

Integrating the SAS® System with other Statistical Applications

George Matthews, University of Georgia, Athens GA 30602-1911

Abstract

As a statistical software consultant in an academic setting, I often find myself responding to questions about which statistical package is best for certain tasks. SAS users tend to want to see their data in SAS, partly because of its familiarity and partly because it offers a powerful data manipulation tool. In many academic settings, other statistical packages are utilized. This paper will explore the aspects of using Dynamic Data Exchange (DDE) as a means of integrating the SAS System with other windows-based statistical packages. Alternate methods, like OLE automation and ODBC, will not be discussed. The following tasks will be discussed:

- ☞ utilizing DDE to transfer data from SPSS® for Windows 6.1.3 to SAS 6.11 for Windows
- ☞ utilizing SAS to create an output dataset with summary statistics (means)
- ☞ transferring the summarized data to Minitab® 10.5 for Windows using DDE

Introduction

The implementation of Dynamic Data Exchange (DDE) first appeared in 1986 with the release of Windows® version 1.03. A migrated implementation of the Windows DDE protocol first appeared in version 1.2 of the OS/2® operating system in 1987. A client-server implementation of DDE for the SAS System first appeared in SAS for Windows 6.08 (OS/2 was released around the same time frame) in 1992. In Release 6.08, 6.10 and 6.11 of the SAS System for Windows and OS/2, SAS is always the client. This means that SAS must initiate the process of reading data from other applications and/or writing data to other applications. SPSS is available only as a server for DDE--that is, client applications like SAS can obtain information from SPSS. Minitab can function as a client, a server, or both at once - receiving data from other applications and/or sending data to other applications. Data can be passed between the SAS System and these applications using DDE.

Overview of Dynamic Data Exchange

To utilize DDE, you must establish a conversation between a client (an application that sends and receives data to or from another application) and a server (an application that responds to the client application's request by providing the necessary information). Once a DDE conversation has been established, client applications can request services from a server in what are known as transactions. A DDE transaction is the exchange of DDE messages between a client and a server application. DDE transactions are normally one-way; that is, a client application initiates the transaction and a server always responds to the transaction. The only exception to this rule is the terminate transaction, which can be initiated by either the client or the server application. DDE transactions are divided into the following six categories:

- ☞ **Request** - to request a one-time data transfer, referred to as a cold-link.
- ☞ **Advise** - to request a permanent data-link, referred to as a *warm* or *hot link*.
A *warm link* notifies the client that a change has been made, but the client must specifically make a request to the server for the data to be updated.
A *hot link* automatically updates the receiving file when the data in the source changes.
- ☞ **Unadvise** - to request to terminate a data-link.
- ☞ **Poke** - to request that a data item be updated.
- ☞ **Execute** - to request that a command or commands be executed.
- ☞ **Terminate** - to request to terminate a conversation link.

A DDE transaction message consists of the following items:

- ☞ A message identifier indicating the type of transaction.

 The window handle of the application that originated the message.

To utilize DDE to exchange information, a client application must specify three elements: the *application name*, a *topic*, and an *item* within the topic. The *application name* is the name of the server application. The *topic* is the topic of conversation. The *item* is the range of conversation specified between the client and server applications. These three elements are commonly referred to as the DDE-triplet. The values for these three elements is defined by the server application.

Description of the Data

The data was taken from an SPSS sample data file (bank.sav). The .sav file consisted of 11 variables and a 474 cases. The variable names and a description of each variable is listed in Table #1.

Table 1

<u>Variables</u>	<u>Description</u>
id	employee code
salbeg	beginning salary
sex	sex of employee
time	job seniority
age	age of employee
salnow	current salary
edlevel	educational level
work	work experience
jobcat	employment category
minority	minority classification
sexrace	sex & race classification

This data represented information about 474 employees hired by a bank between 1969 and 1971.

Transferring Data from SPSS

DDE requires both applications to be active, but one application can start the other. A special FILENAME statement is utilized by SAS to establish a link to the SPSS data. The most general format of the FILENAME statement is:

```
FILENAME fileref DDE 'DDE-triplet';
```

The fileref can be any valid SAS name, DDE is a device-type keyword that tells SAS you want to use DDE, and the DDE-triplet is defined by the server application. The SAS program for reading our SPSS data file follows:

```

/*****
/*   S E S U G 96 S A S P R O G R A M   */
/*                                     */
/*  NAME: MYDDE                       */
/*  TITLE: USING DDE TO TRANSFER DATA */
/*  PRODUCT: SAS                      */
/*  SYSTEM: WINDOWS                   */
/*  KEYS: DYNAMIC DATA EXCHANGE     */
/*  PROCS: PRINT                      */
/*  DATA: BANK.SAV                   */
/*                                     */
/*  DESC: DEMONSTRATES HOW TO USE     */
/*  DDE TO TRANSFER DATA FROM SPSS   */
/*  TO SAS. TO EXECUTE THIS PROGRAM   */
/*  YOU MUST FIRST INVOKE SPSS AND   */
/*  OPEN UP BANK.SAV                 */
*****/

