrees */
data data;
  radius = 1.414; angle = 0; output;
  radius = 1.414; angle = 45; output;
  radius = 1.414; angle = 85; output;
  radius = 3.414; angle = 85; output;
run;

/* Convert polar to cartesian, also convert */
/* degrees to radians. */
%let pi = 3.1415927;
data convert;
  set data;
  drop angle;
  x = radius * sin(2 * &pi * angle / 360);
  y = radius * cos(2 * &pi * angle / 360);
  radius=0;
run;
```

Okay, so we can plot this and it looks kind of useful:

```
proc gplot data = convert;
  plot y * x / vzero hzero;
run;
quit;
```

However, the solution to getting some nice reference lines (circles) is a little more tricky. GPLOT isn't going to simply draw them for you, so we have to make something ourselves.

First we tried annotate with CIRCLE, and SYSTEM values of 2, i.e. 'absolute data values', but it seems that the circle takes its radius only from the Y axis and therefore is not guaranteed to pass through the same point on the X axis. This is workable if your x and y axis are the same lengths *and* your aspect ratio is square. That is to say, the annotate will draw a true circle on your graph, but you most likely need an ellipse since your X and Y axes will have different scales. So we ditched the idea.

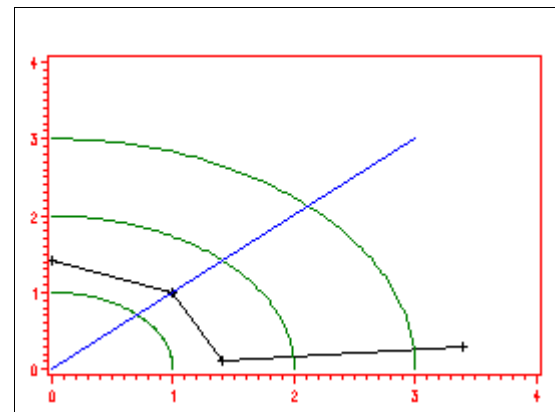
Next we thought of DSGI - it has an ELLIPSE function, but it does not support reference to data values... so it would be very tricky to get a circle/ellipse to go exactly through the major tick mark values. A bit of a non-starter...

Finally, we resorted to good old data values on the actual GPLOT. Here's what we came up with:

```
/* Create reference circles and lines */
/* Circles at 1 and 2 and 3 (plot a point at each degree) */
/* And a line at 45 degrees */
data reflines;
  drop angle;
  do radius = 1 to 3 by 1;
    do angle = 0 to 90 by 1;
      x = radius * sin(2 * &pi * angle / 360);
      y = radius * cos(2 * &pi * angle / 360);
      output;
    end;
  end;
end;
x = 0; y = 0; radius = 4; output;
x = 3; y = 3; radius = 4; output;
run;

/* Combine data with reference lines */
data plotter;
  set convert reflines;
run;

symbol1 i = join c = black v = plus; /* Data */
symbol2 i = join c = green; /* Circles... */
symbol3 i = join c = green;
symbol4 i = join c = green;
symbol5 i = join c = blue; /* 45 degree ref line */
axis1 label = none;
proc gplot data = plotter;
  plot y * x = radius /
    nolegend
    caxis = red
    ctext = red
    haxis = axis1
    vaxis = axis1
    ;
run;
quit;
```



Now all of this contains a lot of hard-coded stuff, like the number of reference circles, but it only took us a short while to knock it up. We hope it gives you some ideas as to which direction to head.

Finally, just for reference, here's the annotate solution we briefly played with:

```

/* FAILED SOLUTION - ANNOTATE */
%annomac;
%helpano(system);
%helpano(circle);
data annodata;
  %dclanno;
  %system(2,2,2);
  %circle(0,0,1,gray);
run;

proc gplot data = convert anno = annodata;
  plot y * x / vzero hzero;
run;
quit;

```

Ratcliffe Technical Services Consultancy Team

Describing a Data Step View

Q: Can you find out the code behind a Data Step View in Version 8?
A: [SAS 8] Yes you can. In Version 6 views could only be described if they had been originally created using PROC SQL. Version 8 provides the ability to describe views created within the Data Step.

