



THE
POWER
TO KNOW.



Troubleshooting

SAS[®] Risk Dimensions[®]

Table of Contents

Scope of This Document	1
Getting Started	1
Locate the SAS® Log File	1
Check the Error and Warning Messages	2
Search the SAS Knowledge Base	2
Applying Software Updates	2
Determine Your SAS® Release Information	3
Check for Maintenance Releases.....	3
Check for Hot Fixes	3
Consider Upgrading to the Current Release of SAS® Risk Dimensions®.....	3
Enhancing the SAS® Log	4
Macro and Source Code Details.....	4
PUT Statement	4
TRACE Statement	5
Viewing the Contents of Your Risk Environment	5
Graphical User Interface.....	6
PRINT Statement.....	6
WRITE Statements	6
RISKENV Procedure	6
Narrowing the Scope of the Problem	7
Subset the Portfolio Data.....	7
Perform a Single Risk Analysis	7
Eliminate Unnecessary PROC Statements	7
Handling Generic Error Messages	7
Check the Syntax.....	9
Search for the Error Online.....	9
Identify Where the Error Occurs	9
Deal with the First Error First.....	10
Investigate Changes That Were Made to the Environment.....	10
Increase the Detail of the SAS® Log	10
Construct a Reproducible Test Case.....	10
Isolate the Cause of the Error.....	10
Check the Quality of the Data.....	11
Validate the COMPILE Code.....	11
Integrating with Third-Party Function Libraries	11
Isolate the Problem.....	11

Check the Syntax.....	12
Function Calls	12
Mismatched Variable Types.....	12
Memory Allocation.....	12
Review Operating System Considerations	12
Operating System Version and Architecture.....	12
Compiler Version and Options	12
Check the Configuration of the Third-Party Library	12
Search Paths and Dependent Libraries	13
Licensing and Environment Variables.....	13
Third-Party Library Support.....	13
Managing Performance	13
Use the FULLSTIMER Option	13
Distribute Processing of the Risk Analysis Project.....	13
Code Method Programs Efficiently	14
Exclude Creation of Unnecessary Output Data.....	14
Control Simulation Options	15
Perform Principal Component Analysis	15
Use the MEMSIZE Option	15
Analyze the Hardware and Operating System	15
Contacting SAS for Assistance	16
Prior to Contacting SAS.....	16
Check the Documentation.....	16
Search SAS Notes	16
Troubleshoot	16
Whom to Contact.....	16
How to Contact SAS Technical Support.....	16
What to Send	17
What to Expect	17
Urgent Issues.....	18
Monitor the Status of a Reported Problem	18

Scope of This Document

SAS® Risk Dimensions® is a multi-tiered software solution that is supported across Windows, UNIX, and Linux platforms. It comprises other SAS software products, such as Base SAS® and SAS® Integration Technologies. Likewise, other SAS software solutions, such as SAS® Risk Management for Banking, incorporate SAS Risk Dimensions.

The focus of this document is the server component set that is specific to SAS Risk Dimensions. This includes SAS procedures such as the RISK and COMPILE procedures.

This document assumes knowledge and understanding of the SAS Risk Dimensions product documentation. For training needs, contact SAS Training.

Getting Started

The following are starting points for troubleshooting problems that occur in SAS Risk Dimensions:

- Locating the SAS log file
- Checking the error and warning messages
- Searching the SAS Knowledge Base

Locate the SAS® Log File

When you encounter a problem while running a SAS Risk Dimensions program, the SAS log is typically the best place to start troubleshooting the problem. A SAS log is generated each time that a SAS program is run.

The way that you will locate the SAS log depends on how you executed the SAS Risk Dimensions program. There are multiple ways to execute a SAS Risk Dimensions program, including the following:

- Through a SAS interactive session
- In batch mode
- Through a shell script
- Through the SAS Risk Dimensions graphical user interface (GUI)
- In SAS® Enterprise Guide®
- On a command line
- Through a cron job

- Through SAS® Stored Processes
- Through Web applications

The SAS program can be executed remotely or locally. The location of the SAS log depends on the logging options that you specify when you run the SAS program.

