



CHAPTER

16

Using Data Transfer Services

<i>Introduction</i>	103
<i>Benefits of Data Transfer Services</i>	103
<i>Considerations for Using Data Transfer Services</i>	104

Introduction

For many applications, data transfer services offer the maximum benefit. SAS/CONNECT provides these services by using the UPLOAD and DOWNLOAD procedures. Presented here are scenarios that are well-suited to a data transfer solution.

However, if you plan to transfer SAS files between two Version 7 or Version 8 SAS sessions, you may be able to take advantage of the cross-environment data access (CEDA) facility. This facility is a more efficient way to transfer data between different types of hosts. For scenarios that use CEDA, see Chapter 13, “Using Cross-Environment Data Access (CEDA),” on page 93

PROC UPLOAD and PROC DOWNLOAD in SAS/CONNECT permit a transferred SAS data set to inherit the characteristics of the input data set. If the OUT= option is omitted when transferring a specific SAS data set, then the transferred data set inherits the characteristics of the input data set. A transferred data set will also inherit the characteristics of the input data set if it is part of a library transfer (See the INLIB= and OUTLIB= options for PROC UPLOAD and PROC DOWNLOAD).

Note: Before using data transfer 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 Data Transfer Services

A major benefit of data transfer services is to offload work from a remote system to one or more local machines and boost response time for production systems that run in the remote environment. After the data is downloaded to the local machine, all subsequent data access and processing is done by the local processor.

By moving a copy or subset of the original data to the local machine, production applications can be run on the local machine without further remote CPU consumption or impact on remote production data systems.

Moving a copy of the data to your local system adds robustness to your decision support environment. In the case of a network failure that would temporarily eliminate access to the remote data, you can continue working with your local copy of the data.

You can transfer only the data that you need by using WHERE processing, data set options, or both to dynamically subset the data as it is being transferred to the local

machine. This reduces network traffic and gives you only the data that you need on the local system.

Data transfer services not only facilitate moving data from a larger source to a requesting local workstation, but also support the model of a centralized control point, such as a mainframe, which initiates communication to a network of workstations.

This model enables centralized distribution of data and applications. Automated jobs can be run during non-peak hours to distribute data and applications to multiple machines that need the data and the applications for the next day's work. Similarly, jobs can be set up to query a network of workstations for the purpose of gathering data and storing it in a centralized repository.

Another reason to use data transfer services is for backup purposes. Data and applications can be copied from a local system that has small memory resources to a remote system that has more memory resources. This provides a backup in case of loss on the local system.

Applications developers can also make use of data transfer services. Many applications are developed to run on a remote host and take advantage of its compute services. Data transfer services reduce the workload on your remote host during development of these applications because you can use the local host as a program development tool. You can perform program editing, testing, and debugging by using the local host's resources instead of the remote host's resources.

The development environments at many computing installations often have a higher number of users who work on one system than on other systems. On the system with the heaviest load, response time, execution queues, and other performance factors are less efficient because so many people are running applications concurrently.

Using data transfer services, you avoid contention for heavily used host resources by creating and testing SAS programs on a less busy system (your *local* system), and then remote-submitting the fully developed and tested program to the heavily loaded system (your *remote* system).

Each time you execute a program in a local host SAS session for testing purposes, you avoid adding to the load on the remote host computer. This method can result in significant savings of remote host resources and convenience for you.

For example, suppose you are developing a SAS program that will run as a production program on the remote host. Your program analyzes data in a SAS data set on the remote host system and creates several reports from the analysis information. To run many tests of the program before it is perfected and avoid the delays involved in working on the remote host, create and store the SAS program on your local host. Test the program by downloading the SAS data set that is analyzed by the program, or test the program by using data that is stored in the local host. After the program is complete and correct, upload the program file to the remote host.

Considerations for Using Data Transfer Services

Depending on the goals of your application, data transfer services may have side effects that conflict with your needs. This section addresses these side effects and offers alternatives to eliminate or minimize them.

Transferring a copy of the data to another file system creates multiple copies of the data. If the remote data is updated frequently, it may not be possible to keep a local copy of the data current enough to be useful. In addition, security restrictions at your site may prohibit multiple copies of the data. In this case, consider using compute services if the amount of data that is involved is large.

If the amount of data is small to medium, RLS allows the processing to take place on the local system and the data to come from a remote source as the execution requests it. Both of these alternatives eliminate the need for multiple copies of data.

Keep in mind that data transfer services are less transparent than RLS. Transferring the data is an explicit step that must be done prior to initiating any local processing. If the volume of data to be accessed is not large and your application requires more transparent access, use RLS.

There may also be situations in which a combination of services is the best choice. For examples of combined services, see Chapter 7, “Examples of Combining Compute Services and Data Transfer Services,” on page 55 and Chapter 12, “Example of Combining Remote Library Services (RLS) and Data Transfer Services,” on page 89.

To understand the examples in this section, you must be familiar with the syntax for the UPLOAD and DOWNLOAD procedures (described in Chapter 17, “The UPLOAD Procedure,” on page 107 and Chapter 18, “The DOWNLOAD Procedure,” on page 129).

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.