The Version 6 Implementation of SAS/SHARE® Software in the CMS Environment
Ronald O. Page, SAS Institute Inc., Cary, NC

ABSTRACT
The purpose of this paper is to discuss the implementation of SAS/SHARE® software in the VM/CMS environment. The paper’s focus is the access methods employed by SAS/SHARE software for data transfer within the VM/CMS environment. The first section of the paper examines the communication model implemented by SAS/SHARE software and studies the relationship between a user application and a remote server. The next section of the paper investigates the internals of the IUCV access method. This access method is currently implemented within the experimental Release 6.06 SAS/SHARE software. The details and limitations of this access method are examined. Next, the APPCVM access method is presented. This access method employs APPCVM services that are a part of the SAA architecture for distributed transaction processing developed by IBM®. Lastly, this paper explores the future of SAS/SHARE software as it relates to VM/CMS. These topics are presented to provide a better understanding of SAS/SHARE software and to help identify the future plans for this product in the VM/CMS environment.

INTRODUCTION
SAS/SHARE software was first introduced in Version 6 of the SAS® System and quickly found a niche within user applications that required concurrent access to SAS data libraries with the ability to read and write data. With the advent of MultiVendor Architecture®, Release 6.06 of the SAS System was released in early 1990, and on its heels came the newest implementation of SAS/SHARE software, shipped as an experimental product for the MVS and VM/CMS operating systems.

The concept of sharing SAS data libraries has become commonplace for many SAS/SHARE users, but oftentimes they might easily overlook the underlying architecture that allows the sharing of data to take place. The purpose of this paper is to explain some of the details involved in the transferal of data between a SAS user and a SAS library service machine. Furthermore, a description of the underlying access methods is supplied to give you a feel for the facilities that each access method provides and to offer a glimpse of the functionality that is planned in future releases of SAS/SHARE software.

The following is a list of terms used in this paper:

- **user or source environment** is the environment from which a request for work services originates and is communicated to the server. All SAS applications run in a user environment.

- **service machine or server** is a special user set up to handle requests and perform services for one or more users.

- **access method** is a set of specific routines, usually combined into a single executable image, that provides a service or gives added functionality to a calling program.

- **asynchronous interrupt** is an incoming signal of a pending event that occurs upon arrival at a communication node and is independent of other signals being sent to the node.

- **concurrent access** is the ability to access and update data from within multiple user environments.

**THE SAS/SHARE COMMUNICATION MODEL**

**A Review of SAS/SHARE Software**
SAS/SHARE software is a specialized product offered to users who wish to have concurrent update access to their SAS data libraries. With SAS/SHARE software, multiple users may simultaneously access a SAS data set to update existing data or create new observations within a shared data set. The shared data set exists under the control of a service machine, or SAS server, that handles all incoming requests for data set access from multiple users of the SAS System. All requests for access to particular observations within a shared data set are screened to maintain data integrity within the data set. Here is an example:

- While one user is creating, reading, or updating a shared SAS data set, other users can simultaneously create, read, or update other data sets in the shared library.

- While using the FSEDIT procedure, two or more users can update a single shared SAS data set, but not the same observation at the same time.

- While one user is reading or updating an existing shared SAS data set, another user can be using the APPEND procedure to add observations to the end of the shared data set.

Concurrent access to SAS data libraries is controlled by the SERVER procedure. Operating within a separate SAS execution, the SAS server takes input and output requests from its communication partners and updates the desired shared data set directly on behalf of the individual users. All of this is done in a manner conducive to maintaining absolute data integrity.

As you can see, SAS/SHARE software offers many benefits to users who desire concurrent access to their SAS data libraries. To understand better how SAS/SHARE software accomplishes these tasks, you must look further into the communication model.

**The Communication Model Description**
The SAS/SHARE communication model consists of at least one SAS server surrounded by multiple user machines. The users access a particular SAS server by submitting a LIBNAME statement and providing the name of the desired SAS server with the SERVER option. After a communication link is established between a user and a SAS server, the user is given the ability to access shared SAS data libraries owned by that particular server machine. The SAS server maintains a dialog with each individual user, processing input and output requests as they arrive. Figure 1 provides a graphic representation of the communication model employed by SAS/SHARE software.
Within each user environment, SAS/SHARE software provides input, output, and update access to shared data libraries residing under a SAS server's control. As previously mentioned, the LIBNAME statement, when specified providing the SERVER option, provides an interface in which SAS Software users can access one or more shared SAS libraries owned by one or more SAS servers. For example, you can access the PROFITS library located on the A disk of SERVER1 by issuing the following LIBNAME statement:

```
LIBNAME PROFITS REMOTE 'A' SERVER=SERVER1;
```

Please note that the REMOTE engine name is optional in this example but is provided here for completeness.

After the LIBNAME statement is submitted, the SAS System passes control to SAS/SHARE software, which then attempts to communicate with the SERVER1 server through a predetermined access method. This access method provides host-specific services the user environment needs to communicate with the SAS server. Figure 2 illustrates how the LIBNAME statement signals the host-specific access method to set up communications with the specified SAS server (SERVER1).

