

A Configuration File Companion: testing and using environment variables and options; templates for startup-only options `initstmt` and `termstmt`

Ronald J. Fehd, Stakana Analytics

Abstract

description : The startup process of SAS® software reads one or more configuration files, *.cfg, which have allocations of environment variables, the values of which are used in SAS startup-only options to provide access to libraries, sets of folders that contain files that SAS uses for functions, macros, and procedures.

purpose : This paper provides programmers and advanced users programs to review the default configuration files; procedures, options, and sql to discover options; and a suite of programs to use in Test-Driven Development (TDD) to trace and verify user-written configuration files.

audience : developers, programmers, advanced users

keywords : API: FOLDERID_Documents;
CSIDL Equivalents: CSIDL_MYDOCUMENTS, CSIDL_PERSONAL
sql dictionary.options, opstart='startup';
environment variables: mysasfiles, path, sasautos, sascfg, sasroot;
batch files: sas.cmd, 01-sas-v9-cfg-test.*;
startup-only options:
 echoauto, initstmt, sysin, termstmt, verbose

In this paper	Bricolage: tools for discovery	3
	values of opstart	3
	echoing configuration files	4
	sql procedure, dictionary.options	5
	Setup for testing	6
	Unit test: sasv9.cfg	9
	Template for initstmt and termstmt	12
	Unit test: autoexec.sas	14
	Summary	16
	References	17

Introduction

overview

This is the overview, which consists of a list of topics in this section.

- what's inside configuration files
- startup process
- why use configuration files

what's inside

Configuration files contain two types of statements: allocation of environment variables and assignment of options. The main configuration file contains over 200 lines; more than half of those lines contain references to the environment variable `sasroot` in environment variables' and options' assignments.

startup process

The startup process of SAS software has two insertion points for startup-only options: on the command line, and in configuration files. Table 1 lists the insertion points and the startup-only options.

Table 1 startup process

1. command-line:		<code>sas -sysin job-name <-options></code>	
2. configuration files:			
(a) primary:		<code>!sasroot/sasv9.cfg</code>	
(b) main:	required	<code>!sasroot/nls/??/sasv9.cfg</code>	
(c) tertiary: project:		<code>sasv9.cfg</code>	
3. options from command-line			
code-insertion points: startup-only options			
<u>option name</u>	<u>default</u>	<u>where set</u>	<u>assignment</u>
<code>config</code>	<code>sasv9.cfg</code>		<code>-config 'filename.ext'</code>
<code>autoexec</code>	<code>autoexec.sas</code>		<code>-autoexec 'filename.ext'</code>
<code>initstmt</code>		command-line or config	<code>-initstmt '*text;'</code>
<code>sysparm</code>		command line:	<code>-sysparm "text"</code>
<code>sysin</code>		command line:	<code>-sysin job-name</code>
<code>sysparm</code>		macro variable, in program	<code>%let sysparm=value;</code>
<code>termstmt</code>		command-line or config	<code>-termstmt '*text;'</code>

Description of this process is found in SAS Documentation under the topic: Files Used by SAS.

Your installation may have several folders under National Language Support (nls); English (en) is shown here. See also the option `locale`.

why use config?

Table 2 shows that almost one third of options are startup-only.

Table 2 frequency of optstart

optstart	Option Set		Cumulative	Cumulative
	Frequency	Percent	Frequency	Percent
<code>anytime</code>	315	67.74	315	67.74
<code>startup</code>	150	32.26	465	100.00

Program 2, `proc freq`, tables `optstart`, pg. 5 produces this frequency table.