```

data example_view / view = example_view;
  set sashelp.class (where = (sex = 'M'));
run;

data view = example_view;
  describe;
run;

```

```

355 data example_view / view = example_view;
356 set sashelp.class (where = (sex = 'M'));
357 run;

NOTE: DATA STEP view saved on file WORK.EXAMPLE_VIEW.
NOTE: A stored DATA STEP view cannot run under a different operating
system.
NOTE: DATA statement used:
      real time          0.01 seconds
      cpu time           0.01 seconds

358
359 data view = example_view;
360 describe;
361 run;

NOTE: DATA STEP view WORK.EXAMPLE_VIEW is defined as:

data example_view / view = example_view;
  set sashelp.class (where = (sex = 'M'));
run;

NOTE: DATA statement used:
      real time          0.01 seconds
      cpu time           0.01 seconds

```

Amadeus Software Consultancy Team.

Now the Euro Coins and Notes are here...

Q: How can you convert the old Euro-Zone currency values to euros?
A: [SAS 8 (+ 6.12)] The EUROCURR function is part of the standard installation of SAS version 8, and as an update pack for version 6.12. The function converts one European currency to another and returns a value. Here are some examples ...

1. To convert from DEM to EUR:

```

data _null_;
  amount=eurocurr(1,'dem','eur');
  put amount= ;
run;

```

If from-currency-code has a blank value, EUROCURR converts currency from euros to the currency of the specific European country that you specify.

2. To convert from EUR to DEM:

```

data _null_;
  amount = eurocurr(1,'eur','dem');
  put amount= ;
run;

```

If to-currency-code has a blank value, EUROCURR converts currency from the specific European country that you specify to euros.

3. Exchange Rates:

Rates were fixed for some countries on 1/1/1999, but can be altered for others by editing the file stored under the fileref EURFRtbl.

```

filename eurfrtbl 'c:\temp\eurfrtbl.dat';

data _null_;
  file eurfrtbl;
  input;
  put _infile_;
datalines;
EURFRGBP=0.641
EURFRCHF=1.4935
EURFRSEK=9.198
run;

data _null_;
  amount = eurocurr(1,'eur','gbp');
  put amount= ;
run;

```

4. SAS Formats for Conversion (*=fixed):

EURFRxxxw.d formats:

- EURFRATSw.d - convert to euros from Austrian schillings*
- EURFRBEFw.d - convert to euros from Belgian francs*
- EURFRCHFw.d - convert to euros from Swiss francs
- EURFR CZKw.d - convert to euros from Czech koruny
- EURFRDEMw.d - convert to euros from German marks*
- EURFRDKKw.d - convert to euros from Danish kroner
- EURFRFESp.w.d - convert to euros from Spanish pesetas*
- EURFRFIMw.d - convert to euros from Finnish markkaa*
- EURFRFRFw.d - convert to euros from French francs*
- EURFRGBPw.d - convert to euros from British pounds
- EURFRGRDw.d - convert to euros from Greek drachmas
- EURFRHUFw.d - convert to euros from Hungarian forints
- EURFRIEPw.d - convert to euros from Irish punts*
- EURFRITLw.d - convert to euros from Italian lire*
- EURFR LUFw.d - convert to euros from Luxembourg francs*
- EURFRNLGw.d - convert to euros from Dutch guilders*
- EURFRNOKw.d - convert to euros from Norwegian krone
- EURFRPLZw.d - convert to euros from Polish zlotys
- EURFRPTWw.d - convert to euros from Portuguese escudos*
- EURFRROLw.d - convert to euros from Romanian lei
- EURFRRURw.d - convert to euros from Russian rubles
- EURFRSEKw.d - convert to euros from Swedish kronor
- EURFRSITw.d - convert to euros from Slovenian tolar
- EURFRTRLw.d - convert to euros from Turkish liras
- EURFRYUDw.d - convert to euros from Yugoslavian dinars

EURTOxxxw.d formats:

- EURTOATSw.d - convert from euros to Austrian schillings*
- EURTOBEFw.d - convert from euros to Belgian francs*
- EURTOCHFw.d - convert from euros to Swiss francs
- EURTOCZKw.d - convert from euros to Czech koruny
- EURTODEMw.d - convert from euros to German marks*
- EURTODKKw.d - convert from euros to Danish kroner
- EURTOESPw.d - convert from euros to Spanish pesetas*
- EURTOFIMw.d - convert from euros to Finnish markkaa*
- EURTOFRFw.d - convert from euros to French francs*
- EURTOGBPw.d - convert from euros to British pounds
- EURTOGRDw.d - convert from euros to Greek drachmas
- EURTOHUFw.d - convert from euros to Hungarian forints
- EURTOIEPw.d - convert from euros to Irish punts*
- EURTOITLw.d - convert from euros to Italian lire*
- EURTOLUFw.d - convert from euros to Luxembourg francs*
- EURTONLGw.d - convert from euros to Dutch guilders*
- EURTONOKw.d - convert from euros to Norwegian krone
- EURTOPLZw.d - convert from euros to Polish zlotys
- EURTOPTWw.d - convert from euros to Portuguese escudos*
- EURTOROLw.d - convert from euros to Romanian lei
- EURTORURw.d - convert from euros to Russian rubles
- EURTOSEKw.d - convert from euros to Swedish kronor
- EURTOSITw.d - convert from euros to Slovenian tolar
- EURTOTRLw.d - convert from euros to Turkish liras
- EURTOYUDw.d - convert from euros to Yugoslavian dinars

The Europack update for Version 6.12 can be downloaded from the SAS web site. Details can be found on the [europackv6.html](http://www.sas.com/software/distribution/readme/europack_final.612/) web page at www.sas.com/software/distribution/readme/europack_final.612/.

Philip Mason, Woodstreet Consulting

Did You Know?

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

Multi-processing in Version 8

[SAS 8] One of the irritations in running a SAS program is having to wait for one step to complete before the next step can complete. Wouldn't it be a lot nicer (and quicker) if multiple steps could run in parallel? Version 9 of SAS will provide multi-processing to a number of procedures and situations, but it is possible to have a similar concept in Version 8.