Typically, the SAS log is located in either the directory where the SAS program is stored or in the directory where the SAS executable is stored. You can easily change the location of the SAS log using the PRINTTO procedure or by specifying options such as the LOG system option.

For more information about storing SAS logs, see Routing and Customizing SAS Output.

Check the Error and Warning Messages

When you run a SAS Risk Dimensions program, or any other SAS program, it is a best practice to check the resulting SAS log for any error or warning messages.

The order of the messages within the log is important. In general, the errors and warnings that appear first in the SAS log should be addressed first. Recognize, too, that problems which cause errors or warnings could have a cascading effect and cause additional problems as the program continues to run.

Read the message. Check the corresponding SAS code for problems. When descriptive error and warning messages are available, use those messages to diagnose the problem.

For more information about interpreting SAS logs, see The SAS Log and Types of Errors in SAS.

Search the SAS Knowledge Base

Solutions to common problems are documented in the SAS Knowledge Base. Search for related SAS Notes by using a description of the problem that you encounter or the actual text of the error message as the search-field keywords. If you find a SAS Note that is related to the problem, ensure that the version and operating system information in the SAS Note is applicable to your SAS Risk Dimensions environment.

Applying Software Updates

Your SAS software installation is hierarchically organized, as follows:

SAS version (example: SAS 9.2)

→ SAS maintenance level (example: SAS 9.2 M2)

→ SAS product version (example: SAS Risk Dimensions 5.2)

→ SAS product maintenance level (example: SAS Risk Dimensions 5.2_M1)

→ SAS product hot fix (example: B42001)

That is, hot fixes are issued according to the maintenance level of a SAS product. SAS products are built on top of an underlying SAS version and its associated maintenance level.

Keep your SAS Risk Dimensions software up-to-date by taking advantage of any available software bug fixes and feature enhancements. Determine the version information for your SAS software products, check for any software updates that might be available, and apply the updates accordingly.

Determine Your SAS® Release Information

To determine version information for your SAS installation, follow the steps in these SAS Notes:

For SAS 9.2 and later

SAS Note 35968: Using the ViewRegistry Report and other methods to determine the SAS® 9.2 and SAS® 9.3 software releases and hot fixes that are installed

For SAS 9.1 and earlier

SAS Note 20390: The SAS Installation Reporter program creates a report showing which applications, clients, and hot fixes are installed

Check for Maintenance Releases

To check for available maintenance releases, see [Hot Fixes and Downloads](#). You can also subscribe to the [TSNEWS-L listserv](#) to receive e-mail alerts regarding new maintenance releases.

Check for Hot Fixes

To check for available hot fixes, see [Technical Support Hot Fixes](#). The hot-fix downloads are organized by SAS version, SAS product, and product version. Note: Because SAS Risk Dimensions comprises and uses other SAS products, hot fixes for several SAS products might be applicable to your SAS Risk Dimensions installation. Examples of other relevant products include Base SAS, SAS/OR® software, SAS/STAT® software, SAS/IML® software, and SAS/ETS® software.

Consider Upgrading to the Current Release of SAS® Risk Dimensions®

If you do not have the latest version of SAS Risk Dimensions installed, consider upgrading in order to take advantage of the latest features and fixes. For information about upgrading your SAS Risk Dimensions installation, contact the SAS account representative for your organization.

Enhancing the SAS® Log

It is often useful to increase the amount of logging detail that is produced. This can help you to more specifically identify where in the SAS Risk Dimensions program a problem might be originating. The following are ways to produce more debugging information from your SAS Risk Dimensions program.

Macro and Source Code Details

If your SAS Risk Dimensions program contains macro processing or use of the %INCLUDE statement, then submit the following SAS code before running the program. This code produces the macro processing and SAS source code details of the program:

```
options mprint mlogic symbolgen source2;
```

The **MPRINT**, **MLOGIC**, and **SYMBOLGEN** SAS system options cause the macro processing details and macro variable values to be printed to the SAS log. The **SOURCE2** SAS system option causes the SAS code to be printed to the SAS log for any SAS programs that are included in the SAS Risk Dimensions program via the **%INCLUDE statement**.

