



CHAPTER

2

Using Compute Services

<i>Introduction</i>	15
<i>Benefits of Using Compute Services with RSUBMIT</i>	15
<i>Considerations for Using Compute Services with RSUBMIT</i>	16
<i>How to Use Compute Services</i>	17
<i>Synchronous Remote Submits</i>	17
<i>Asynchronous Remote Submits</i>	17
<i>MP CONNECT</i>	18
<i>Output Delivery System</i>	18

Introduction

Compute services give you easy access to many of the remote resources on your network from a single local SAS session. SAS/CONNECT provides access to remote resources in two forms, synchronous and asynchronous. With synchronous processing, you wait for the results of the remote processing before you are able to continue local processing. With asynchronous processing, you immediately regain control to continue local processing while the remote job executes, and results can be obtained at a later time.

This type of access results in maximum utilization of all of the computing resources within your organization. Compute services are provided by the RSUBMIT command and the Remote SQL Pass-Through (RSPT).

Compute services also allows you to take advantage of the Output Delivery System (ODS) in SAS software to change the format and appearance of SAS output that is generated on a remote host.

Note: Before using compute services, you must establish a connection between a local and a remote host by using the SIGNON statement. See Chapter 23, “Starting and Stopping SAS/CONNECT Software,” on page 193 for more details. Δ

Benefits of Using Compute Services with RSUBMIT

One or more remote machines on your network may have vector processors or other larger, faster hardware resources that would more efficiently accomplish the CPU intensive portions of your application. Compute services enable you to move any or all segments of an application to a remote machine for execution. The results of the remote execution are returned to the local SAS session for processing, management by the local graphical user interface (GUI), or both.

In addition to CPU resources, a remote machine may have peripherals attached that would otherwise be inaccessible to you on a stand-alone local system. A PC in another

part of your organization may have a plotter or a printer attached to it that you need to use. By directing your local SAS session to move its graphics processing to the remote PC and pointing to the PC's output device, you can get a hard copy of your graph from the remote plotter.

Data center rules or data characteristics may mandate a single, centralized copy of the data that is needed by your application. By moving the processing to the remote system where the data resides, there is no need to transfer or create additional copies of the data. Using only one copy of data can fulfill security needs as well as enable access to data sources that are too large or too dynamic to be transferred.

For example, although data links between host systems make file transfers convenient and easy, large files do not move quickly between hosts. Furthermore, it is inefficient to maintain multiple copies of large files when developing and testing programs that are designed to process those files. With SAS/CONNECT you can overcome this limitation by developing programs on one system while running them and keeping the data that they use on a different system. (The programs run on the remote host when they are remote-submitted.)

To test your programs, you execute the SAS program on the remote host by remote-submitting it from the local SAS session. All processing takes place on the remote host computer where the data resides, but the output appears on your local host. This method requires no file transfer, yet it permits you to process data that is stored on one system using programs that were developed on another system.

The ability to execute remote submits asynchronously allows you to continue processing on your local host while the remote submit processes in the background. Asynchronous processing provides increased time efficiency by allowing you to perform multiple tasks at the same time. Using this technique, you can start a long running task in the background on a remote host, and immediately be able to begin another task on another remote host or continue local processing, rather than wait until the first remote task is complete before regaining control of your local SAS session. This also provides you with flexibility as to when and where tasks are performed.

Both synchronous and asynchronous remote submits can be used with the Output Delivery System (ODS) to make changes to the format and appearance of the SAS output that is generated on the remote host. Using ODS, each piece of output is an "object" that can be manipulated and viewed in many ways.

Considerations for Using Compute Services with RSUBMIT

Using compute services may introduce some effects that conflict with the goals of your distributed application. This section explains these side effects and offers alternatives to eliminate or minimize them.

Compute services require remote CPU cycles. Because the processing takes place on the remote machine, it requires CPU cycles from that machine. This could be a drawback if the goal is to offload work from the remote system.

You can minimize the remote CPU impact by using data transfer services to transfer a copy of the data to the local machine while maintaining the processing on the local CPU or by combining compute services and data transfer services. See Chapter 7, "Examples of Combining Compute Services and Data Transfer Services," on page 55 for more information.

Compute services also create network traffic in order to return the results of the processing to the local machine. If the processing needs to be repeated frequently, the best solution is to use data transfer services to transfer the necessary data to the local system and maintain the processing in the local environment.

Another effect of using compute services is its demand on remote production data systems as the data is subset or pre-processed. You can minimize this effect by careful implementation of your application to eliminate or to reduce multiple passes of the data.

How to Use Compute Services

Synchronous Remote Submits

Compute Services allows you to remote submit SAS/CONNECT tasks to execute on a remote host. You can take advantage of more powerful processors and other resources that are available on your network. By default, control over your local SAS session is not regained until the remote processing is complete. The sequential nature of this type of processing is called *synchronous processing*. The results from synchronous processing are automatically transferred to your local host upon completion.

The SAS/CONNECT asynchronous processing feature allows you to perform the remote processing in parallel to your local processing and retrieve the accumulated log and output at a later time.