An access method is chosen for communication software depending on the specification of the COMAMID option within the SAS System. The default value of the COMAMID option in Release 6.06 of the SAS System is IUCV. As the number of available access methods increases, you will be able to select an access method based on your particular needs. To illustrate the abilities of these individual access methods better, a description of the currently available and planned access methods is provided in the following sections.

### THE IUCV ACCESS METHOD FOR SHARING DATA

#### Overview

The IUCV access method is implemented using the Inter-User Communications Vehicle under VM/CMS. This access method allows two virtual machines running under a common Control Program or CP to communicate by passing virtually limitless amounts of data between them in full duplex. SAS/SHARE software was first implemented using IUCV services in Version 5 of the SAS System under VM/CMS. Likewise, Release 6.06 of the SAS System provides an IUCV access method as part of the experimental SAS/SHARE software under VM/CMS. To understand how the IUCV access works, let us examine the particulars of the IUCV support under CP and CMS.

The first step in establishing an IUCV connection between a user and a server is to identify an Interrupt handler to receive control when an IUCV interrupt is presented to the user environment by the operating system. After this handler is established, an IUCV connection is attempted and is not completed until the partner environment receives notification that an attempt to connect is in progress and accepts the pending connection. Within SAS/SHARE software, this series of events is initiated by the presence of a LIBNAME statement with the name of the desired SAS server provided by the SERVER option.

Once a communication path is established, a user may begin the process of communicating with the server through the sending and receiving of messages. Each data transfer is uniquely identified within the operating system by an assigned message identifier. IUCV services require that all data transfers originate within the user environment. As a result, data that flow from the service machine back to a user must always be in reply to a previous data transmission from that user. Both the user and server environments within SAS/SHARE software receive asynchronous interrupts describing the types of data flows that are being presented. Since the interrupt mechanism is asynchronous, a SAS server can accommodate the needs of multiple users at once because most users make requests for data services at short, regular intervals and seldom require a server's attention for extended periods of time. The following example shows how IUCV services are used within SAS/SHARE software.

#### An IUCV Example

Perhaps a user of SAS/SHARE software issues a LIBNAME statement for the SALES library located under the control of the SAS server SERVER1, and then proceeds to use the FSEdit procedure to examine and edit the FEB90 data set in the SALES library. First, a connection is attempted from the user environment to the SERVER1 server. If the connection is accepted by the SAS server, the user environment begins by sending data to the server, requesting that the SALES.FEB90 data set be opened in preparation for update access. Confirmation messages flow back from the SAS server, letting the user environment know that the desired data set has been opened successfully. As the user peruses each observation of the SALES.FEB90 data set, SAS/SHARE software, with the help of the underlying IUCV access method, sends requests for particular data portions of the data set and receives the requested data from the SAS server through incoming replies. When updates to the

![Figure 1 The SAS/SHARE Communication Model](image)

![Figure 2 Establishing Communication with a Server](image)
SALES.FEB90 data set are made by the user, these changes are relayed to the SAS server where the changes become part of the actual data set. SAS/SHARE software maintains the integrity of the data set by allowing only a single user to edit a particular observation within the data set during any given moment.

The IUCV Limitations
The IUCV access method provides the layer of communications support needed to run SAS/SHARE software in the VM/CMS environment. This access method is an extremely useful tool for transferring data between two VM/CMS users. However, IUCV services have limitations as well.

While the IUCV access method is a preferred method of transferring data between two VM/CMS users, these services are limited in their scope. Unlike other access methods that support cross-system communication between users, the IUCV access method does not support communication between VM/CMS users residing on separate hardware platforms, or between a VM/CMS user and a user of another operating system. While this limitation might not create immediate problems for users of the IUCV access method, it is exciting to see other access methods that do support communication between users of distributed systems. One such access method is the APPC/VM access method.

THE APPC/VM ACCESS METHOD FOR SHARING DATA

Overview
The APPC/VM access method is implemented using the Advanced Program-to-Program Communication facility present within VM/SP Release 6.07 of the SAS System, can be used to communicate with any other access method in any other domain that supports the LU 6.2 protocol. APPC/VM support makes use of general IUCV services and enforces a half duplex protocol through the definition of program states present within each communication partner.

For instance, while a user is sending data to a SAS server, the server must be in receive state and waiting for the incoming data. As dictated by the LU 6.2 protocol, APPC/VM communication takes place over a conversation and is coupled at each end by a transaction program. As soon as incoming data are presented to and interpreted by the service machine, the SAS server can process the request of the individual user. Like IUCV communications, APPC/VM events are presented to the SAS server through an interrupt handler asynchronously to improve the efficiency of the server as it handles the requests of multiple users.