To also add macro nesting details to the SAS log, specify the **MPRINTNEST** and **MLOGICNEST** SAS system options.

PUT Statement

It is a common usage problem to supply invalid variable values to a SAS Risk Dimensions program. Invalid variable values could be specified in the instrument position data, the market data, and in the PROC COMPILE code where you construct pricing methods and call third-party function libraries.

The PUT statement is a useful way to check for invalid variable values within your SAS Risk Dimensions program. Specify the PUT statement within the COMPILE procedure to print variable values to the SAS output file. For example, to print the value of the variable “var1,” specify the following within a method in the COMPILE code:

```
proc compile ...  
  
    put var1=;  
  
    ...
```

The variable values are written to the SAS output file, instead of to the SAS log. Check the values that are printed and ensure that they are valid. As needed, correct your input data and modify your programming code to handle the possibility of invalid variable values.

For more information about using the PUT statement, see [PUT Statement](#). See also the *PUT Statement* section in [PROC FCMP and DATA Step Differences](#) for PUT statement information that is specific to the FCMP (and COMPILE) procedures.

TRACE Statement

In comparison to the PUT statement, which enables you to see the values for variables during the processing of a SAS Risk Dimensions program, the TRACE statement provides a more comprehensive view of the SAS Risk Dimensions program processing. It enables you to see run-time tracing information such as the detailed method-program calculations, the flow of method programs, and dependency lists for the risk factor variables.

The TRACE statement prints to the SAS output file, instead of to the SAS log. Note that the output from the TRACE statement can be extensive, depending on the size of your SAS Risk Dimensions environment and which TRACE statement options you specify.

Specify the TRACE statement within a PROC RISK step, after the PROJECT statement and prior to the RUNPROJECT statement, for a risk analysis project that you want to debug:

```
proc risk <options>;
  <RISK statements>
  project <options>;
  ...
  trace <options>;
  ...
  runproject <options>;
  ...
run;
```

For more information about the TRACE statement and its syntax, see the TRACE Statement section of The RISK Procedure chapter in the [SAS Risk Dimensions Procedures Guide](#) for your version of SAS Risk Dimensions.

Viewing the Contents of Your Risk Environment

When you run a SAS Risk Dimensions program, you work with risk environments that are created and stored via the ENVIRONMENT statement in the RISK procedure. Depending on which version of SAS Risk Dimensions you are using, the risk environments are stored as SAS data sets, as SAS catalog files, or as a combination of both.

When you encounter a problem in SAS Risk Dimensions, it is often helpful to view the contents of the risk environments that you are using. You can view those contents via any of the following ways:

- In the GUI
- Through the PRINT statement
- Through WRITE statements
- Through the RISKENV procedure

Graphical User Interface

SAS Risk Dimensions 4.2.1 and later releases have a Java-based client GUI, and releases 4.2.1 and earlier of SAS Risk Dimensions have an SAS/AF® software-based server GUI. In either case, you can use the GUI to open the risk environment and view its contents. You can use the GUI to view the environment configuration, risk-factor model specifications, market and portfolio data registration, and analysis project settings.

The SAS Risk Dimensions GUI documentation is available in the Help menu for each respective GUI.

PRINT Statement

The PRINT statement provides a useful alternative to the GUI for viewing the contents of a risk environment. It prints the object definition information for a risk environment, including search paths, inheritance structures, summary information, option specifications, registered variables, risk factor models, and method programs. It lists the information in the SAS output file instead of in the SAS log.

For more information about the PRINT statement, see the *PRINT Statement* section of *The RISK Procedure* chapter in the [SAS Risk Dimensions Procedures Guide](#) for your version of SAS Risk Dimensions.

WRITE Statements

In addition to the PRINT statement, there are several WRITE statements that you can use to print out information about a risk environment. This can be a useful way to readily see complete lists of objects that are registered in a risk environment. The WRITE statements include the WRITEARRAYS, WRITECATEGORIES, WRITEREFS, WRITETYPES, and WRITEVARS statements in the RISK procedure.