Bricolage: tools for discovery

values of opstart

overview

There are two values of opstart: anytime and startup.
In the output of program proc-options-define-value-config.sas these values are seen in the output table in the log in the section When Can Set:

When Can Set: Startup or anytime during the SAS Session

When Can Set: Session startup (command line or config) only

proc-options-define-value-config.sas

```
*name: proc-options-define-value-config.sas;  
proc options define value option=config;  
run;
```

log: see line 23, for optstart='startup'

```
1 Option Value Information For SAS Option CONFIG  
2 Value: ( "C:\program-files\SASHome\SASFoundation\9.4\sasv9.cfg"  
3         "C:\program-files\SASHome\SASFoundation\9.4\nls\en\sasv9.cfg"  
4         "C:\SAS-projects\2021-book\sas-startup-configuration\sasv9.cfg"  
5         )  
6 Scope: SAS Session  
7 How option value set: Internal  
8  
9 Option Definition Information for SAS Option CONFIG  
10 Group= INSTALL  
11 Group Description: Site installation settings  
12 Description: Specifies the configuration file that is used  
13 when initializing or overriding the values of SAS system options.  
14 Type: The option value is of type CHARACTER  
15 Maximum Number of Characters: 32000  
16 Casing: The option value is retained with original casing  
17 Quotes: If present during "set", start and end quotes are removed  
18 Parentheses: The option value does not require enclosure  
19 within parentheses.  
20 If present, the parentheses are retained.  
21 Expansion: Environment variables, within the option value,  
22 are not expanded  
23 When Can Set: Session startup (command line or config) only
```

echoing configuration files

overview

This is the overview, which consists of a list of programs in this section.

- config syntax
 - echo any file
 - echo config primary
 - echo config main
-

config syntax

While viewing any configuration file, *.cfg, look for these two statements:

- allocate environment variable: `-set evar value`
 - assign option: `-<option-name> value`
-

echo-any-file

```
*name: echo-any-file.sas;
%put echo &in_file;
%let lrecl = 80;
DATA _null_;
    infile "&in_file"    lrecl = &lrecl    pad
                    end    = endofile;

do until(endofile);
    input @1 line $char&lrecl.;
    put   @1 line $char&lrecl.;
end;
stop;
run;
```

echo-config-primary

```
*name: echo-config-primary.sas;
%let in_file = %sysfunc(sysget(sasroot))\sasv9.cfg;
%include 'echo-any-file.sas';
```

```
log: echo IN_FILE=C:\program-files\SASHome\SASFoundation\9.?\sasv9.cfg
NOTE: The infile "C:\program-files\SASHome\SASFoundation\9.?\sasv9.cfg" is:
      Filename=C:\program-files\SASHome\SASFoundation\9.?\sasv9.cfg,
      -config "C:\program-files\SASHome\SASFoundation\9.?\nls\en\sasv9.cfg"
```

echo-config-main

Program 1 echo-config-main.sas

```
*name: echo-config-main.sas;
%let in_file = %sysfunc(sysget(sascfg))\sasv9.cfg;
%include 'echo-any-file.sas';
```

```
log: echo IN_FILE= C:\program-files\SASHome\SASFoundation\9.?\nls\en\sasv9.cfg
...
-SET SASROOT "C:\program-files\SASHome\SASFoundation\9.?"
...
-SET SASAUTOS ("!SASROOT\core\sasmacro"
...
              "!SASROOT\graph\sasmacro"
              "!SASROOT\stat\sasmacro"
              )
```

notes: The environment variable `sasroot` is used in the environment variable `sasautos`, which is the list of folders of the *fileref* `sasautos`, which is the default value of option `sasautos`.

Note that lists of values, e.g. `sasautos`, are enclosed in parentheses.

sql procedure, dictionary.options

overview

This is the overview, which consists of a list of topics in this section.

- describe dictionary options
- freq opstart

OnLine Doc: Customizing Your SAS Session by Using Configuration and Autoexec Files

describe dictionary options

```
*name: proc-sql-describe-dictionary-options.sas;
proc sql; describe table dictionary.options;
quit;
```

```
log: create table dictionary.options
      optname char(32) label='Option Name',
      opttype char(8) label='Option type',
      offset num label='Offset into option value',
      setting char(1024) label='Option Setting',
      optdesc char(160) label='Option Description',
      level char(8) label='Option Location',
      optstart char(8) label='Option Set',
      group char(32) label='Option Group'
```

freq optstart

Program 2 was used to produce table 2, shown above on pg. 2.

