

Paper 472-2013

Configure your foundation SAS(r) client easily and without risk

Peter Crawford, Crawford Software Consultancy Ltd

ABSTRACT

For your client install of Foundation SAS(r), don't use the default provided, instead, use the simple features in this paper and presentation, to support the flexibility you want. This method of applying options as SAS starts, eliminates risky techniques of the past when developers would update the configuration file provided by SAS. The technique is described and demonstrated with examples for the Microsoft Windows environment for foundation SAS, but the issues are very similar if launching SAS clients and servers on Unix(including Unix on zOS). There is a small overhead with this proposal, but if you review the paper, I'm sure you will consider it worthy enough to give it a try...

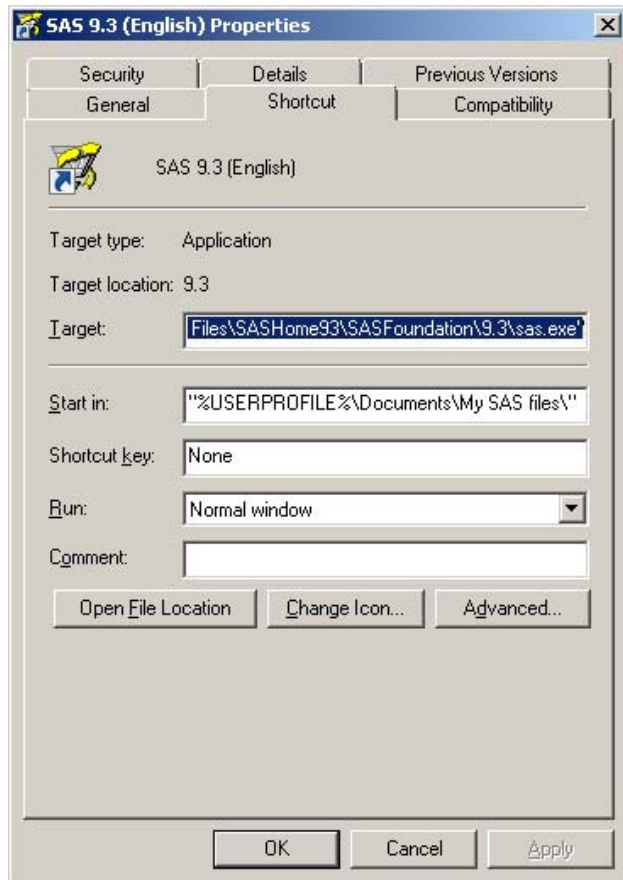
INTRODUCTION

This paper provides some ideas to manage the many differing uses we make of the SAS System in our computing environments. The ideas are intended to make supporting these varied uses by simplifying the standard install.

This paper is not expected to support raw users of SAS. An understanding of the SAS System and the platform on which you operate SAS is assumed. References to on-line help are made throughout the paper.

STANDARD DEFAULT INSTALLATION

The default, for a simple example - on my laptop with MS-Windows VISTA, the properties of the installed launch SAS icon:



Launch Icon Properties 1

472-2013 Configure your foundation SAS(r) client easily and without risk

The "Target" (command line) is obviously too long to appear in this screen snip, so I have repeated it here

```
C:\Program Files\SASHome93\SASFoundation\9.3\sas.exe" -CONFIG "C:\Program
Files\SASHome93\SASFoundation\9.3\nls\en\sasv9.cfg")
```

It contains the path to SAS.EXE as well as the path and name of a configuration file established when SAS was installed on this laptop. Being set up in this way hides one of the elegant features of the SAS System – how it launches:

FIRST CRITICISM

This installed default specifies an inflexible launch:

Elegance would introduce not only brevity, but also flexibility which I (and soon you will too, I hope) consider really helpful - and too good to miss. Time lost at this stage has cost a lot of wasted effort (imho!).

PRIMARY SOLUTION

Remove all the options that follow "sas.exe", into a file.

How you name that file and where you place that file is very important.