Unlike the PRINT statement, which prints risk environment information to an output listing, the WRITE statements print the environment information into SAS data sets. Incidentally, these output data sets are structured such that they can be directly registered into a risk environment by using the corresponding READ statements.

For more information about the WRITE statements, see the corresponding WRITE statement sections of *The RISK Procedure* chapter in the [SAS Risk Dimensions Procedures Guide](#) for your version of SAS Risk Dimensions.

RISKENV Procedure

For SAS Risk Dimensions 4.2.1 and earlier, you can use the RISKENV procedure to convert a risk environment file into batch code that in turn can be used to regenerate the risk environment file. This is especially useful if you have created or modified an environment in the SAS Risk Dimensions GUI and want to work with the environment outside of the GUI, using SAS code.

Narrowing the Scope of the Problem

For a problem that occurs in a large SAS Risk Dimensions program, it is often possible to narrow the scope of the problem to a part of the program and to isolate the problem. The following are ways to simplify a SAS Risk Dimensions program in order to isolate a problem:

- Subsetting the portfolio data
- Performing a simple risk analysis
- Eliminating unnecessary PROC statements.

Subset the Portfolio Data

Check whether it is possible to reproduce the problem with a single instrument position in the instrument data. To do this, either subset the instrument data before registering it in the RISK procedure or use the FILTER statement in the RISK procedure.

Perform a Single Risk Analysis

Check whether it is possible to reproduce the problem with a single risk analysis, or even a simple mark-to-market analysis. If you are performing a simulation analysis, check whether the problem occurs when you generate only the market states with the ENDPROJECT= option, instead of performing the full risk analysis.

Eliminate Unnecessary PROC Statements

Check whether any RISK procedure configuration steps are necessary to reproduce the problem. For example, you can check whether cross classification, filters, batch reports, or post-VaR analyses are related to the problem by commenting out these statements and options and checking whether the problem is still reproducible.

Check whether the RISK procedure is needed to reproduce the problem. Modify your PROC COMPILE code to use hardcoded inputs and check whether the problem still occurs when only the modified PROC COMPILE step is executed.

Handling Generic Error Messages

When you run a SAS Risk Dimensions program, you might encounter generic error messages that do not identify the cause of the problem or provide direction to solve it. Here are some examples:

- Segmentation violations
 - Segmentation Violation In Task [RISK]

Fault and traceback information not available

Task Traceback

- Segmentation Violation In Task [RISK]
 /d1/SAS_9.2/SASFoundation/9.2/sasexe/sas:do_traceback+0xd1
 /d1/SAS_9.2/SASFoundation/9.2/sasexe/sas:tktracex+0x1c
 /d1/SAS_9.2/SASFoundation/9.2/sasexe/sas:ht_handle_program_fault+0x217
 ...
- Illegal Instruction In Task [RISK]
 Fault Occurred at [0x7000000082b8ff0(?)]
 Task Traceback
 cmpfrun+0x38 -- bridge stub
 rskrun+0x38 -- bridge stub
 _uc_trampoline+0xffffffffffffc
- Segmentation Violation In Task [RISK (3)]
 /SAS/SASFoundation/9.2/sasexe/sas [0x4f8890]
 /SAS/SASFoundation/9.2/sasexe/sas [0x4f8a34]
 /SAS/SASFoundation/9.2/sasexe/sas [0x41adbc]
 ...
- Access violations
 - ERROR: Write Access Violation In Task [COMPILE)
 Exception occurred at (5DCB17A2)
 Task Traceback
 Address Frame (DBGHELP API Version 4.0 rev 5)
 5DCB17A2 0350F8C0 tkebio:tkebio+0x7A2
 65C19B8E 0350F91C sasempa:mcn_main+0x48B8E
 65C32679 0350F990 sasempa:mcn_main+0x61679
 ...
 - ERROR: Read Access Violation In Task [RISK)
 Exception occurred at (0BC1FB14)
 Task Traceback
 Address Frame (DBGHELP API Version 4.0 rev 5)
 65C3140D 0355F028 sasempa:mcn_main+0x6040D
 05452A5F 0355F05C sasriska:mcn_main+0xD1A5F
 053CAAC4 0355F110 sasriska:mcn_main+0x49AC4
 ...
 - ERROR: Write Access Violation In Task [RISK]
 Exception occurred at (086F398C)
 Task Traceback
 Address Frame (DBGHELP API Version 4.0 rev 5)
 086F398C 08A6F8CC sasriska:mcn_main+0xB298C
 086B671F 08A6FAD4 sasriska:mcn_main+0x7571F
 086B553E 08A6FAF4 sasriska:mcn_main+0x7453E
 ...