Program 2 proc freq, tables optstart

```
*name: proc-freq-d-options-opstart.sas;
PROC sql; create table dictionary_options_optstart as
      select optstart
      from dictionary.options;
quit;

PROC freq data = &syslast;
      tables optstart;
run;
```

Setup for testing overview

This is the overview, which consists of a list of topics in this section. This section shows how to create a suite of files for a unit test of a user-written configuration file.

- functions used here
- write test suite, config
- sas.cmd
- sasv9.cfg
- autoexec.sas
- initstmt.sas
- termstmt.sas
- 01-config-test.*
- 02-autoexec-test.*

functions used here

These functions are used in the following programs.

- `getoption(option-name<,options>)` fetch value of option
 - `putn(value,format)` echo computed value
 - `%scan(string,n,delimiter)` fetch word of string
 - `%sysfunc(fn(...))` access to data-step functions
 - `%sysget(evar)` fetch value of environment variable (evar)
-

write-test-suite- config.sas

Program 3 writes to the specified folder a set of files for this article.

Program 3 write test suite of config test files

```
*name: write-test-suite-config.sas;
%let folder=c:\temp\sas-test;
%let folder=..\sas-startup-config-test-suite;
DATA _null_;
    file "&folder\sas.cmd";
put    'rem name: sas.cmd '
/    '"" "%sysget(sasroot)\sas.exe" "" %*';

    file "&folder\01-config-test.bat";
put    'rem name: 01-config-test.bat'
/    'sas 01-config-test -verbose -pagesize max';

    file "&folder\01-config-test.sas";
put    '*name: 01-config-test.sas';

    file "&folder\sasv9.cfg";
put    '/*name: sasv9.cfg */'
/    "-sasinitialfolder '.'"
/    "-set site_inc 'C:\SAS-projects\SAS-site\sas-includes'"
/    "-set site_inc '.'"
/    '-initstmt "" "%include '!site_inc\initstmt.sas';" ""'
/    '-termstmt "" "%include '!site_inc\termstmt.sas';" ""';

    file "&folder\02-autoexec-test.bat";
put    'rem name: 02-autoexec-test.bat'
/    'sas 02-autoexec-test -echoauto -source -source2';

    file "&folder\02-autoexec-test.sas";
put    '*name: 02-autoexec-test.sas;'
/    'options pagesize=max;'
/    'filename _all_ list;'
/    'libname _all_ list;'
/    '%put echo mautosource=%sysfunc(getoption(mautosource));'
/    '%put %sysfunc(getoption(sasautos,keyword));'
/    '%let mvar2=text.2 for show sort order;'
/    '%let mvar1=text.1 for termstmt;';

    file "&folder\autoexec.sas";
put    '*name: autoexec.sas;'
/    "title1 '???' new project ???';"
/    "filename project '.';"
/    "libname library '.';";

    file "&folder\initstmt.sas";
put    '*name: initstmt.sas;'
/    '%put echo %sysfunc(getoption(initstmt));';

    file "&folder\termstmt.sas";
put    '*name: termstmt.sas;'
/    '%put echo %sysfunc(getoption(termstmt,keyword));';

stop;
run;
```

