CHAPTER 12

Functions and CALL Routines

Functions and CALL Routines Under UNIX

This chapter describes SAS functions and CALL routines whose behavior is specific to UNIX environments. Each function and CALL routine description includes a brief "UNIX specifics" section that tells which aspect of the function and CALL routine is specific to UNIX. For more information about all of these CALL routines and functions except SYSGET, see *SAS Language Reference: Dictionary*.

BYTE

Returns one character in the ASCII collating sequence

UNIX specifics: uses the ASCII collating sequence

Syntax

BYTE(*n*)

Details

UNIX operating environments use the ASCII system, so the BYTE function returns the *n*th character in the ASCII collating sequence. The values for *n* can range from 0 to 255.

See Also

- □ *SAS Language Reference: Dictionary*

CALL SYSTEM

Issues operating system commands

UNIX specifics: *Command* must evaluate to a valid UNIX command

Syntax

CALL SYSTEM(*command*);

Details

The CALL SYSTEM routine issues operating system commands. *Command* can be a UNIX command enclosed in quotes, an expression whose value is a UNIX command, or the name of a character variable whose value is a UNIX command. The output of the command appears in the window from which you invoked the SAS System.

The value of the XSYNC system option affects how the CALL SYSTEM routine works.

Note: The CALL SYSTEM routine can be executed within a DATA step. However, neither the X statement nor the %SYSEXEC macro program statement is intended for use during the execution of a DATA step. △

In the following example, for each record in **answer.week**, if the **resp** variable is **y**, the CALL SYSTEM routine will mail a message.

```
data _null_;
   set answer.week;
   if resp='y' then
      do;
      call system('mail mgr < $HOME/msg');
      end;
   run;
```

See Also

□ “Executing Operating System Commands from Your SAS Session” on page 12

COLLATE

Generates a collating sequence character string

UNIX specifics: uses ASCII collating sequence

Syntax

COLLATE(*start-pos <,end-pos>*) | (*start-pos<,,length>*)

Details

The COLLATE function returns a string of ASCII characters. The ASCII collating sequence contains 256 positions, referenced with the numbers 0 through 255. The string returned by the COLLATE function begins with the ASCII character specified by the *start-pos* argument. Characters above 127 correspond to characters used in European languages as defined in the ISO 8859 character set. If *end-pos* is specified, the string returned by COLLATE contains all the ASCII characters between *start-pos* and *end-pos*. If *length* is specified instead of *end-pos*, then the COLLATE function returns a string with a length of *length*. The returned string ends, or truncates, with the character that has the value 255 if you request a string length that would contain characters exceeding this value.

Unless you assign the return value of the COLLATE function to a variable with a defined length less than 200, the ASCII collating sequence string is padded with blanks to a length of 200. If you request more than 200 characters, the returned string is truncated to a length of 200.

DINFO

Returns the value of a directory information item

UNIX specifics: directory pathname is the only information available

Syntax

DINFO(*directory-id*, *info-item*)

Details

Under UNIX, the only *info-item* available is Directory, which is the pathname of *directory-id*. If *directory-id* points to a list of concatenated directories, then Directory is the list of concatenated directory names.

DOPTNAME

Returns the name of a directory information item

UNIX specifics: directory is the only item available

Syntax

DOPTNAME(*directory-id*, *nval*)

Details

Under UNIX, the only directory information item available is Directory, which is the pathname of the *directory-id*. The *nval*, or sequence number, of Directory is 1. If *directory-id* points to a list of concatenated directories, then Directory is the list of concatenated directory names.

DOPTNUM

Returns the number of information items available for a directory

UNIX specifics: drectory is the only item available

Syntax

DOPTNUM(*directory-id*)

Details

Under UNIX, only one information item is available for a directory. The name of the item is Directory; its value is the pathname or list of pathnames for *directory-id*, and its sequence number is 1.

FDELETE

Deletes an external file

UNIX specifics: *fileref* can be assigned with an environment variable

Syntax

FDELETE(*fileref*)

Under UNIX, *fileref* can also be an environment variable. The fileref cannot be associated with a list of concatenated filenames or directories. If the fileref is associated with a directory, the directory must be empty. You must have permission to be able to delete the file. Refer to the UNIX man page for **chmod** for more information about permissions.

FEXIST

Verifies the existence of an external file by its fileref and returns a value

UNIX specifics: *fileref* can be assigned with an environment variable

Syntax

FEXIST(*fileref*)

Under UNIX, *fileref* can also be an environment variable.

FILEEXIST

Verifies the existence of an external file by its physical name and returns a value

UNIX specifics: *filename* can be assigned with an environment variable

Syntax

FILEEXIST(*filename*)

Under UNIX, *filename* can also be an environment variable.