- Bus Error In Task [RISK]
Fault Occurred at
[/SASRD_D/SAS9system/SAS_9.1/sasexe/sasriska:rskmem3+0x590]
Task Traceback
/SASRD_D/SAS9system/SAS_9.1/sasexe/sasriska:rskmem3+0x590
/SASRD_D/SAS9system/SAS_9.1/sasexe/sasriska:rskrun+0x13e8
/SASRD_D/SAS9system/SAS_9.1/sasexe/sasrisk:rspstmt+0x6d28
...
• ERROR: Signal caught by CMP from PROC

The following are steps that you can take to diagnose the problem when you encounter generic error messages

- Checking the syntax
- Searching for an error online
- Identifying where the error occurs
- Dealing with the first error first
- Investigating changes in the environment
- Increasing the amount of detail in the SAS log
- Constructing a reproducible test case
- Isolating the cause of the error
- Checking the quality of the data
- Validating the COMPILE code

These steps are described below.

Check the Syntax

Ensure that your SAS code is syntactically correct according to the [SAS Risk Dimensions documentation](#).

Search for the Error Online

Search the text of the error message in the [SAS Knowledge Base](#) for any SAS Notes or other support documents that address the problem.

Identify Where the Error Occurs

Determine which part of the SAS Risk Dimensions program generates the error message. Run the SAS code in sections and check the SAS log for the error messages after you run each section.

Also, in the SAS log, notice the set of messages that occur immediately before the error message. Use the location of the error message in the SAS log to determine what part of the SAS Risk Dimensions program fails. Trace back to the SAS code and data that result in the error message.

Deal with the First Error First

If the SAS log contains more than one error message, your first step is to deal with the error message that occurs first. The subsequent error messages might be an effect of the problem that caused the initial error message.

Also pay attention to the warning messages in the SAS log, especially if they occur near and prior to the error message in the SAS log. Modify your SAS Risk Dimensions program so that the warning messages do not occur. Then rerun the program and check whether the error message still occurs.

Investigate Changes That Were Made to the Environment

If your SAS Risk Dimensions program ran successfully in one run but then failed in another, identify any changes that were made between the successful run and the failed run. For example, investigate the differences in the input data, in the SAS code, or in any other parts of the environment that were modified. Check whether those modifications result in the error messages.

Increase the Detail of the SAS® Log

Specify the MPRINT, MLOGIC, SYMBOLGEN, and SOURCE2 SAS system options, and rerun the program. Specify the PUT and TRACE statements to see more details in the SAS log.

See the [Enhancing the SAS Log](#) section above for more information.

Construct a Reproducible Test Case

Check whether the error can be reproduced with a simplified SAS program that does not require the entire SAS Risk Dimensions program to run. Set up the test case on a separate computer to check whether the problem is specific to a particular operating environment.

Isolate the Cause of the Error

Isolate the problem by paring down the risk environment so that the error is reproduced with the minimum data, analyses, and options. See the [Narrowing the Scope of the Problem](#) section above for more details.

Check the Quality of the Data

Check how invalid data is handled in your method programs. Insert error-handling code before registering data sources or within the method programs to prevent mishandling of invalid data in the method programs. Use the PUT and TRACE statements to verify the variable values that are passed to the method programs. (See the [Enhancing the SAS Log](#) section above for more information about specifying the PUT and TRACE statements.)

Validate the COMPILE Code

Work through the method programs and the function and subroutine definitions, and check for problematic code, such as infinite loops or misallocated memory. Comment out the method programs, specify constant output values (example: `_VALUE_=0`), rerun the SAS Risk Dimensions program, and check whether the errors still occur.