sas.cmd	<p>File <code>sas.cmd</code> centralizes the call to <code>sas.exe</code>.</p> <pre>rem name: sas.cmd "C:\...\SASHome\SASFoundation\9.?\sas.exe" %*</pre> <hr/> <p>notes: Percent asterisk (%) is the DOS syntax for 'pass all command-line parameters to program'. DOS strings are numbered: %1 ... %n ; asterisk means 'all'.</p> <hr/>
sasv9.cfg	<p>File <code>sasv9.cfg</code> replicates the syntax of the main configuration file by providing an environment variable <code>site_inc</code> which is then used in options <code>initstmt</code> and <code>termstmt</code>.</p> <p>Program 4 config test file: sasv9.cfg</p> <pre>/*name: sasv9.cfg */ -sasinitialfolder '.' -set site_inc 'C:\SAS-projects\SAS-site\sas-includes' -set site_inc '.' -initstmt "%include '!site_inc\initstmt.sas';" -termstmt "%include '!site_inc\termstmt.sas';"</pre> <hr/> <p>notes: Option <code>sasinitialfolder</code> is the argument to Display Manager command: File Save As; dot means 'here': current folder. Environment variable <code>site_inc</code> has two values as a reminder that you may, in the future, wish to provide access to subroutine files <code>initstmt.sas</code> and <code>termstmt.sas</code> for all projects in a central folder. The arguments of both contain a reference to the evar <code>!site_inc</code>; macro variables are referenced with an ampersand (&); evars are referenced with a bang: exclamation point (!).</p> <hr/>
autoexec.sas	<pre>*name: autoexec.sas; title1 '??? new project ???'; filename project '.'; libname library '.';</pre> <hr/> <p>notes: Options <code>mautosource</code> and <code>sasautos</code> are the pair of options of the <i>autocall</i> of macros.</p> <hr/>
initstmt.sas	<p>Files <code>initstmt.sas</code> and <code>termstmt.sas</code></p> <pre>*name: initstmt.sas; %put echo %sysfunc(getoption(initstmt));</pre> <hr/>
termstmt.sas	<pre>*name: termstmt.sas; %put echo %sysfunc(getoption(termstmt,keyexpand));</pre> <hr/> <p>notes: Function <code>getoption</code> has a required option of <code>option-name</code> and an optional argument of <code>keyexpand</code> which is used to format the value; both options contain an evar reference: <code>!site_inc</code>, which will be resolved when echoed.</p> <hr/>

**Unit test: sasv9.cfg
overview**

This is the overview, which consists of a list of topics in this section.

- 01-config-test.*
- sections in 01-config-test
- 01-config-test.log, pg. 1, 2, 3, 4, 12

01-config-test.*

Files 01-config-test.bat and 01-config-test.sas are used for a unit test of the configuration file sasv9.cfg.

Program 5 01-config-test.*

```
rem name: 01-config-test.bat
sas 01-config-test -verbose -pagesize max

*name: 01-config-test.sas;
```

notes: Option `-verbose` writes the settings of the system options to the log. Option `-pagesize max` removes page breaks from the log, which may be 500+ lines; at 50 lines/page, more than 10 pages.

**sections in
01-config-test**

- Options specified in the config files:
- Options set internally at initialization:
- Options specified on the command line:
- Options set internally at initialization:
- Options set internally during session startup:
- Options set from locale:
- Options set internally at initialization:

**01-config-test.log,
pg. 1**

```
23 Options specified in the config file
24     C:\program-files\SASHome\SASFoundation\9.4\sasv9.cfg:
25 -----
26 CONFIG = "C:\program-files\SASHome\SASFoundation\9.4\nls\en\sasv9.cfg"
27 -----
```

notes: line 26, primary config file:
compare to listing from program echo-config-main.sas, pg. 4;

```
28 Options specified in the config file
29     C:\program-files\SASHome\SASFoundation\9.4\nls\en\sasv9.cfg:
30 -----
31 ...
32 SET = SASROOT "C:\program-files\SASHome\SASFoundation\9.4"
33 JREOPTIONS = (-DPFS_TEMPLATE=!SASROOT\tkjava\sasmisc\qrpfstpt.xml

...
46 SET = SASAUTOS ("!SASROOT\core\sasmacro"
47 ...
48             "!SASROOT\graph\sasmacro"
49             "!SASROOT\stat\sasmacro"
50             )
```

notes: Lines 32–33 show the config file allocating an environment variable, `sasroot` and using it in an option, `jreoptions`; lines 46–50 show its use in another environment variable, `sasautos`, which is the the list of folders for the *fileref* and option `sasautos`. Note the list of folders is enclosed in parentheses.