The on-line documentation "SAS(R) 9.3 Companion for Windows" describes the way SAS finds configuration files (search the on-line doc for "[How SAS Finds and Processes Configuration Files](#)". You can drill down to that page through the SAS9.3 documentation, as in this screen snip of the "SAS® 9.3 Companion for Windows" at <http://support.sas.com/documentation/cdl/en/hostwin/63047/HTML/default/titlepage.htm>: After selecting book "SAS(R) 9.3 Companion for Windows", the menu selections are

Running SAS under Windows,

Getting Started under Windows

Files Used by SAS

and in the page found

How SAS Finds and Processes Configuration Files

The screenshot shows the SAS(R) 9.3 Companion for Windows documentation page. The left sidebar contains a table of contents with the following items:

- PDF
- Contents
- About
- Title Page
- What's New in the SAS 9.3 Windows Companion
- Accessibility Features in SAS under Windows
- About This Book
- Running SAS under Windows
 - Getting Started under Windows
 - SAS: Exploiting the Power of Windows
 - Starting SAS
 - Files Used by SAS
 - Submitting SAS Code
 - Interrupting Your SAS Session

The main content area shows the following sections:

- Previous Page | Next Page
- Files Used by SAS
- Introduction to Files Used by SAS
- SAS Configuration Files
 - Purpose of Configuration Files
 - The Default Configuration File
 - Specifying System Options in a Configuration File
 - Creating a Customized Configuration File
 - Starting SAS with an Alternate Configuration File
 - How SAS Finds and Processes Configuration Files

On-Line Doc 1

472-2013 Configure your foundation SAS(r) client easily and without risk

Unfortunately at this time (end-October 2012) that web page of documentation also describes creating your own configuration file by copying parts from the installation configuration.

This should be unnecessary with the elegance of the SAS System at versions since 9.0 (and possibly earlier but I cannot confirm that any longer). This misfortune does not invalidate the rest of the page which is really useful – just re-consider using the section “Creating a Customized Configuration File”.

Once you understand “How SAS Finds and Processes Configuration Files” you will appreciate the easy – and risk-free technique proposed here.

STAGE TWO

Create a text file named ‘sasv9.cfg’ in your home folder (\My Documents in win-xp, “documents” on Vista and win-7)

You can leave it empty, but I found the two most valuable additions were

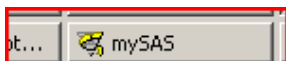
```
-awsTitle mySAS
-verbose
```

The first option will confirm for me that the approach has taken effect.



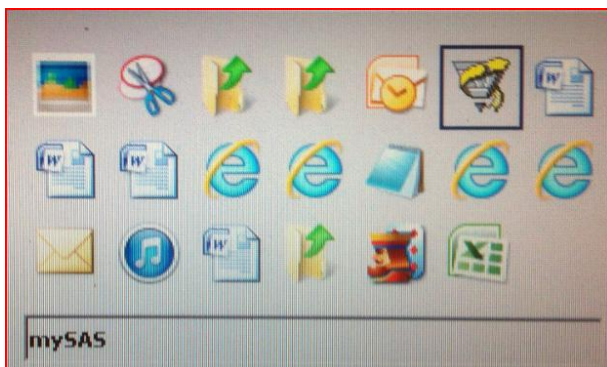
Identifying My SAS Session 1

The “mySAS” label will appear with the SAS icon, on the taskbar like



Identifying My SAS Session 2

Or, when I want to switch from one application to another with the alt+tab keyboard combination, the “mySAS” label appears with the icon, like



Identifying My SAS Session 3

The second option (-VERBOSE) dumps a lot of options into the SASlog – but, most importantly, it reports all the configuration files that have been used – (or not used when there is some mistake that I make).

(historical note: my methods for starting SAS have been “under development” for some time (30+ years) and it used to be that

```
proc options option=config ;
run ;
```

did not report all the configuration files that had been used – just the last one (even earlier around v7, the value would be empty!)

(end of historical note)

Here is how my configuration files information is reported by the VERBOSE option.

472-2013 Configure your foundation SAS(r) client easily and without risk

```
==== Processed Configuration File(s) ====
C:\Program Files\SASHome93\SASFoundation\9.3\sasv9.cfg
C:\Program Files\SASHome93\SASFoundation\9.3\nls\en\sasv9.cfg
C:\Users\Peter Crawford\Documents\sasv9.cfg
C:\Users\Peter Crawford\Documents\my sas files\9.3\sasv9.cfg
```

An explanation of the ordering of this list should be available at the link [How SAS Finds and Processes Configuration Files](#) mentioned above.

