



CHAPTER

7

Analyzing the Server Log

<i>Audience</i>	87
<i>Introduction</i>	87
<i>Starting the Server Log</i>	88
<i>Using the Server Log Analysis Tools</i>	88
<i>Customizing the Programs</i>	89
<i>Executing the Driver Program</i>	89
<i>About the SLTOOL1 Program</i>	89
<i>About the SLTOOL2 Program</i>	90
<i>SLOGDATA.SERVINFO</i>	90
<i>SLOGDATA.CONNINFO</i>	90
<i>SLOGINFO.CONNSUM</i>	90
<i>SLOGDATA.TASKINFO</i>	90
<i>SLOGDATA.LIBINFO</i>	91
<i>SLOGDATA.PHYSINFO</i>	91
<i>SLOGDATA.ENGSUM1</i>	91
<i>SLOGDATA.MEMINFO</i>	91
<i>SLOGDATA.OBJINFO</i>	91
<i>SLOGDATA.IDXINFO</i>	92
<i>SLOGDATA.DIRINFO</i>	92
<i>SLOGDATA.IDXSUM</i>	92
<i>SLOGDATA.ACCTINFO</i>	92
<i>About the SLTOOL3 and SLTOOL4 Programs</i>	92

Audience

This chapter is recommended for SAS/SHARE server administrators who can write SAS programs.

Introduction

The server log records messages that result from starting and stopping a server and from many intervening client-server transactions. To make the log's raw data meaningful, you can use a set of server log analysis programs to examine specific data resources and to create usable reports. Analysis of the logged data resources of several SAS/SHARE servers allows you to compare server performance and to balance work loads among them.

Starting the Server Log

You use the PROC SERVER statement to explicitly start server logging to have the particular features that you want. To prepare for server log analysis, set the following logging feature:

Message numbering

assigns a number to each message that is recorded in the log. The server log analysis programs parse messages by using the associated numbers.

Here is the syntax for the PROC SERVER statement with two options, which you use to start logging:

```
PROC SERVER MSGNUMBER ID=DEMOSERV;
```

See Chapter 8, “The SERVER Procedure,” on page 95 for details about the PROC SERVER statement.

Using the Server Log Analysis Tools

SAS Institute provides a set of sample programs that you can use as a basis for developing your own programs to analyze server log data. These sample programs are accessible through SAS help -> Sample Code -> SAS/SHARE. Names and descriptions of the sample programs follow:

SLTOOL0.SAS

is a driver program that automates the execution of all the other programs.

SLTOOLM.SAS

is a look-up table that associates a macro variable with each message number that is generated in the server log. For example, the DINIMSG macro variable is assigned to the 43131 message number, which corresponds to a PROC SERVER start-up.

SLTOOL1.SAS

converts the server's log from a file into a SAS data file.

SLTOOL2.SAS

creates a set of SAS data files from the SAS data file that is generated by SLTOOL1.SAS and stores the files in library SLOGDATA. Each of the created files relates to specific data that is collected in the log. For example, SLOGDATA.SERVINFO is a data file that records the server's name and the times that it started and stopped.

SLTOOL3.SAS

is a sample program.

SLTOOL4.SAS

is another sample program.

Before you use these programs, you must tune them to suit your host operating environment and your log analysis needs. Running an untuned program produces unpredictable results.

Customizing the Programs

SLTOOL0.SAS is a model driver program that you can use to automate the execution of the set of programs all together. As an alternative, you may run them individually in consecutive order.

```
FILENAME INLOG '.TESTLOG';
LIBNAME SLOGDATA '.SLOGDATA';
%INCLUDE (SLTOOLM);
%INCLUDE (SLTOOL1);
%INCLUDE (SLTOOL2);
%INCLUDE (SLTOOL3);
%INCLUDE (SLTOOL4);
```

The first line associates the fileref INLOG with a host-specific name for a file that contains the server log. In this example '.TESTLOG' is the name of the file that contains the server log on a CMS host. In the second line, the LIBNAME statement associates libref SLOGDATA with the host-specific SAS data library '.SLOGDATA'. These two lines identify the external file as input and specify a SAS data library in which to write out SAS data files.

The remaining lines in the driver program are SAS INCLUDE macro statements, which read and execute each named program consecutively.

Modify the FILENAME and LIBNAME statements in SLTOOL0.SAS to specify your server's log and the repository for the SAS data files, respectively.

SLTOOLM.SAS, SLTOOL1.SAS, and SLTOOL2.SAS do not require modification. They can be run as supplied to produce SAS data files that contain information about the server's session. Those data files are the input to the analysis phase, which is performed by the programs in SLTOOL3.SAS and SLTOOL4.SAS.

You typically customize the SLTOOL3.SAS and SLTOOL4.SAS programs to produce the analyses of your server that are most relevant to your investigation.

Executing the Driver Program

How you execute the driver program depends on your operating environment. Here is an example of how to run the driver program on a UNIX system:

```
sas pathname/sltool0.sas -log /pathname/logfile
```

You invoke SAS and specify the location of SLTOOL0 and the location of the log file.

About the SLTOOL1 Program

Before you use the SLTOOL1 program, the server log must be in a file that can be read by the INFILE and the INPUT statements of the DATA step. The FILENAME statement points to the physical location that contains the server's log.