Synchronous processing may be more advantageous than asynchronous processing in some situations, such as when a task is known to execute relatively quickly on a remote host. Because the results of the remote processing are automatically transferred back to the local session, this alleviates any extra steps you must take to retrieve your results and ensures that the results are returned as soon as they are available.

If you use the WAIT=YES option, the local SAS session will wait until the remote processing is complete before returning local control to you. This is the default processing technique for RSUBMIT if the WAIT= option is not specified.

Asynchronous Remote Submits

Asynchronous processing allows you to remote submit SAS/CONNECT tasks and immediately resume local processing. By using the WAIT=NO option in the RSUBMIT statement, you indicate that the local SAS session is *not* to wait until completion of the remote task before returning local control to you. Therefore, you can continue local processing (this includes submitting additional processes to other remote hosts) immediately.

SAS/CONNECT stores the accumulated log and output lines from the remote process until you request the data by using the RGET command. After the RGET command is issued, the accumulated output is retrieved and merged with the local output.

By using the RDISPLAY command, you can view the contents of the accumulated log and output without interrupting the asynchronous processing of the remote submit. After the RDISPLAY command is issued, two windows are created. One window displays the contents of the accumulated log and the other window displays the accumulated output. The presentation of two windows provides a means to view accumulated log and output without merging them into your local log and output windows.

You can also specify a macro variable by using the MACVAR= option in the RSUBMIT statement. This macro variable can be tested to see if the remote processing has finished.

Note: The system option `_LAST_`, which is used to specify the name of the most recently created data set, is not updated for asynchronous remote submits. Many

intervening DATA steps and/or procedures could have been processed before the asynchronous remote submit completes, thus making the system option `_LAST_` meaningless for the asynchronous submission. Δ

MP CONNECT

Version 8 introduces support for parallel (or multi-) processing with Multi-Process (MP) CONNECT. This facility exploits a local host's multi-processor capability by allowing parallel processing of self-contained tasks and the coordination of all the results into the original SAS session. SAS/CONNECT accomplishes multi-processing by means of a new SASCMD option, which establishes a connection to one or more "remote" SAS sessions that run on the local host. MP CONNECT's "remote" session actually executes on the local host.

Note: MP CONNECT is available with the TCP/IP access method only. Δ

To exploit MP CONNECT, you use options in the SIGNON and RSUBMIT statements to identify one or more tasks to be processed by a "remote" SAS session. The remote host identifier is supplied by either the REMOTE= or the PROCESS= option, which is an alias for REMOTE=. In this context, the identifier does not identify the remote host but is an arbitrary name that you associate with a specific task.

MP CONNECT provides the SASCMD= option, which enables signon to the local host's processor. The WAITFOR statement allows you to wait for one or more asynchronously executing tasks before returning control to the local session. The LISTTASK statement lists all active or completed tasks.

Output Delivery System

The Output Delivery System (ODS) is used to manage the results of a procedure by allowing you to make changes to the format and appearance of SAS output. Using compute services, you can generate ODS output from a remote host when using either a synchronous or an asynchronous remote submit. ODS statements and procedures that support ODS can be used to

- convert each piece of output into an "object" that can be manipulated and viewed
- specify options for different ODS destinations, such as HTML or a SAS data set
- create or select templates to customize the output
- select and exclude output from the ODS object.

For more information about the Output Delivery System, see *The Complete Guide to the SAS Output Delivery System*.

There are four basic ways to use compute services and ODS to manage the output that is produced on a remote host.

- 1 Remote submit ODS statements and procedure statements to produce the ODS output on the remote host.

Processing and output generation takes place entirely on the remote host. Therefore, the resulting output (for example, HTML or a data set) must be downloaded from the remote host to the local host.

- 2 Remote submit procedure statements without any ODS statements.

Any output that is produced by the remote submit produces a node in the Results window that has the name `Rsubmit: remote-session-id`. The Results window uses ODS to generate pointers (nodes) to various locations in the Output window. The resulting node is a record of the output generated during a SAS session. A remote submit node can be accessed by clicking the plus sign (+) next to

the **Rsubmit** line using the left mouse button. The subtitle **Rsubmit Output** will appear as the next item in the tree structure. Using the right mouse button, select the subtitle line and open the file by using the pop-up menu. The Output window will be refocused to the top of the remote submit output.

Note: If an asynchronous remote submit is used, the node does not appear in the Results window until RGET is executed. Δ

- 3 Enclose the remote submit block (RSUBMIT through ENDRSUBMIT) with ODS statements.

The output from the remote submit will be saved into an HTML object. If the object is selected, a browser is invoked and the output is displayed in a simple monospace listing format.

Note: For a condition when a monospace listing is not suitable, refer to the first option for managing the output from a remote host, remote submit the ODS statement, and download the results. Δ

If an asynchronous remote submit is used, the **ods html** statements must enclose the RGET statement.

- 4 Precede RSUBMIT with the **ods data set** statement.

The output from the remote submit appears in the Output window and is saved as a SAS data set.

If an asynchronous remote submit is used, the **ods data set** statements must enclose the RGET statement.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/CONNECT User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 537.

SAS/CONNECT User's Guide, Version 8

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

ISBN 1-58025-477-2

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[®], 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.