01-config-test.log,
pg 2

```
88 SET = MYSASFILES "?FOLDERID_Documents\My SAS Files\9.4"  
89 SASUSER = "?FOLDERID_Documents\My SAS Files\9.4"  
90 WORK = "!TEMP\SAS Temporary Files"  
91 ...  
92 SET = SASCFG "C:\program-files\SASHome\SASFoundation\9.4\nls\en"
```

notes: ?FOLDERID_Documents line 88–89:

question mark (?) is special character used to reference an API

!TEMP line 92, is a reference to a Windows environment variable

! → In previous versions, my-SAS-files SAS-user values fetched value of CSIDL*

01-config-test.log,
pg 3

```
...  
-----  
111  
112 Options specified in the config file  
113 C:\SAS-projects\2021-book\sas-startup-config-test-suite\sasv9.cfg:  
114 -----  
115 SASINITIALFOLDER = '.',  
116 SET = site_inc 'C:\SAS-projects\SAS-site\sas-includes'  
117 SET = site_inc '.',  
118 INITSTMT = "%include '!site_inc\initstmt.sas';"  
119 TERMSTMT = "%include '!site_inc\termstmt.sas';"  
120 -----  
121 Options set internally at initialization:  
122 -----  
123 CONFIG = ( "C:\program-files\SASHome\SASFoundation\9.4\sasv9.cfg"  
124 "C:\program-files\SASHome\SASFoundation\9.4\nls\en\sasv9.cfg"  
125 "C:\SAS-projects\2021-book\sas-startup-config-test-suite\sasv9.cfg" )  
126 -----  
127 Options specified on the command line:  
128 -----  
129 SYSIN = 01-config-test  
130 VERBOSE  
131 PAGESIZE = max  
132 -----  
133 Options set internally at initialization:  
134 -----  
135 AUTOEXEC = "C:\SAS-projects\2021-book\sas-startup-config-test-suite\autoexec.sas"
```

notes: Options set internally at initialization:

123–125: the list of configuration files and order in which they are 'loaded',

! → note the use of the SAS HelpDesk verb *loaded*:

this means the last evar allocation or option assignment is the value used

Options specified on the command line:

129–131: command-line assignments 'loaded' after configuration file assignments

Options set internally at initialization:

135: option autoexec is set to value of sasinitialfolder
plus default filename: autoexec.sas

**01-config-test.log,
pg. 4**

```
155 NOTE: Copyright (c) 2002-2012 by SAS Institute Inc., Cary, NC, USA.
156 NOTE: SAS (r) Proprietary Software 9.?
157 ...
158 NOTE: SAS initialization used:
159     real time          2.55 seconds
160
161 ===== Processed Configuration File(s) =====
162 C:\program-files\SASHome\SASFoundation\9.4\sasv9.cfg
163 C:\program-files\SASHome\SASFoundation\9.4\nls\en\sasv9.cfg
164 C:\SAS-projects\2021-book\sas-startup-config-test-suite\sasv9.cfg
165
166 ===== Environment Variable Options =====
167 <none>
168
169 Option      Value
170 =====
171 AUTOEXEC    C:\SAS-projects\2021-book\sas-startup-config-test-suite\autoexec.sas
172 ...
```

notes: beginning of notes from option sysin=01-config-test.sas
Processed Configuration File(s)
162–164: compare to page 3, lines 123–125