Because SLTOOL1 produces a compressed SAS data file that contains the server log and some additional data, it is recommended that you reserve an amount of space that is twice the size of the server log file.

SLTOOL1 produces a data file that is stored in the library WORK. Because the WORK library is temporary for the duration of the SAS session, it is deleted upon termination. To keep the data file for use after the session terminates, save it in a permanent library. For example,

```
DATA SLOGDATA.CVTLOG (DROP=SERVREL ANALREL
                     COMPRESS=YES
                     LABEL='Server Log');
```

About the SLTOOL2 Program

SLTOOL2 reads the SAS data file produced by SLTOOL1 and creates a group of SAS data sets. Ordinarily, SLTOOL2 is executed during the same SAS session as the SLTOOL1 program.

The libref SLOGDATA is associated with a SAS data library by the LIBNAME statement in SLTOOL0.SAS.

The data sets created by SLTOOL2 are stored in the SAS data library and are associated with the libref SLOGDATA. It is most efficient to run SLTOOL0.SAS one time to create the data files in the library SLOGDATA, and then run any number of analysis programs that access the library SLOGDATA. With large server logs, creating the data sets in the library SLOGDATA can take quite a long time and should be done only once for each server log.

SLTOOL2 creates data sets whose names include either INFO or SUM. INFO files contain observations that record specific SAS/SHARE activities, such as each time a server is started and stopped. SUM files typically present the total number of instances of an activity, such as the total number of connections made to a server.

The following sections describe each data set that is created by SLTOOL2.

SLOGDATA.SERVINFO

SERVINFO records the server name and the times at which it started and stopped. You can use this information to write a descriptive header on a report that relates to that server.

SLOGDATA.CONNINFO

CONNINFO contains connection information, one observation for each time a user connects to the server and one observation for each time a user disconnects from the server.

This data set can be used to obtain a list that shows who connected to a server, how long each user remained connected, or how many times each user connected to the server. This data set can also be used to chart the simultaneous number of connections to a server over a period of time, looking for peaks and valleys in the number of users who access data through a server.

SLOGINFO.CONNSUM

CONNSUM contains only one observation that has one variable, which stores the total number of connections to this server.

SLOGDATA.TASKINFO

TASKINFO contains one observation for each creation of a mirror resource environment and one observation for each termination of a mirror resource environment.

The name of a resource environment in this data set corresponds to the name of a SAS procedure or a window that is used to access data through the server. (SAS procedures and windows that are *not* used to access data through a server are not recorded in this data set.)

This data set can be used to obtain a list of SAS procedures and windows that were used to access data through the server, and the amount of time each procedure or each window remained active.

SLOGDATA.LIBINFO

LIBINFO contains one observation for each time a user accesses a SAS library and one observation for each time a user releases a SAS library.

This file can be used to determine how many times each library was accessed through the server and the length of time that each library was accessed. You should use the "physical name" variable because each library can be referred to by different librefs at different times.

To obtain a list of the libraries accessed through a server, use the file SLOGDATA.PHYSINFO.

SLOGDATA.PHYSINFO

PHYSINFO contains a list of the physical names that correspond to libraries that were accessed through the server.

SLOGDATA.ENGSUM1

ENGSUM1 contains a list of the engines that were used to access SAS data libraries through the server.

SLOGDATA.MEMINFO

MEMINFO contains an observation for each time a SAS library member is opened, re-opened, closed, re-named, repaired, or deleted.

This file, which tends to be very large, can be used to derive a list of members for each library that has been accessed through the server; the amount of time each member was accessed; how many times each member was accessed; whether each member was read, updated, or created; and the number of simultaneous users of each member over a period of time.

You should use the "physical name" variable because each library can be referred to by different librefs at different times.

SLOGDATA.OBJINFO

OBJINFO contains an observation for each time a SAS catalog entry is opened, closed, re-named, deleted, aliased, or has its directory information or options changed.

This file is similar to the file SLOGDATA.MEMINFO, but it contains information for catalog entries instead of members of SAS libraries.

SLOGDATA.IDXINFO

IDXINFO creates an observation for each time a user creates or deletes an index through the server.

Because creating an index tends to be expensive, this file is probably most useful as a warning signal. Furthermore, because indexes tend to be very helpful for SAS data sets that are accessed concurrently, having a list of indexes that were deleted during a server's session can also be a warning signal.

SLOGDATA.DIRINFO

DIRINFO contains an observation for each time the directory of a SAS library or the directory of a SAS catalog is opened or closed through the server.

SLOGDATA.IDXSUM

IDXSUM contains only one observation that has two variables. One variable counts how many indexes were created through the server; the other variable counts how many indexes were deleted through the server.

SLOGDATA.ACCTINFO

ACCTINFO contains one observation for each accounting message that is written to the server's log when a user disconnects. You must specify the LOG= option in the PROC SERVER statement to collect this data.

About the SLTOOL3 and SLTOOL4 Programs

You can examine the source code of these sample programs by accessing SAS help -> Sample Code -> SAS/SHARE.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/SHARE User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 247.

SAS/SHARE User's Guide, Version 8

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

ISBN 1-58025-478-0

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, September 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[®], AIX[®], DB2[®], OS/2[®], OS/390[®], RMT[™], RS/6000[®], System/370[™], and System/390[®] are registered trademarks or trademarks of International Business Machines Corporation. ORACLE[®] is a registered trademark or 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.