Notice that each file in the list has the same name “sasv9.cfg”. It is an important part of this facility, that the name is one of those expected and that the file appears in one of the locations defined in the documentation.

These files serve different purposes:

The first and second define the “install of SAS”. They provide paths to the executable application and also to helpful data – like the SASHELP library.

STAGE THREE

Implement personal preferences:

After naming your personal environment (-AWSTITLE) and confirming your configuration (-VERBOSE) consider two types of extension:

1. Personal environment
2. Team working

We will return to these extensions of the SAS-launch in the next section of paper.

MORE BENEFITS OF THE MAIN IDEA

This section of the paper shows how this “Risk-Free Flexible SAS launch” supports most of the invocation of SAS you are likely to need. The paper looks at

1. Supporting different users with one launch icon
2. Launching batch processes
3. Concurrent batch processes without contention
4. Supporting multiple sessions for the current user
5. Team network sharing
6. SAS Server sessions

1 SUPPORTING DIFFERENT USERS WITH THE SAME LAUNCH ICON

Since the icon refers only to SAS.EXE with no parameters, how can it support multiple users (on the same or different machines)?

The icon only needs to start in a personal folder to become “user-specific” – so set the “Start in” folder to \my documents (or my preference) \My Documents\My SAS Files\. Of course that is terminology that changed after Microsoft Windows XP. A continuing reference is %USERPROFILE%. This points to the parent of the equivalent Documents folder- eliminating the need to discover or define the structure administrators implement in a “standard” desktop environment.

The same machine with one icon shared among all users to start SAS only needs this convention (a %userprofile% based “Start In” folder) and a personal configuration file in that folder, to support independent user preferences.

2 LAUNCHING BATCH PROCESSES

Following the theme above, for multiple users, the launch icon for a batch process only needs to start in its own folder – it too needs no options on the launch command.

In that “Start In” folder the config file (named sasv9.cfg) will contain all the options that might appear on a traditional Launch-SAS command, in particular

```
-SYSIN path\name_of_program_to_run_in_batch.sas
```

472-2013 Configure your foundation SAS(r) client easily and without risk

```
-LOG      path\name.log
```

Introduced with SAS version 9 are some really useful options to support LOG files: If they are unfamiliar, and you use SAS for regular processes as well as “ad-hoc processes under development” these are options (below) have features that you might consider justify attending this conference. Review

```
-LOGPARM
```

Although established to support SAS servers, the LOGPARM option supports the old option \$UNBUFLOG and inserting process-id and/or timestamp information into the LOG/ALTLOG file names. More relevant to SAS servers than SAS user sessions, it also enables restarting the log in a new file on an appropriate frequency.

I use the time-stamping facility to ensure I never lose a log for any processes I would describe as “production”.

3 CONCURRENT BATCH PROCESSES WITHOUT CONTENTION

The approach I recomAs SAS processes run, I need each isolated from all others to reduce the experience of inaccessible files and catalogs and registry itemstores.

The approach described above can be refined to help. If you want a common SAS registry and common SASUSER area, just add to the config file for each batch process:

```
-SASUSER      "path_to the appropriate shared SASuser folder\"
-RSASUSER     /* ENSURE SASUSER is accessed only READ ONLY*/
```

Should a batch process need its own SASUSER and SAS registry to be updateable,

```
-SASUSER      "batch process specific folder\"
-NORSASUSER   /* ENSURE SASUSER is private*/
```

When there is a common tree for paths, the config file can help with a shortcut environment variable.

```
-SET          shCutName  "path for application\"
-SASUSER     !shCutName\sasuser  /* app- specific sasuser*/
```

4 SUPPORTING MULTIPLE SESSIONS FOR THE CURRENT USER

As with concurrent batch, concurrent user sessions need a launch icon for each. As well as having a separate launch folder, it will probably also need a separate SASUSER areas – easily created by copying from the normal session location – but only make that copy when the SAS session isn’t running. The most portable location for the SASUSER would be a sub-directory of the launch folder.

