

SAS/STAT 15.2[®] User's Guide Shared Concepts in High Performance Computing

This document is an individual chapter from *SAS/STAT® 15.2 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2020. *SAS/STAT® 15.2 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/STAT® 15.2 User's Guide

Copyright © 2020, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard-copy book: 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.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

November 2020

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.

SAS software may be provided with certain third-party software, including but not limited to open-source software, which is licensed under its applicable third-party software license agreement. For license information about third-party software distributed with SAS software, refer to <http://support.sas.com/thirdpartylicenses>.

Chapter 20

Shared Concepts in High-Performance Computing

Contents	
Overview	531
Single-Machine Mode	531
Single-Machine Data Access Mode	532
Output Data Sets	532
Working with Formats	532
PERFORMANCE Statement	533

Overview

This chapter describes common concepts that are used by SAS/STAT high-performance procedures. It also provides the syntax for the PERFORMANCE statement, which is common to all high-performance analytical procedures.

Single-Machine Mode

Single-machine mode is a computing model in which multiple processors or multiple cores are controlled by a single operating system and can access shared resources, such as disks and memory. In this book, single-machine mode refers to an application running multiple concurrent threads on a multicore machine in order to take advantage of parallel execution on multiple processing units. More simply, single-machine mode for high-performance analytical procedures means multithreading on the client machine.

All high-performance analytical procedures run in single-machine mode. The procedure uses the number of CPUs (cores) on the machine to determine the number of concurrent threads. High-performance analytical procedures use different methods to map core count to the number of concurrent threads, depending on the analytic task. Using one thread per core is not uncommon for the procedures that implement data-parallel algorithms.

Single-Machine Data Access Mode

When high-performance analytical procedures run in single-machine mode, they access data in the same way as traditional SAS procedures. They use Base SAS to access input and output SAS data sets.

Output Data Sets

Many procedures in SAS software add the variables from the input data set when an observationwise output data set is created. The assumption of high-performance analytical procedures is that the input data sets can be large and contain many variables. For performance reasons, the output data set contains the following:

- variables that are explicitly created by the statement
- variables that are listed in the ID statement, as described in Chapter 21, “[Shared Concepts and Topics in High-Performance Statistical Procedures](#)”
- distribution keys or hash keys that are transferred from the input data set

Including this information enables you to add to the output data set information necessary for subsequent SQL joins without copying the entire input data set to the output data set.

Working with Formats

You can use SAS formats and user-defined formats with high-performance analytical procedures as you can with other procedures in the SAS System.

High-performance analytical procedures examine the variables that are used in an analysis for association with user-defined formats. If you are running multiple high-performance analytical procedures in a SAS session and the analysis variables depend on user-defined formats, you can preprocess the formats. This step involves generating an XML stream (a file) of the formats and passing the stream to the high-performance analytical procedures.

Suppose that the following formats are defined in your SAS program:

```
proc format;  
  value YesNo      1='Yes'      0='No';  
  value checkThis  1='ThisisOne' 2='ThisisTwo';  
  value $cityChar  1='Portage'   2='Kinston';  
run;
```

The next group of SAS statements create the XML stream for the formats in the file *Myfmt.xml*, associate that file with the file reference myxml, and pass the file reference with the FMTLIBXML= option in the PROC HPLOGISTIC statement:

```

filename myxml 'Myfmt.xml';
libname myxml XML92 xmltype=sasfmt tagset=tagsets.XMLsuv;
proc format cntlout=myxml.allfmts;
run;

proc hplogistic data=six ftmlibxml=myxml;
  class wheeze cit age;
  format wheeze best4. cit $cityChar.;
  model wheeze = cit age;
run;

```

Generation and destruction of the stream can be wrapped in convenience macros:

```

%macro Make_XMLStream(name=tempxml);
  filename &name 'fmt.xml';
  libname &name XML92 xmltype=sasfmt tagset=tagsets.XMLsuv;
  proc format cntlout=&name..allfmts;
  run;
%mend;

%macro Delete_XMLStream(fref);
  %let rc=%sysfunc(fdelete(&fref));
%mend;

```

If you do not pass an XML stream to a high-performance analytical procedure that supports the FMTLIBXML= option, the procedure generates an XML stream as needed when it is invoked.

PERFORMANCE Statement

PERFORMANCE < *performance-options* > ;

The PERFORMANCE statement defines performance parameters for multithreaded computing, and requests detailed results about the performance characteristics of a high-performance analytical procedure.

You can specify the following *performance-options* in the PERFORMANCE statement:

BPC=*n*

specifies the number of bytes per character that is used in processing character strings in multibyte encodings. The default is the bytes per character of the encoding. The number of characters in a string is calculated as the byte length of the string divided by the bytes per character of the encoding. This will be incorrect when the strings in the multibyte encoding contain one or more single byte characters. In such cases, setting BPC=1 enables appropriate byte lengths to be used in processing such strings.

DETAILS

requests a table that shows a timing breakdown of the procedure steps.

NTHREADS=*n***THREADS=*n***

specifies the number of threads for analytic computations and overrides the SAS system option `THREADS` | `NOTHEADS`. If you do not specify the `NTHREADS=` option, the number of threads is determined based on the number of CPUs on the host on which the analytic computations execute. The algorithm by which a CPU count is converted to a thread count is specific to the high-performance analytical procedure. Most procedures create one thread per CPU for the analytic computations.

By default, high-performance analytical procedures run in multiple concurrent threads unless multithreading has been turned off by the `NOTHEADS` system option or you force single-threaded execution by specifying `NTHREADS=1`. The largest number that can be specified for *n* is 256. Individual high-performance analytical procedures can impose more stringent limits if called for by algorithmic considerations.

Index

- BPC= option
 - PERFORMANCE statement (high-performance analytical procedures), [533](#)
- DETAILS option
 - PERFORMANCE statement (high-performance analytical procedures), [534](#)
- high-performance analytical procedures,
 - PERFORMANCE statement, [533](#)
 - BPC= option, [533](#)
 - DETAILS option, [534](#)
 - NTHREADS= option, [534](#)
 - THREADS= option, [534](#)
- NTHREADS= option
 - PERFORMANCE statement (high-performance analytical procedures), [534](#)
- PERFORMANCE statement
 - high-performance analytical procedures, [533](#)
- THREADS= option
 - PERFORMANCE statement (high-performance analytical procedures), [534](#)