



CHAPTER

31

The REGISTRY Procedure

Overview	849
Procedure Syntax	849
<i>PROC REGISTRY</i> Statement	849
Creating Registry Files	852
Specifying Key names	852
Specifying Values for Keys	853
Examples	854
Example 1: Example of a Registry Entry	854
Example 2: Importing a File to the Registry	854
Example 3: Listing and Exporting the Registry	855
Example 4: Comparing the Registry to an External File	856

Overview

The REGISTRY procedure maintains the SAS Registry. The registry consists of two parts. One part is stored in the SASHELP library, and the other part is stored in the SASUSER library.

Procedure Syntax

```
PROC REGISTRY <option(s)>;
```

PROC REGISTRY Statement

```
PROC REGISTRY <option(s)>;
```

To do this	Use this statement
Erase the contents of the SASUSER registry	CLEARASASUSER
Compare the contents of a registry to a file	COMPARETO
Enable registry debugging	DEBUGON
Disable registry debugging	DEBUGOFF

To do this	Use this statement
Write the contents of a registry to the specified file	EXPORT=
Provide additional information in the SAS log about the results of the IMPORT= and the UNINSTALL options	FULLSTATUS
Import the specified file to a registry	IMPORT=
Write the contents of the entire registry to the SAS log	LIST
Write the contents of the SASHELP portion of the registry to the SAS log	LISTHELP
Write the contents of the SASUSER portion of the registry to the SAS log	LISTUSER
Start exporting or writing the contents of a registry at the specified key	STARTAT=
Delete from the specified registry all the keys and values that are in the specified file	UNINSTALL
Perform the specified operation on the SASHELP portion of the SAS Registry	USESASHELP

Options

CLEARASUSER

erases the content of the SASUSER portion of the SAS Registry.

COMPARETO=*file-specification*

compares the contents of a file to a registry. It returns information about keys and values that it finds in the file that are not in the registry. It reports as differences

- keys that are defined in the external file but not in the registry
- value names for a given key that are in the external file but not in the registry
- differences in the content of like-named values in like-named keys.

COMPARETO= does not report as differences any keys and values that are in the registry but not in the file because the registry could easily be composed of pieces from many different files.

file-specification is one of the following:

'external-file'

is the name of an external file that contains the registry information.

fileref

is a fileref that has been assigned to an external file. Use the FILENAME statement to assign a fileref. (For information on the FILENAME statement, see the section on statements in *SAS Language Reference: Dictionary*.)

Interaction: By default, PROC REGISTRY compares *file-specification* to the SASUSER portion of the registry. Use USESASHELP to compare *file-specification* to the SASHELP portion of the registry.

See also: For information on how to structure a file that contains registry information, see “Creating Registry Files” on page 852.

DEBUGON

enables registry debugging.

DEBUGOFF

disables registry debugging.

EXPORT=*file-specification*

writes the contents of a registry to the specified file, where *file-specification* is one of the following:

'external-file'

is the name of an external file that contains the registry information.

fileref

is a fileref that has been assigned to an external file. Use the FILENAME statement to assign a fileref. (For information on the FILENAME statement, see the section on statements in *SAS Language Reference: Dictionary*.)

If *file-specification* already exists, PROC REGISTRY overwrites it. Otherwise, PROC REGISTRY creates the file.

Interaction: By default, EXPORT= writes the SASUSER portion of the registry to the specified file. Use USESASHELP to write the SASHELP portion of the registry.

Interaction: Use the STARTAT= option to export a single key and all of its subkeys.

FULLSTATUS

provides additional information in the SAS log about the results of the IMPORT= and the UNINSTALL options.

IMPORT=*file-specification*

specifies the file to import into the SAS Registry. PROC REGISTRY does not overwrite the existing registry. Instead, it updates the existing registry with the contents of the specified file.

file-specification is one of the following:

'external-file'

is the name of an external file that contains the registry information.

fileref

is a fileref that has been assigned to an external file. Use the FILENAME statement to assign a fileref. (For information on the FILENAME statement, see the section on statements in *SAS Language Reference: Dictionary*.)

Interaction: By default, IMPORT= imports the file to the SASUSER portion of the SAS registry. Use USESASHELP to import the file to the SASHELP portion of the registry. You must have write permission to SASHELP to use USESASHELP.

Interaction: Use FULLSTATUS to obtain additional information in the SAS log as you import a file.

See also: For information on how to structure a file that contains registry information, see “Creating Registry Files” on page 852.

LIST

writes the contents of the entire SAS Registry to the SAS log.

Interaction: Use the STARTAT= option to write a single key and all of its subkeys.