**01-config-test.log,
pg. 12**

```
550 NOTE: AUTOEXEC processing beginning; file is
551     C:\SAS-projects\2021-book\sas-startup-config-test-suite\autoexec.sas.
552
553 NOTE: Libref LIBRARY was successfully assigned as follows:
554     Engine:          V9
555     Physical Name: C:\SAS-projects\2021-book\sas-startup-config-test-suite
556
557 NOTE: AUTOEXEC processing completed.
558
559 echo %include '!site_inc\initstmt.sas'
560 1          *name: 01-config-test.sas;
561
562 echo TERMSTMT=%include '\termstmt.sas'
563 NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
564 NOTE: The SAS System used:
565     real time          2.60 seconds
566     cpu time           0.46 seconds
```

notes:
550–557 option autoexec processing is
%include 'autoexec.sas'\nosource2;
there is no echo of the filename statement
but any *libref(s)* allocated are duly noted
559 compare to termstmt, line 562
560 line 1 from sysin
562 compare to initstmt, line 559: keyexpand produces TERMSTMT=

The next section shows polishing of echo from options initstmt and termstmt.

Template for initstmt and termstmt

overview

This section explains how to use the start-up only options `initstmt`, which statements are executed after the `autoexec`, and `termstmt`, which statements are executed at `endsas`. The task here is to replicate the beginning ... completed notes like those in `autoexec` processing. This is the list of topics in this section.

- `autoexec` notes
- syntax
- `initstmt.sas` and log
- `termstmt.sas` and log

autoexec notes

```
NOTE: AUTOEXEC processing beginning; file is
      C:\SAS-projects\2021-book\sas-startup-config\autoexec.sas.
...
NOTE: AUTOEXEC processing completed.
```

syntax

semicolon Task 1 is how to write a note with a semicolon in it. This is accomplished by enclosing the semicolon and CR/LF in the `%str()` function:
`%str(;`
`)`

%scan Task 2 is parsing the value of the options, which contains two words. The `initstmt` value is `'%include "file-specification";'`. The `%scan(string,n,delimiters)` function is used to pick the 2nd word; the `%sysfunc(getoption(option-name))` pair of functions is used to fetch the value of the option.

job info Task 3 is to write a set of notes containing job information: startup date+time-stamp, `userid`, and `job-name`; this information is necessary to match the log with the program when parsing logs of program runs.

time Task 4: The read-only macro variable `sysprocessid` is a 32-character hexadecimal string, the first 16 characters of which contain the program startup date+time-stamp.

initstmt.sas

```
* name initstmt.sas;
%put Note initstmt processing beginning%str(
)file is %scan(%sysfunc(getoption(initstmt)),2,%str( ));

%let _string = JOB-INFO;
%put &_string datetime_hex16=%substr(&sysprocessid,1,16);
%put &_string datetime=%sysfunc
      (putn(%substr(&sysprocessid,1,16)x,datetime21.2));
%put &_string &=sysuserid;
%put &_string %sysfunc(getoption(sysin,keyword)) #;
%symdel _string;

%put Note initstmt processing completed.;
```

log

```
Note: initstmt processing beginning; file is '!site_inc\initstmt.sas'.
JOB-INFO: datetime_hex16=40D5944000000000
JOB-INFO: datetime=31JUN2021:11:23:58.13
JOB-INFO: SYSUSERID=Ronald
JOB-INFO: SYSIN=C:\SAS-projects\...\02-autoexec-test.sas #
Note: initstmt processing completed
```

termstmt.sas

```
* name: termstmt.sas;
%put Note: termstmt processing beginning%str(
)file is %scan(%sysfunc(getoption(termstmt,keyexpand)),2,%str( ));

%put echo list global _user_ mvars, if any;
%put _user_;
*include '!site_inc\echo-mvars.sas';

%put Note: termstmt processing completed.;
```

log

```
Note: termstmt processing beginning; file is '.\termstmt.sas'.
echo list global _user_ mvars, if any
Note: termstmt processing completed
```

Unit test: autoexec.sas

overview

This section shows the programs and log.