Integrating with Third-Party Function Libraries

You can integrate a SAS Risk Dimensions program with third-party function libraries by using the COMPILE and PROTO procedures. The following are diagnostics that you can apply if you encounter a problem with SAS and third-party library integration.

- Isolating the problem
- Checking the syntax
- Reviewing operating system considerations
- Checking the configuration of the third-party library

Isolate the Problem

A key to resolving problems that involve third-party library integration is to isolate the problem. The problem could be independent of SAS and specific to the third-party library, it could be vice versa, or it could be a problem in the interaction between the two. Even if SAS reports an error in the SAS log, SAS is not necessarily the origin of the error message.

To isolate the problem, set up a separate test program for the third-party library that is independent of SAS. In the test program, reference the third-party library by using the same input values that you specify in SAS. If the problem still occurs, then you need to resolve the problem outside of SAS before you use the library with SAS.

In SAS, set up a test program that does not use the RISK procedure, but instead uses only the COMPILE and PROTO procedures along with the third-party library. Different ways to accomplish this include hardcoding input values in PROC COMPILE, using the DATA= option in the PROC COMPILE statement, and using the RUNMETHOD statement. Check whether the problem still occurs when the RISK procedure is not involved.

Check the Syntax

Check for any incorrect syntax that could cause a problem, such as in the following locations.

Function Calls

Ensure that the syntax of the third-party function call is correct. Function names, argument lists, and variable types need to be specified correctly.

Mismatched Variable Types

There are several places to check for variable type mismatches. The parameter types in the prototypes that are specified in the PROC PROTO step and the variable types that are set in the PROC COMPILE step need to be consistent with the parameter types that are specified in the third-party function header file. C types that are supported in SAS are noted in the [PROTO procedure documentation](#).

Memory Allocation

Prevent memory leaks by ensuring that memory allocations for the third-party libraries are freed as needed.

Review Operating System Considerations

Note any operating-system-specific requirements.

Operating System Version and Architecture

In most cases, a third-party library works only on the operating system and architecture (32-bit versus 64-bit) on which it was built. In some cases, the third-party library might work only with the same operating system version.

Compiler Version and Options

The compiler, compiler options, and compiler version that were used to build the third-party library are also crucial. Check which compiler and compiler version were used to build the third-party library. If you are encountering a problem with the third-party library on your machine, test whether the library works on a machine that has the same compiler, compiler version, and operating system version installed on it as that used to build the library. On Windows, ensure that your STDCALL option specification in the [PROC PROTO statement](#) is consistent with whether the `__stdcall` convention was used to build the library.

Check the Configuration of the Third-Party Library

Ensure that the third-party library is configured correctly according to the requirements for that library.

Search Paths and Dependent Libraries

Ensure that the search paths are set as needed (example: LD_LIBRARY_PATH on UNIX). Ensure that any dependent libraries are installed and that the library knows where to find them. Make sure that the permissions for each of the necessary libraries are set as needed, such as by checking for Read and Execute permissions on the libraries for the user that would be accessing those libraries.

Licensing and Environment Variables

Ensure that the licensing is active and correctly configured for the third-party library. Check for any special environment variables that need to be set, for licensing or otherwise.

Third-Party Library Support

If you need help with a problem that is specific to a third-party library and that does not involve SAS, contact the vendor that supports that library.

Managing Performance

SAS Risk Dimensions is designed to perform heavy computations with large amounts of data. There are many factors that play a role in the performance of running a risk analysis project in SAS Risk Dimensions. Many of those factors, such as hardware, significantly impact performance but are set up outside of the SAS software system. This section highlights common ways to improve performance within SAS Risk Dimensions. Performance factors to consider that are outside of SAS Risk Dimensions are also noted.

Use the FULLSTIMER Option

Specify the **FULLSTIMER** SAS system option in order to display detailed run-time information in the SAS log:

```
options fullstimer;
```

To capture the detailed timing information for your entire SAS Risk Dimensions program, specify the FULLSTIMER option at the beginning of your SAS Risk Dimensions program. Scan the resulting SAS log for unexpectedly high run times.