```
options noxwait noxsync;

systask command = 'c:\multi\first_prog.bat'
         taskname = 'first_program';
systask command = 'c:\multi\second_prog.bat'
         taskname = 'second_program';

waitfor _all_ first_program ;

systask command = 'c:\multi\third_prog.bat'
         taskname = 'third_program';

waitfor _all_ second_program third_program;

libname amadeus 'c:\amadeus';

data amadeus.all;
  merge amadeus.first amadeus.second amadeus.third;
  by id_no;
run;
```

The program above contains three other programs, which are processed in turn. The first and second programs are begun in unison. Once the first program has finished the third can begin, regardless of the status of the second program. This is because this program uses the output dataset from program one as input. Upon completion of all three programs the main program can continue processing and merge all three output datasets together.

An example of one of the batch files is shown below:

```
cd 'c:\Program Files\SAS Institute\SAS\V8'
sas.exe -sysin c:\multi\first_prog.sas -log c:\multi\first_prog.log
```

Amadeus Software Consultancy Team.

Scheduling SAS Programs

[SAS 8] This is *simple* program scheduling tool, implemented using a macro. It was written on Windows XP running SAS 8.2. I'm sure it can be greatly improved, but it does demonstrate some useful basic techniques.

It has these features:

- Uses a dataset to list programs to run in sequence.
- Records return codes and start/end times.
- Provides a good basis for adding extra features, such as:
 - o Scheduling programs to run at particular times.
 - o Using a GUI for setting up dataset with jobs to run and to change sequence.
- Use of multiple processes via SAS/Connect.

To use follow these 3 steps:

Step 1 - Create the jobs dataset:

- Holds names of programs to run.
- Programs are in the sequence in which they should run.
- Fields defined for start/end datetime values, return code from program, system message from program.

```
data sasuser.jobs;
  length location $200
         start end 8
         rc 4
         msg $200;
  format start end datetime.;
  input location;
  put location=;
  return;
datalines;
c:\temp\progl.sas
c:\temp\prog2.sas
;;;
run;
```

Step 2 - Define the controller macro, which uses the jobs dataset to determine what to run next:

```
%macro controller;
  data sasuser.results;
  format start end datetime.;
  set sasuser.jobs;
  run;
  %let dsid = %sysfunc(open(sasuser.jobs,i));
  %syscall set(dsid);
  %let n = 0;
  %do %while(%sysfunc(fetch(&dsid)) = 0);
  %let n = %eval(&n + 1);
  %let start = %sysfunc(datetime());
  %include "&location";
  %let end = %sysfunc(datetime());
  data sasuser.results;
  set sasuser.results;
  if _n_ = &n then
  do;
    start = &start;
    end = &end;
    rc = symget('sysrc'); *may not be best source of rc;
    msg = symget('sysmsg');
  end;
  run;
  %end;
  %let dsid = %sysfunc(close(&dsid));
%mend controller;
```

Step 3 - Run it:

```
%controller
```

Philip Mason, Woodstreet Consulting

Formats, Options, and Functions

This section is devoted to the description of useful, or unusual, SAS Formats, Options, and Functions.

[SAS 6+8] The \$w. format and informat are widely used in SAS programs on all platforms, but every now and then you need to read data from, or write data to, other platforms, e.g. MVS data on Windows, UNIX data on MVS, etc. In fact \$w. is the same as \$ASCIIw. when used on Windows and UNIX platforms, and the same as \$EBCDICw. when used on MVS platforms, so to read data from, and write data to, different platforms just use \$ASCIIw. or \$EBCDICw. explicitly.

[SAS 8] The FOOTNOTES option of the FILE statement allows you to see global footnotes in your Data Step reports, like the TITLES option of the FILE statement allows you to see of global titles. The default setting is NOFOOTNOTES.

[SAS 6+8] The SAS/AF SCL EXIST, CEXIST, FEXIST and FILEEXIST functions can also be used in Data Step processing to test for the existence of different types of files. The EXIST function tests for a member of a SAS data libraries, including a dataset, view, or catalog. The CEXIST function tests for a SAS catalog, or an individual catalog entry. The FEXIST function tests for an external file associated with a SAS fileref. The FILEEXIST function tests for an external file by its physical name.

News

IEWS Subgroup for February cancelled

The VIEWS Committee regret to inform you that, due to circumstances beyond their control, the ODS Subgroup meeting listed in the VIEWS News diary in issue 16 has been cancelled.

----- **VEWS is now completely independent of SAS**

In January 2002 SAS UK stopped providing financial support for VIEWS activities. As a consequence *VEWS News* will now only be distributed by email, or via the web site, and there will be no big VIEWS Conference in 2002. However, *VEWS News* will continue to provide news and information to SAS users in the UK. To make sure you are still sent the newsletter, and details of other VIEWS activities, please send your name, email address and postal address to membership@views-uk.org

----- **FAQ's, Hints & Tips WebRing**

The web is full of information about SAS software, but it is not always easy to find. To make your searching simpler there is a group of web sites dedicated to supplying SAS-related hints and tips, and they are all linked together by a WebRing, which is a managed circle of similar sites, called the **SAS FAQ's, Hints and Tips WebRing**. If you want to see what is available in the WebRing, then one of the members of the ring is the home of the FAQ pages of Phil Holland, which can be found at www.hollandnumerics.co.uk/sasfaq/.

In Brief

- Back issues of *VEWS News* are available from the VIEWS web site.
- Amadeus Software are running a series of seminars throughout 2002 on popular topics at their offices near Witney in Oxfordshire (see **Diary** below). There are a maximum of 16 places at each seminar, so early booking is recommended. The seminars, with all handouts and refreshments, including lunch, are free of charge! For more details, and a complete and up-to-date series list, go to [Technical_Seminars.html](http://www.amadeus.co.uk/Events/Technical_Seminars.html) at www.amadeus.co.uk/Events/.

Diary

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

March 2002

18 SAS V8 Changes and Enhancements for Applied Statisticians, Amadeus Software

This is a free seminar! See **In Brief** for contact details.

April 2002

9 Tricks and Tips Seminar, Amadeus Software

This is a free seminar! See **In Brief** for contact details.

14-17 SUGI, Orlando, Florida, USA

The SAS User Group International (SUGI) Conference is held each year and focuses on the technical aspects of SAS software. Venues alternate between inland or coastal locations: Orlando is inland!

29 Tricks and Tips Seminar, Amadeus Software

This is a free seminar! See **In Brief** for contact details.

May 2002

1 VIEWS News 18th edition

28 Data Warehousing Seminar, Amadeus Software

This is a free seminar! See **In Brief** for contact details.

June 2002

5 SAS V9 Changes Seminar, Amadeus Software

This is a free seminar! See **In Brief** for contact details.

7 SAS V9 Changes Seminar, Amadeus Software

This is a free seminar! See **In Brief** for contact details.

11-13 SeUGI, Paris, France

The European SAS User Group International (SeUGI) Conference is held each year and focuses more on the commercial aspects of SAS software. The venue for 2003 is Vienna.

August 2002

1 VIEWS News 19th edition

October 2002

10 Communication Seminar, Amadeus Software

This is a free seminar! See **In Brief** for contact details.

November 2002

1 VIEWS News 20th edition

29 Data Warehousing Seminar, Amadeus Software

This is a free seminar! See **In Brief** for contact details.

All event information will be posted and/or emailed to registered members of VIEWS, and will include an event application form. Please send queries about any VIEWS events to events@views-uk.org, and don't forget to look at the web site for the latest news.

Contacts

Amadeus Software	Email: info@amadeus.co.uk Tel: 01993-878287 Web: www.amadeus.co.uk
Philip Mason Woodstreet Consulting	Email: philipmason@woodstreet.org.uk
Ratcliffe Technical Services	Email: enquiries@ratcliffe.co.uk Tel: 01322-525672 Fax: 08700-509662 Web: www.ratcliffe.co.uk
SAS UK	Tel: 01628-486933 Web: www.sas.com/uk
VEWS	
Web Site	www.views-uk.org
Daniel Morris Chairman	Email: chair@views-uk.org Tel: 01993-878287 Web: www.amadeus.co.uk
Philip R Holland Newsletter Editor	Email: newsletter@views-uk.org Tel: 07714-279085 Fax: 01763-244497 Web: www.hollandnumerics.co.uk
Stephanie Penning SAS Liaison	Email: sasliaison@views-uk.org Tel: 01628-404347 Fax: 01628-404303

VEWS News is published quarterly by VIEWS - the UK's independent SAS user group. VIEWS is a not-for-profit organisation. Our Mission Statement and Rules can be viewed on our web site together with a list of current committee members.