```

```

filename mydata dde
'spsswin|bank.sav!tabbedcontents';
data one;
infile mydata firstobs=2;
input id salbeg sex time age salnow edlevel
work jobcat minority
sexrace;
proc print;
run;

```

Summarizing your Data

Once the data has been transferred into SAS via DDE, you can proceed with your analysis. A common task for many SAS users is to generate output files that can be utilized as input to other statistical packages. The following code demonstrates how to generate means of our salary information, output those means to a second data set and create a new data set with both new and original variables.

```

/* Calculate the mean salaries          */
proc means data=one mean noprint;      */
var salbeg salnow;                    */
output out=test mean=avgbeg avgnow;   */
run;
/* Create a new data set with original  */
variables                               */
data both(keep=salbeg avgbeg diffbeg  */
avgnow diffnow);                      */
set one;                               */
retain;                                */
diffbeg=salbeg-avgbeg;                */
diffnow=salnow-avgnow;                */
if _n_=1 then set test;               */
/* Print the resulting data set        */
proc print data=both;                  */
run;

```

Now that we have transformed the data in SAS, we can transfer the data to Minitab.

Transferring Data to Minitab

The previous DDE example demonstrated the technique of transferring data while both applications were active. In this example, we will demonstrate how to start-up the server application while working in the client application. To start-up Minitab from within SAS you can use the X statement, as in the following example:

```
options noxwait noxsync;  
x 'c:\mtbwin\mtbw16';
```

The noxwait option keeps you from having to type EXIT to return to SAS, and the noxsync option allows the SAS System to retain control. To transfer the new data set from SAS to Minitab, I used the following code:

```
/******  
/*      SESUG96  SAS PROGRAM          */  
/*      */  
/*  NAME: DDEM                        */  
/*  TITLE: USING DDE TO TRANSFER DATA */  
/*  PRODUCT: SAS                      */  
/*  SYSTEM: WINDOWS                   */  
/*  KEYS: DYNAMIC DATA EXCHANGE     */  
/*  PROCS:                             */  
/*  DATA:                             */  
/*      */  
/*  DESC: DEMONSTRATES HOW TO USE    */  
/*  DDE TO TRANSFER DATA FROM SAS   */  
/*  TO MINITAB.                      */  
/******  
filename out dde 'minitab\data\r1c1:r474c6';  
data one; set both;  
file out;  
put salbeg avgbeg diffbeg salnow avgnow  
diffnow;  
run;
```

Performance Considerations

SAS is a robust DDE client, suitable for use over a wide range of applications, however, several factors outside SAS may affect DDE performance including:

-  the memory limitations and general performance of your machine.
-  the general performance of DDE in Windows.

 the number and size of other applications you have running.

 the performance of DDE in other applications.

If you need to improve DDE performance, consider the following:

 consult the application's documentation or on-line help for information on enhancing performance.

 consult the application's documentation or on-line help for information on using a priority field to prioritize your updates.

 try using or upgrading to a more powerful machine to run the applications.

Conclusion

You have learned the DDE steps necessary to integrate SAS with other statistical applications. DDE provides a method of passing information between statistical applications. It is an excellent example of client/server technology where information can be shared among application software running on the same machine. It is a very powerful tool for sharing data and can save SAS users the time and hassle of learning the perplexities of other statistical packages. Lastly, these DDE steps are transparent under NetDDE, allowing you to extend DDE capabilities across networks.

Acknowledgment

The author wishing to thank Jeannie McElhannon for word processing support and the PC Interface Support Group at SAS Institute for technical support. The author thanks Barbie Matthews, Joe Kelley, and Greg Ashley for moral support throughout the development of this paper. The author can be contacted at:

The University of Georgia
UCNS
Computer Services Annex
Athens, Ga. 30602-1911
(706)542-5359
gmatthew@uga.cc.uga.edu

Further Reading

For more information regarding topics discussed in this paper, refer to these manuals and documents:

SAS Institute Inc. (1993), *Companion for the Microsoft Windows Environment*, Version 6, First Edition, Cary, NC. SAS Institute Inc.

SAS Institute Inc. (1996), *Companion for the Microsoft Windows Environment*, Version 6, Second Edition, Cary, NC. SAS Institute Inc.

SAS Institute Inc. (1993), *Companion for the OS/2 Environment*, Version 6, Second Edition, Cary, NC. SAS Institute Inc.

The SAS System and DDE, TS325.

Bodt, M. (1996), "Talking to PC Applications Using Dynamic Data Exchange," *Observations: The Technical Journal for SAS Software Users*, 5(3) 18-28.

SPSS, Inc. (1993), *SPSS for Windows Base System User's Guide Release 6.0*.

Minitab, Inc. (1994), *The MINITAB Reference Manual*.

Trademarks

SAS is a registered trademark of SAS Institute, Inc., in the USA and other countries. ® indicates USA registration.

SPSS is a registered trademark of SPSS, Inc. in the USA and other countries. ® indicates USA registration

MINITAB is a registered trademark of MINITAB, Inc. In the USA and other countries. ® indicates USA registration.

OS/2 is a registered trademark of IBM, Inc. in the USA and other countries. ® indicates USA registration.

Windows is a registered trademark of Microsoft, Inc. in the USA and other countries. ® indicates USA registration.