Distribute Processing of the Risk Analysis Project

SAS Risk Dimensions contains macros that you can use to distribute the execution of a risk-analysis project across multiple processors. The project can be split by market states or by portfolio positions and then run in parallel by using SAS/CONNECT® or by using SAS® Grid Computing. The spawning macros leverage the tools in the RUNPROJECT statement that enable you to break up the risk analysis project.

For more information about the spawning macros in SAS Risk Dimensions, see the following [documentation](#), depending on the version of SAS Risk Dimensions that you have installed:

Releases 5.2 and 5.3

%RDCSPAWN Macro and *%RDGSPAWN Macro* sections in the Procedures Guide

Release 4.2.1

RDSPAWN Macro for Parallel Processing section in SAS® Risk Dimensions® 4.2 Batch Interface

See also the following sections of documentation that explain the RUNPROJECT breakup tools:

Releases 5.2 and 5.3

RUNPROJECT Statement and *Distribution of Project Processing* sections in the Procedures Guide

Release 4.2.1

RUNPROJECT Statement and *RUNPROJECT Statement Task Distribution Details* sections in SAS® Risk Dimensions® 4.2 Batch Interface

The default location for the spawning macro files is the **ISASROOT/sasautos** directory in UNIX environments and the **ISASROOT/risk/sasmacros** directory in Windows environments.

Code Method Programs Efficiently

Method programs are a central component of a SAS Risk Dimensions program. They are typically called many times during the execution of a risk analysis project. A small improvement in the method program code can significantly improve the performance of the overall program.

For example, if a portfolio file contains 1,000 different positions in one type of instrument, then the pricing method that is associated to that instrument is called 1,000 times during a mark-to-market valuation of that portfolio. If a simulation analysis is added to this project that has 1,000 simulation replications, then the pricing method is called 1,000,000 times. When multiple horizons are specified, that number of calls is made for each horizon.

Because method programs are heavily used by a risk analysis project, efficient coding is important for achieving strong performance.

For more information about writing method programs, see *The COMPILE Procedure* chapter of the [SAS Risk Dimensions Procedures Guide](#) (Batch Interface for release 4.2.1). For method programs that include calls to third-party functions or C helper functions, see [PROTO Procedure](#).

Exclude Creation of Unnecessary Output Data

If you need only particular output data sets from your SAS Risk Dimensions program, then you can limit the output data set creation to only those tables that you need.

A list and description of the SAS Risk Dimensions output data sets are available in the *SAS Risk Dimensions Output* appendix of the User's Guide in the [documentation](#) (for release 4.2.1, see the *Output Data Sets* section of the Analysis and Modeling book).

For more information about specification of output data set options, see the *SETOPTIONS* statement section in the Procedures Guide of the [SAS Risk Dimensions documentation](#).

Control Simulation Options

While a large number of simulation replications is desirable for increasing accuracy of the results, the tradeoff is that more time is required to run the analysis project. Use the *NDRAW=* option in the *SIMULATION* statement of the *RISK* procedure to control the number of replications that are performed during a simulation analysis.

If your SAS Risk Dimensions program unnecessarily contains extra horizons, then you can remove the extra horizon specifications from the *HORIZON=* option in the *SIMULATION* statement of the *RISK* procedure.

Perform Principal Component Analysis

The number of risk factors in a risk analysis project can have a significant impact on the run time. PCA (principal component analysis) is one way to reduce memory usage and improve performance without greatly sacrificing accuracy.

See the *Principal Component Analysis* section of the Procedures Guide in the SAS [Risk Dimensions documentation](#) for more information about PCA. In SAS Risk Dimensions 5.2 and later, you can perform PCA by specifying the *PCA* option in the *SETOPTIONS* statement.

See also [The PRINCOMP Procedure](#), which you can use to analyze principal components. You can use the *PRINCOMP* procedure in conjunction with versions 4.2.1 and earlier to perform PCA in a SAS Risk Dimensions program.