LISTHELP

writes the contents of the SASHELP portion of the registry to the SAS log.

Interaction: Use the STARTAT= option to write a single key and all of its subkeys.

LISTUSER

writes the contents of the SASUSER portion of the registry to the SAS log.

Interaction: Use the STARTAT= option to write a single key and all of its subkeys.

STARTAT='key-name'

exports or writes the contents of a single key and all of its subkeys.

Interaction: USE STARTAT= with the EXPORT=, LIST, LISTHELP, or LISTUSER option.

UNINSTALL=file-specification

deletes from the specified registry all the keys and values that are in the specified file. *file-specification* is one of the following:

'external-file'

is the name of an external file that contains the keys and values to delete.

fileref

is a fileref that has been assigned to an external file. Use the FILENAME statement to assign a fileref. (For information on the FILENAME statement, see the section on statements in *SAS Language Reference: Dictionary*.)

Interaction: By default, UNINSTALL deletes the keys and values from the SASUSER portion of the SAS registry. Use USESASHELP to delete the keys and values from the SASHELP portion of the registry. You must have write permission to SASHELP to use this option.

Interaction: Use FULLSTATUS to obtain additional information in the SAS log as you uninstall a registry.

See also: For information on how to structure a file that contains registry information, see “Creating Registry Files” on page 852.

USESASHELP

performs the specified operation on the SASHELP portion of the SAS Registry.

Interaction: Use USESASHELP with the IMPORT=, EXPORT=, COMPARETO, or UNINSTALL option. To use USESASHELP with IMPORT= or UNINSTALL, you must have write permission to SASHELP.

Creating Registry Files

You can create registry files with the SAS Registry Editor or with any text editor.

A registry file must have a particular structure. Each entry in the registry file consists of a key name, followed on the next line by one or more values. The key name identifies the key or subkey that you are defining. Any values that follow specify the names or data to associate with the key.

For more information on the SAS Registry and the Registry Editor, see the SAS online Help.

Specifying Key names

Key names are entered on a single line between square brackets ([and]). To specify a subkey, enter multiple key names between the brackets, starting with the root key. Separate the names in a sequence of key names with a backslash (\). The length of a single key name or a sequence of key names cannot exceed 255 characters (including the square brackets and the backslashes). Key names can contain any character except the backslash.

Examples of valid key name sequences follow. These sequences are typical of the SAS Registry:

```
[CORE\EXPLORER\MENUS\ENTRIES\CLASS]
[CORE\EXPLORER\NEWMEMBER\CATALOG]
[CORE\EXPLORER\NEWENTRY\CLASS]
[CORE\EXPLORER\ICONS\ENTRIES\LOG]
```

Specifying Values for Keys

Enter each value on the line following the key name that it is associated with. You can specify multiple values for each key, but each value must be on a separate line.

The general form of a value is

value-name=value-content

A *value-name* can be an at sign (@), which indicates the default value name, or it can be any text string in double quotation marks. If the text contains an ampersand (&), the character (either uppercase or lowercase) that follows the ampersand is a shortcut for the value name. See Example 1 on page 854.

A value cannot contain more than 255 characters (including quotation marks and ampersands). It can contain any character except a backslash (\).

Value-content can be any of the following:

- a string (inside double quotation marks). You can put anything inside the quotes, including nothing ("").

Note: To include a backslash character in the quoted string, use two adjacent backslashes. To include a double quotation mark, use two adjacent double quotation marks. Δ

- the string **int**: followed by a signed long integer value.
- the string **uint**: followed by an unsigned long integer value.
- the string **dword**: followed by an unsigned long hexadecimal value.
- the string **hex**: followed by any number of hexadecimal characters separated by commas. If you extend the hexadecimal characters beyond a single line, end the line with a backslash to indicate that the data continues on the next line.
- the string **double**: followed by a double value.

The following examples illustrate and explain some of the different types of values that are currently stored in the SAS Registry:

"&Open"="BUILD %8b.%32b.%32b.%8b MSG=NO"

This value is a quoted string that issues the BUILD command on the selected object.

The character **O** is a shortcut for this action.

"P&rint"="GSUBMIT 'PROC PRINT DATA=%8b.%32b;RUN;' ;LISTING;DLGPRT;"

This value is a quoted string that executes the GSUBMIT command, which in turn executes the PRINT procedure and the LISTING and DLGPRT statements.

The character **r** is a shortcut for this action.

"&Edit"="IMAGEEDIT;IMPORT'%s' 'FORMAT=GIF' "

This value is a quoted string that imports the selected GIF file into the SAS Image Editor.

The character **E** is a shortcut for this action.