FILENAME

Assigns or deassigns a fileref for an external file, directory, or output device and returns a value

UNIX specifics: device types and host options

Syntax

FILENAME(*fileref, filename* <,*device*<,*host-options*<,*dir-ref*>>>)

fileref
specifies the fileref to assign to the external file. (See *SAS Language Reference: Dictionary* for details.) Under UNIX, the fileref can also be a UNIX environment variable.

filename
differs according to the device type. Table 16.1 on page 239 shows the information appropriate to each device. Remember that UNIX filenames are case-sensitive. Specifying a blank filename deassigns a *fileref* that was previously assigned.

device
specifies type of device to which the output will be sent. It can be any one of the devices listed in Table 16.1 on page 239. DISK is the default device type.

host-options
are options that are specific to UNIX. You can use any of the options that are available on the FILENAME statement. See "FILENAME" on page 237 for a description of the host options.

dir-ref
specifies the *fileref* that is assigned to the directory in which the external file resides.

FILEREF

Verifies that a fileref has been assigned for the current SAS session and returns a value

UNIX specifics: *fileref* argument can also specify a UNIX environment variable

Syntax

FILEREF(*fileref*)

Details

Under UNIX, *fileref* can also be a UNIX environment variable. See “FILENAME” on page 201 for more information.

FINFO

Returns the value of a file information item

UNIX specifics: *info-item*s available

Syntax

FINFO(*file-id*, *info-item*)

file-id

specifies the identifier that was assigned when the file was opened (generally by the FOPEN function).

info-item

specifies the name of the file information item to be retrieved. Under UNIX, *info-item* can have the following values:

- □ **`File Name`**
- □ **`Owner Name`**
- □ **`Group Name`**
- □ **`Access Permission`**
- □ **`File Size (bytes)`**

If you concatenate filenames, then an additional *info-item* is available: **`File List`**.

FOPTNAME

Returns the name of a file information item

UNIX specifics: information items available

Syntax

FOPTNAME(*file-id*, *nval*)

file-id
specifies the identifier that was assigned when the file was opened (generally by the FOPEN function).

nval
specifies the number of the file information item to be retrieved. The following table shows the values that *nval* can have in UNIX operating environments.

	Information Items Available For...	
nval	Single File	Concatenated Files
1	File Name	File Name
2	Owner Name	File List
3	Group Name	Owner Name
4	Access Permission	Group Name
5	File Size (bytes)	Access Permission
6		File Size (bytes)

FOPTNUM

Returns the number of information items available for a file

UNIX specifics: information items available

Syntax

FOPTNUM(*file-id*)

file-id
specifies the identifier that was assigned when the file was opened (generally by the FOPEN function).

Details

Under UNIX, five information items are available for all types of files:

- □ File Name
- □ Owner Name
- □ Group Name
- □ Access Permission
- □ File Size (bytes)

If you concatenate filenames, then an additional information item is available: File List.

LIBNAME

Assigns or deassigns a libref for a SAS data library and returns a value

UNIX specifics: behavior of ”libref

Syntax

LIBNAME(*libref*<,*SAS-data-library*<,*engine*<,*options*>>>)

Details

Under UNIX, if you specify a *SAS-data-library* of ”(with a space), SAS deassigns the *libref*. Also, you can use a UNIX environment variable as a *libref*.

MOPEN

Opens a file by directory ID and by member name and returns either the file identifier or a 0

UNIX specifics: open-modes

Syntax

MOPEN(*directory-id, member-name*,<*open-mode*<,*record-length*,<*record-format*>>>)

Details

The *open-mode* can be

A Append mode allows writing new records after the current end of the file.

I Input mode, which is the default.

O Output mode defaults to the OPEN mode specified in the host option in the FILENAME statement or function. If no host option is specified, it allows writing new records at the beginning of the file.

S Sequential input mode, which is used for pipes and other sequential devices such as hardware ports.

U Update mode allows both reading and writing.

W Sequential update mode is used for pipes and other sequential devices such as ports.

PATHNAME

Returns the path of a SAS data library or of an external file or returns a blank

UNIX specifics: *fileref* or *libref* argument can also specify a UNIX environment variable

Syntax

PATHNAME(*fileref* | *libref*)

Details

Under UNIX, the *fileref* or *libref* can also be a UNIX environment variable. See “FILENAME” on page 201 for more information.

RANK

Returns the position of a character in the ASCII collating sequence

UNIX specifics: uses ASCII collating sequence

Syntax

RANK(*x*)

Details

UNIX operating environments use the ASCII system, so the RANK function returns an integer representing the position of a character in the ASCII collating sequence. The argument *x* must represent a character in the ASCII collating sequence. *x* can also be a variable containing a nonprintable character. If the length of *x* is greater than 1, the function returns the rank of the first character in the string.

SYSGET

Returns the value of a UNIX environment variable

UNIX specifics: *variable-name* is a UNIX environment variable

Syntax

SYSGET(’*variable-name*’)

Details

The SYSGET function returns the value of an environment variable as a character string.

You can specify the following argument with the SYSGET function:

variable-name
is the name of an environment variable.

For example, this statement returns the value of the HOME environment variable:

```
here=sysget('HOME');
```

TRANSLATE

Replaces specific characters in a character expression

UNIX specifics: *to* and *from* arguments are required

Syntax

TRANSLATE(*source,to-1,from-1*<,...*to-n,from-n*)>

Details

Under UNIX you must specify pairs of *to* and *from* arguments, and you can use a comma as a place holder.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Companion for UNIX Environments, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS® Companion for UNIX Environments, Version 8

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

ISBN 1–58025–502–7

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 by the government is subject to restrictions as 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.

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.