A Further Description of APPC/VM
Since the APPC/VM protocol is half duplex, the presence of state changes is very important in the flow of data. As you learned in a previous section, IUCV services dictate that all data flows originate from within the user environment. The process of getting data from a SAS server back to the originating user is by replying to a previous data transmission. State changes do not occur within IUCV services because the user environment is always the sender of data and the SAS server is a permanent receiver of this data. This is not the case with the APPC/VM access method. Since the protocol is half duplex, either partner in the conversation is capable of sending or receiving data. The process of sending and receiving is dictated only by the current state of the partners participating in the conversation.

As with the IUCV access method, the APPC/VM access method attempts to connect to a desired SAS server when a LIBNAME statement is submitted to the SAS System, specifying a desired server with the SERVER option. At this point, an APPC/VM connection is attempted with the chosen server and if the communications partner accepts the connection, a conversation is established. Since the APPC/VM access method has the ability to communicate across system boundaries, this connection request might be handled by an intermediate resource such as Transparent Services Access Facility (TSAF) or Virtual Telecommunications Access Method (VTAM). If the names imply, this is done transparently and should appear no differently to a user then communication with a service machine present in the originating operating environment. With all of the new functionality of the APPC/VM access method, it is interesting to examine the impact that this access method will have on the future of SAS/SHARE software in the VM/CMS environment.

The Impact of APPC/VM on SAS/SHARE Software
Today, more and more computing sites are quickly moving toward distributed computing environments. It has become fairly commonplace for users to communicate with other systems, accessing and updating data on other hardware platforms. The APPC/VM access method places VM/CMS users of the SAS System in such an arena. Much like the IUCV access method, the APPC/VM access method allows the users of SAS/SHARE software to access data libraries owned by SAS servers present on a single VM system. However, the APPC/VM access method additionally allows users of SAS/SHARE software to access SAS libraries present on other hardware platforms.

For example, a VM/CMS user, specifying the APPC/VM access method within the SAS System, can access a SAS library present on a neighboring MVS operating system and update the library while other MVS users are accessing and updating the library. Another example of the usefulness of the APPC/VM access method is that it allows an OS/2® user of the SAS System to access and update a SAS library present on a VM/CMS system. Users of the SAS System can gain access to any SAS library residing on any connected operating environment through the specification of the LIBNAME statement. Since APPC/VM is part of the IBM SAA® platform and it is an approved IBM API, the future of the APPC/VM access method holds great potential for the users of SAS/SHARE software under Release 6.07 of the SAS System and beyond. Aside from the new APPC/VM access method, other access methods are planned to enhance the efficiency and connectivity of SAS/SHARE software. The next section explores the future development of access methods within the VM/CMS environment.

FUTURE ACCESS METHOD DEVELOPMENT

Shared Data Space Exploitation
VM/ESA® Release 1.1 introduces a facility known as VM Data Spaces. This facility extends the amount of virtual storage available to a user environment and also allows the sharing of data between multiple virtual machines. There are many useful and exciting applications planned to exploit this facility. One such application is a planned SAS/SHARE access method under VM/CMS. If offered, this access method would provide increased performance through the sharing of data between a user of SAS/SHARE software and a SAS server.

This new functionality of the VM/ESA operating system holds much potential for improving the performance of SAS/SHARE software by reducing the size and amount of data transmissions that occur between a SAS/SHARE software user and a SAS server. Much like the IUCV access method, the users of an access method to exploit VM Data Spaces are restricted to communicating only with SAS servers located within the originating VM system. However, the expected performance benefits gained through the reduction in data transmission might make this access method a useful addition to
the growing number of access methods planned for SAS/SHARE software in the VM/CMS environment.

A TCP/IP Access Method

Although this communication protocol has not been explored as an alternative access method for SAS/SHARE software, the Transmission Control Protocol/Internet Protocol (TCP/IP) is certainly a viable possibility for the future. Because of its widespread acceptance in the area of networking and connectivity, an access method to support TCP/IP communication could prove to be a sound addition to the list of access methods available for SAS/SHARE software users on VM/CMS.

CONCLUSION

This paper has explored the implementation of SAS/SHARE software within the VM/CMS environment. The SAS/SHARE communication model was discussed and an explanation of the relationship between a user of the SAS System and a SAS server was presented. The importance of the LIBNAME statement within SAS/SHARE software was also discussed. An examination of the IUCV and APPC/VM access methods was provided to help you better understand the strengths and limitations of these individual access methods.

Also discussed in this paper were future plans for SAS/SHARE software within the VM/CMS environment. This section revealed some of the access methods that are planned for future releases of the SAS System on VM/CMS. An access method to exploit the functionality of VM Data Spaces was presented as a possible future direction. Also, the possibility for a future TCP/IP access method was discussed. Although there is no commitment to develop access methods to exploit these communication protocols, these and other transmission protocols are under consideration. I hope this paper has provided you with a better understanding of SAS/SHARE software and a keen sense of how the underlying access methods support this product.

SAS and SAS/SHARE are registered trademarks and Multi-Vendor Architecture is a trademark of SAS Institute Inc. Cary, NC, USA. IBM and OS/2 are registered trademarks and VM/ESA and SAA are trademarks of International Business Machines Corporation.