"Brick"=hex:02,03,00,f4,00,01

The value of 'Brick' is hex values.

Examples

Example 1: Example of a Registry Entry

This example stores three values in the subkey named TOOLBOX:

The key name sequence identifies a subkey.

```
[ CORE\EXPLORER\MENUS\ENTRIES\TOOLBOX ]
```

The default value executes the TOOLEEDIT command on the selected TOOLBOX entry.

```
@="TOOLEEDIT %8b.%32b.%32b"
```

This value defines Open, which is the same as the default. The character **o** is a shortcut for the action.

```
"&Open"="TOOLEEDIT %8b.%32b.%32b"
```

This value defines Load, which executes the TOOLLOAD command on the selected TOOLBOX entry.

```
"Load"="TOOLLOAD %8b.%32b.%32b"
```

Example 2: Importing a File to the Registry

Procedure features: IMPORT=

Other features: FILENAME statement

This example imports a file into the SASUSER portion of the SAS Registry.

Program

The FILENAME statement assigns the fileref **source** to the external file that contains the text to read into the registry.

```
filename source 'external-file';
```

IMPORT= imports the file that is associated with the fileref **source**. By default, IMPORT= writes to the SASUSER portion of the SAS Registry.

```
proc registry
  import=source;
run;
```

SAS Log

```
1  filename source 'external-file';
2  proc registry
3     import=source;
4  run;
Parsing REG file and loading the registry please wait....
Registry IMPORT is now complete.
```

Example 3: Listing and Exporting the Registry

Procedure features:

EXPORT=
LISTUSER

This example lists the SASUSER portion of the SAS Registry and exports it to an external file.

Program

LISTUSER writes the contents of the SASUSER portion of the registry to the SAS log.

```
proc registry
  listuser
```

EXPORT= exports the registry to the specified file. By default, EXPORT= writes the SASUSER portion of the SAS Registry.

```
    export='external-file';
run;
```

SAS Log

```

1  proc registry listuser export='external-file';
2  run;
Starting to write out the registry file, please wait...
The export to file external-file is now complete.
Contents of SASUSER REGISTRY.
[ HKEY_USER_ROOT]
[   CORE]
[     EXPLORER]
[       CONFIGURATION]
[         Initialized= "True"
[     FOLDERS]
[       UNXHOST1]
[         Closed= "658"
[         Icon= "658"
[         Name= "Home Directory"
[         Open= "658"
[         Path= "~"

```

Example 4: Comparing the Registry to an External File

Procedure features: COMPARETO=

Other features: FILENAME statement

This example compares the SASUSER portion of the SAS Registry to an external file.

Program

The FILENAME statement assigns the fileref **testreg** to the external file that contains the text to compare to the registry.

```
filename testreg 'external-file';
```

COMPARETO compares the file that is associated with the fileref **testreg** to the SASUSER portion of the SAS Registry.

```
proc registry
  compareto=testreg;
run;
```

SAS Log

This SAS log shows two differences between the SASUSER portion of the registry and the specified external file. In the registry, the value of "Initialized" is "True"; in the external file, it is "False". In the registry, the value of "Icon" is "658"; in the external file it is "343".

```
1  filename testreg 'external-file';
2  proc registry
3     compareto=testreg;
4  run;
Parsing REG file and comparing the registry please wait....
COMPARE DIFF: Value "Initialized" in
[HKEY_USER_ROOT\CORE\EXPLORER\CONFIGURATION]: REGISTRY TYPE=STRING, CURRENT
VALUE="True"
COMPARE DIFF: Value "Initialized" in
[HKEY_USER_ROOT\CORE\EXPLORER\CONFIGURATION]: FILE TYPE=STRING, FILE
VALUE="False"
COMPARE DIFF: Value "Icon" in
[HKEY_USER_ROOT\CORE\EXPLORER\FOLDERS\UNXHOST1]: REGISTRY TYPE=STRING,
CURRENT VALUE="658"
COMPARE DIFF: Value "Icon" in
[HKEY_USER_ROOT\CORE\EXPLORER\FOLDERS\UNXHOST1]: FILE TYPE=STRING, FILE
VALUE="343"
Registry COMPARE is now complete.
COMPARE: There were differences between the registry and the file.
```


The correct bibliographic citation for this manual is as follows: SAS Institute Inc., SAS® *Procedures Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. 1729 pp.

SAS® Procedures Guide, Version 8

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-482-9

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

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.

IBM® and DB2® are registered trademarks or trademarks of International Business Machines Corporation. ORACLE® is a registered trademark of Oracle Corporation. ® indicates USA registration.

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

The Institute is a private company devoted to the support and further development of its software and related services.