5 SHARING A TEAM NETWORK AREA

There are very few difficulties to overcome that you have not already solved!

The remaining issue relates to the environment set-up:

have you a common drive mapping?

Without this, users of the service must take responsibility for adapting “connections” for their view of the shared area. These connections will be used as LIBNAME and FILENAME paths. Most convenient (I find) are assignments made in a configuration file that will only become “active” when used. (for a quick demonstration, consider the LIBREF sampsio. A library list shortly after SAS starts is unlikely to include sampsio. However, you can use it. The library SASHELP has no employee table, but you can execute

```
PROC CONTENTS DATA= sampsio.employee ;
RUN ;
```

This library connects to the data by providing the path in the value of an environment variable. These are most easily defined in a configuration file with a statement like

```
-SET          TEAM1     "path for common data area\"
```

472-2013 Configure your foundation SAS(r) client easily and without risk

You might like to prepare several of these, some personal to a user for sharing their data or code. Others would be set up as read-only.

Because a library connected in this way is only added to dictionary tables once it has been used in code, you may add far more than needed with no performance penalty.

So I am recommending that you share a configuration file (in a common data area) which holds “team parameters”. These could include more than just these common connection definitions, but that is the primary purpose.

Examples of shared data, includes not only SAS datasets, but also format catalogs and ODS template itemstores,

This could expand into a whole topic in its own paper and presentation..... But time presses.

6 SAS SERVER SESSIONS

Administering a stand-alone SAS foundation in these ways provides a test-bed to develop ideas for bigger environments.

The invocation of a service (SAS SHARE, SAS WORKSPACE SERVER, SAS OBJECT SERVER, SAS REMOTE CONNECT SERVER) usually involves many parameters and all the default installation examples that I have seen, involve extremely long command lines. When you have responsibility for maintaining these services, you might feel more secure in accepting a fully working implementation delivered by SAS Institute without considering all these options. (However, I never heard that ignorance was an aid to security -- this is my opinion and no one else's). Fully appreciating each of the options applied as SAS starts a server, will enable an administrator to adapt and maintain servers with less risk. Having each of these parameters in a configuration file specific to the service, well – it seems like a step towards the risk-free environment we all seek. Just launch SAS.EXE in the appropriate folder (it holds a sasv9.cfg file with all necessary parameters).

LAUNCHING SAS SESSIONS ON UNIX

Although the arrangement of SAS executables on Unix platforms is different from the arrangement on Microsoft Windows platforms, the technique described here reduces that issue, because that part is not touched by the suggestions made here. Just place a configuration file in the folder where SAS starts (for example, a user's home folder ~/) and that configuration file will be used in addition to the configuration created when SAS was installed.

The concept will require a separate folder for every batch process and every SAS server to be instantiated with the flexibility offered by this technique. The “trade-off” is that no longer will your support administrators (or you) need to pour over very, very long command lines.

AFTER UPGRADING SAS

Once you have installed a new release of SAS, the solutions described here remain valid as long as you ensure new “launch icons” refer to the new path for the new SAS.EXE – all other features remain. As for upgraded features – you might wish to discover how compatibly the old and new releases work with SASUSER.PROFILE, format catalogs, metadata server, item stores holding ODS templates and datasets too – but all that is a topic for a migration paper – and the concept here - SAS.EXE with no options – remains valid and effective.

CONCLUSION

This paper will remain a work-in-progress for me, but I hope you are going to find some nuggets – both large and small and perhaps golden – to add value and convenience to your working environment launching SAS processes – and reduce risk, by removing the need to copy and change the SAS-install version of its configuration file.

REFERENCES

ACKNOWLEDGMENTS

This is the text for the acknowledgments.

RECOMMENDED READING <HEADING 1>

- For an excellent work on SAS System options, read *Denise M.Poll's* [Paper 240-2011:SAS® Options: Versatile Players in the Game of SAS](#)

472-2013 Configure your foundation SAS(r) client easily and without risk

- *Base SAS® Procedures Guide*
- *SAS® For Dummies®*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name:	Peter Crawford
Enterprise:	Crawford Software Consultancy limited
Address:	London, UK
E-mail:	Peter.Crawford@blueyonder.co.uk

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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