



CHAPTER

1

SAS ARM Interface Overview

What Is ARM? 3

Why Is ARM Needed? 3

Will ARM Affect an Application's Performance? 4

What Are the SAS ARM Interface Features? 4

Comparing the SAS 9.1 ARM Interface with the SAS 9.2 ARM Interface 4

What Is ARM?

Application Response Measurement (ARM) enables you to monitor the availability and performance of transactions within and across diverse applications. ARM enables enterprise management tools to extend directly to applications to measure application availability, performance, usage, and transaction response time.

The SAS ARM interface implements a number of features, which are compliant with the ARM 4.0 standards. SAS cooperates with open source ARM agents or vendor products that implement the ARM open API standard.

Why Is ARM Needed?

There are many techniques for measuring response times, but only ARM measures them accurately. Other techniques, although useful in other ways, might measure business service levels by assuming or guessing what a business transaction is, and when it begins and ends. Also, other techniques cannot provide the important information that ARM can, such as whether a transaction completed successfully.

Using ARM, you can log transaction records from an application to do the following:

- determine the application response times
- determine the workload and throughput of your applications
- verify that service-level objectives are being met
- determine why the application is not available
- verify who is using an application
- determine why a user is experiencing poor response time
- determine what queries are being issued by an application
- determine the subcomponents of an application's response time
- determine which servers are being used
- calculate the load time for data warehouses

Will ARM Affect an Application's Performance?

ARM is designed to be a high-speed interface that has minimal impact on applications. An ARM agent is designed to quickly extract the information that is needed and to return control to the application immediately. Processing of the information is done in a different process that can run when the application is otherwise idle.

What Are the SAS ARM Interface Features?

The SAS ARM interface uses the following features to measure and log application availability, performance, usage, and transaction response time:

- ARM agent, which is an executable program that contains an implementation of the ARM API
- ARM appender, which processes ARM transaction events and sends the events to a specified output destination
- ARM macros, which you strategically place in your SAS programs to define, start, and stop ARM data collection
- ARM system options:
 - ARMAGENT=, which specifies an executable module or keyword
 - ARMLOC=, which specifies the location of an ARM log
 - ARMSUBSYS=, which specifies whether to initialize the ARM subsystems
- default correlators, which are used to track parent and child transactions
- user metrics, which are used to measure start, update, and stop times
- performance macros, which contain default user metrics
- SAS logging facility, which enables more flexibility and control of the ARM log locations and message formats

Comparing the SAS 9.1 ARM Interface with the SAS 9.2 ARM Interface

The following table lists the differences between the SAS 9.1 ARM interface and the SAS 9.2 ARM interface.

Table 1.1

Feature	SAS 9.1 ARM Interface	SAS 9.2 ARM Interface	SAS 9.2 ARM Interface Using the SAS Logging Facility
ARM macros	%ARMEND %ARMGTID %ARMINIT %ARMSTRT %ARMSTOP %ARMUPDT	%ARMEND %ARMGTID %ARMINIT %ARMSTRT %ARMSTOP %ARMUPDT %PERFEND %PERFINIT %PERFSTOP %PERFSTRT	%ARMEND %ARMGTID %ARMINIT %ARMSTRT %ARMSTOP %ARMUPDT %PERFEND %PERFINIT %PERFSTOP %PERFSTRT
ARMAGENT= <i>ARM agent</i> or SAS (default)	ARMAGENT= <i>ARM agent</i> or SAS (default)	ARMAGENT= <i>ARM agent</i> or SAS (default)	ARMAGENT=LOG4SAS
ARMLOC=	yes	yes	no
ARMSUBSYS=	yes	yes	yes
Correlators	yes, user-defined correlators	yes, user-defined correlators	by default
Post-processing macros	%ARMJOIN %ARMPROC	%ARMJOIN %ARMPROC	none
User metrics	limited user-defined metrics	user-defined metrics; additional user metrics are memory, thread count, and Read and Write statistics, which are default metrics when using %PERFSTRT and %PERFSTOP	all user metrics, by default, when using %PERFSTRT and %PERFSTOP macros