Use the MEMSIZE Option

The *MEMSIZE* SAS system option on UNIX systems specifies the total amount of memory available to each SAS session. Setting *MEMSIZE* to 0 or *MAX* is discouraged except for testing purposes. (See [SAS Note 14125](#).) You can use the *OPTIONS* procedure to determine how *MEMSIZE* is specified on your system. Submit the following SAS code and check the *MEMSIZE* value in the SAS log:

```
proc options; run;
```

If a risk analysis project is failing due to a lack of available memory, try increasing the *MEMSIZE* value. For details about how to specify the *MEMSIZE* SAS system option, see the *MEMSIZE* System Option documentation (for [UNIX](#) or [Windows systems](#)).

Analyze the Hardware and Operating System

Hardware enhancements such as a more powerful CPU, more memory, and fast-saving storage space can improve system performance. Monitor the system to determine where processing time is occurring. The paper, [Solving SAS Performance Problems: Employing Host-Based Tools](#), provides good recommendations for analyzing performance on a given system.

Contacting SAS for Assistance

Prior to Contacting SAS

Check the Documentation

Ensure that your risk environment configuration and analysis is consistent with the [SAS Risk Dimensions documentation](#).

Search SAS Notes

Search for [SAS Notes](#) that address the problem.

Troubleshoot

Work through the troubleshooting tips that are provided above in this document.

Whom to Contact

Depending on the type of problem that you are encountering, different SAS contacts are available to assist you. The following table offers guidance:

<i>Type of Problem</i>	<i>Contact</i>
Licensing	The SAS site representative for your organization. If you do not have one, then contact the SAS account representative for your organization. Otherwise, see the page How to Contact Contracts Support .
SAS software not functioning as documented	SAS Technical Support
Customized design needed for your SAS software implementation	SAS Consulting
Additional SAS software needed	SAS Sales

How to Contact SAS Technical Support

There are three ways to contact SAS Technical Support for assistance with a problem:

1. Submit the problem via the [online submission form](#).

This method provides several advantages. It already has an input field for the information that SAS Technical Support needs in order to assist you. It pre-fills fields for you if you have already used this form for previous requests.

2. Send an e-mail to support@sas.com.

Include the **Site Number** for your SAS license, your name and contact information, the operating system, the SAS product and version that you are using, a description of the problem that includes any relevant error messages, and any SAS log, SAS code, data, or other files that are relevant to the problem.

3. Call (919) 677-8008.

Be prepared to provide the **Site Number** for your SAS license, the operating system, and the SAS product and version that you are using.

What to Send

In addition to a complete description of the problem and the information related to your SAS Risk Dimensions installation, the following might also be useful:

- Detailed SAS log. See the Enhancing the SAS Log section above for more information
- The objective that you are attempting to accomplish when you encounter this problem
- A clear set of instructions to reproduce the problem
- Any relevant screen displays
- A simplified stand-alone test program that demonstrates the problem
- A copy of the SAS Risk Dimensions environment to be diagnosed at SAS; include all code, data, and instructions that are necessary to replicate the environment and reproduce the problem

If you send a .zip file to SAS, prefix the filename with zqj_ (example: zqj_myfile.zip). You can e-mail files that are 4MB or less in size. Use FTP to submit larger files.

What to Expect

For issues that you report to SAS Technical Support, you are issued a tracking number. Issues are handled in accordance with the stated SAS Technical Support Services and Policies. The level of support that is available for each version of SAS Risk Dimensions is noted in Usage Note 39230.

Urgent Issues

When you contact SAS with a critical problem that requires an urgent resolution, in addition to the details of the problem, provide the information about the problem that makes it an urgent request. For example, if a critical production system is down and causing a significant operational impact to your business, or if multiple users are affected and unable to perform their job function, then note this by describing the business impact and the number of users affected who cannot perform their work.

See the [SAS Technical Support Services and Policies](#) Web page for more information about how SAS Technical Support handles urgent issues. See especially the *Problem Response Time* and *Escalating Problems* sections.

Monitor the Status of a Reported Problem

Once you have a tracking number for an issue that you have reported, you can log on to [Technical Support Tracks](#) to monitor its status. The Tracks system contains the written communications between you and SAS Technical Support.

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. Copyright © 2011 SAS Institute Inc., Cary, NC, USA. All rights reserved.