- 02-autoexec-test.*
- 02-autoexec-test.log, initstmt
- 02-autoexec-test.log, program
- 02-autoexec-test.log, termstmt

02-autoexec-test.*

```
rem name: 02-autoexec-test.bat
sas 02-autoexec-test -echoauto -source -source2
```

```
*name: 02-autoexec-test.sas;
options pagesize=max;
filename _all_ list;
libname _all_ list;
%put echo mautosource=%sysfunc(getoption(mautosource));
%put %sysfunc(getoption(sasautos,keyword));
%let mvar2=text.2 for show sort order;
%let mvar1=text.1 for termstmt;
```

notes: Options `-echoauto` and `-source` turn on the display of statements in the autoexec file.

Option `-source2` turns on the display of statements of `%include` files; see files `initstmt.sas` and `termstmt.sas`.

02-autoexec-test.log, initstmt

```
NOTE: %INCLUDE (level 1) file !site_inc\initstmt.sas is file
      C:\SAS-projects\2021-book\sas-startup-config\initstmt.sas.
1      ** name: initstmt.sas;
2      +%put Note: initstmt processing beginning%str(
3      +)file is %scan(%sysfunc(getoption(initstmt)),2,%str( ));
Note: initstmt processing beginning; file is '!site_inc\initstmt.sas'
4      +
5      +%let _string = JOB-INFO;
6      +%put &_string: datetime_hex16=%substr(&sysprocessid,1,16);
JOB-INFO: datetime_hex16=41DCF8DA18B73B64
7      +%put &_string: datetime=%sysfunc
8      + (putn(%substr(&sysprocessid,1,16)x,datetime21.2));
JOB-INFO: datetime=11AUG2021:06:40:34.00
9      +%put &_string: &=sysuserid;
JOB-INFO: SYSUSERID=Ronald
10     +%put &_string: %sysfunc(getoption(sysin,keyword)) #;
JOB-INFO: SYSIN=C:\SAS-projects\2021-book\sas-startup-config\02-autoexec-test.sas #
11     +%syndel _string;
12     +
13     +%put Note: initstmt processing completed.;
Note: initstmt processing completed.
NOTE: %INCLUDE (level 1) ending.
```

02-autoexec-test.log, program

```
1          *name: 02-autoexec-test.sas;
2          options pagesize=max;
3          filename _all_ list;
NOTE: Fileref= PROJECT
      Physical Name= C:\SAS-projects\2021-book\sas-startup-configuration
4          libname _all_ list;
NOTE: Libref= LIBRARY
      Scope=      Program 02-autoexec-test
      Engine=     V9
      Physical Name= C:\SAS-projects\2021-book\sas-startup-configuration
      Filename= C:\SAS-projects\2021-book\sas-startup-configuration
...
5          %put echo mautosource=%sysfunc(getoption(mautosource));
echo mautosource=MAUTOSOURCE
6          %put %sysfunc(getoption(sasautos,keyword));
SASAUTOS=SASAUTOS
7          %let mvar2=text.2 for show sort order;
8          %let mvar1=text.1 for termstmt;
```

02-autoexec-test.log, termstmt

```
NOTE: %INCLUDE (level 1) file !site_inc\termstmt.sas is file
      C:\SAS-projects\2021-book\sas-startup-configuration\termstmt.sas.
2          ** name: termstmt.sas;
3          +%put Note: termstmt processing beginning%str(
4          +)file is %scan(%sysfunc(getoption(termstmt,keyexpand)),2,%str( ));
Note: termstmt processing beginning; file is '\termstmt.sas'
5          +
6          +%put echo list global _user_ mvars, if any;
echo list global _user_ mvars, if any
7          +%put _user_;
GLOBAL MVAR1 text.1 for termstmt
GLOBAL MVAR2 text.2 for show sort order
8          **include '!site_inc\echo-mvars.sas';
9          +
10         +%put Note: termstmt processing completed.;
Note: termstmt processing completed.
NOTE: %INCLUDE (level 1) ending.
```

Summary

Suggested Reading

api: Wikipedia editors, et al. [9]
Microsoft Corp: csidl: Microsoft Corp. [5]; known-folder-id: Microsoft Corp. [6]
SAS Institute: csidl: [7]; files used by SAS: SAS Institute [8]
companions: Fehd [3] autoexec companion
Fehd [4] sysparm companion
testing: Fehd [1] writing testing-aware programs using command-line options
echoauto and verbose
sql Fehd [2] using sql for list processing;
see program ProcSQL-D-Macros-select-into-list-ordered.sas for use
in termstmt.sas

Conclusion

Configuration files contain two kinds of statements: allocations of environment variables and options' assignments. SAS configuration files use environment variables as arguments for options. Environment variables contain list of folders, libraries, with files that contain its functions, procedures, and macros. Writing configuration files can be a simple matter. The purpose of this paper is to provide a suite of tools that make testing user-written configuration files easier.

Author Information

Ronald J. Fehd Ron.Fehd.macro.maven at gmail.com
LinkedIn <https://www.linkedin.com/in/ronald-fehd-5125991/>
affiliation Stakana Analytics, Senior Maverick
also known as macro maven on SAS-L, Theoretical Programmer, developer

Trademarks

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.

References

- [1] Ronald J. Fehd. Writing testing-aware programs that self-report when testing options are true. In *NorthEast SAS Users Group Conference Proceedings*, 2007. URL <http://www.lexjansen.com/nesug/nesug07/cc/cc12.pdf>. Coders' Corner, 20 pp.; topics: options used while testing: echoauto, mprint, source2, verbose; variable testing in data step or macros; call execute; references.
 - [2] Ronald J. Fehd. How to use proc SQL select into for list processing. In *SouthEast SAS Users Group Conference Proceedings*, 2010. URL <http://analytics.ncsu.edu/sesug/2010/H0W06.Fehd.pdf>. Hands On Workshop, 40 pp.; topics: writing constant text, and macro calls, using macro %do loops; references.
 - [3] Ronald J. Fehd. An autoexec companion, allocating location names during startup. In *SouthEast SAS Users Group Conference Proceedings*, 2018. URL http://www.lexjansen.com/sesug/2018/SESUG2018_Paper-196_Final_PDF.pdf. 16 pp.; autocall macros, global symbol table, catrefs, filerefs, librefs, cexist catalogs, exist data set, sasautos.
 - [4] Ronald J. Fehd. A sysparm companion, passing values to a program from the command line. In *SouthEast SAS Users Group Conference Proceedings*, 2018. URL http://www.lexjansen.com/sesug/2018/SESUG2018_Paper-197_Final_PDF.pdf. 8 pp.; shows use of sysparm as macro variable and option which can be assigned value on command line in batch programs; program parse-sysparm parses a list of comma-separated values (csv) of form var1=value1,var2=value2,...,varN=valueN into macro variables.
 - [5] Microsoft Corp. CSIDL: constant special item id list, 2018. URL <https://docs.microsoft.com/en-us/windows/win32/shell/csidl>.
 - [6] Microsoft Corp. KnownFolderId: ?FolderId_Documents, 2021. URL <https://docs.microsoft.com/en-us/windows/desktop/shell/knownfolderid>.
 - [7] SAS Institute. CSIDL: Configuring SAS for DCOM, 2021. URL https://support.sas.com/rnd/itech/doc9/admin_oma/sasserver/comdcom/sascfg.html.
 - [8] SAS Institute. Files used by SAS :: SAS(R) 9.4 Companion for Windows, fifth edition, 2021. URL <http://support.sas.com/documentation/cdl/en/hostwin/69955/HTML/default/viewer.htm#p0bmj7wjme32ayn1h4wim7trkhp6.htm>.
 - [9] Wikipedia editors, et al. API: application programming interface, 2018. URL <https://en.wikipedia.org/wiki/API>.
-