Using Dynamic Data Exchange (DDE) to Pass Data to and from Microsoft® Excel Spreadsheets from within the SAS® System

Mark Bodt, SUNKEN TREASURE SOFTWARE SYSTEMS LIMITED, New Zealand

Abstract

Dynamic Data Exchange (DDE) is a method of communicating with PC based applications from within the SAS System. DDE allows the reading and writing of data from or to the application. Indeed, the applications can even be controlled from within the SAS system. While most day-to-day tasks can be achieved within the SAS system, it is sometimes necessary to produce reports, letters etc using other applications such as Microsoft® Excel or Microsoft® Word.

This paper discusses how the use of DDE can achieve the reading and writing of data from or to Microsoft Excel.

Scope

This paper covers topics discussed during the 10 minute Coders’ Corner presentation. The basics of reading and writing of data from or to Excel using DDE with Excel are outlined. However it is not intended to thoroughly discuss the topic. This topic will be covered in depth in Observations Vol. 5, Num 3, Second Quarter 1996

Introduction

Many sites use PC based applications such as Microsoft Word and Microsoft Excel as their standard word processing and spreadsheet applications. These tools are often used to produce standard letters and reports. Output from SAS processing is sometimes required as part of an Excel Spreadsheet. Similarly, data held in an Excel spreadsheet is sometimes required as part of SAS processing. This can be achieved in a number of ways, DDE being one way.

What is Dynamic Data Exchange?

Dynamic Data Exchange (DDE) is a method of dynamically exchanging information between applications. DDE is a feature of the SAS system for:

• Windows™
• Windows NT™
• OS/2®
• Windows 95™

DDE uses a Client/Server Relationship to enable the client to request information from a server. It is available in Version 6.08 (6.08 for OS/2) and later versions. SAS is always the client in Version 6.08, 6.10 and 6.11 of the SAS System. This means that SAS can read data from or write to other applications and that commands can be sent to the other application, but that the process is initiated by SAS. Currently SAS cannot act as a DDE Server, that is, another application cannot initiate the passing of data to or from the SAS system.

Setting up a DDE Link.

To use DDE you must have both SAS and Excel running. It is very easy to set up a DDE link. A special Filename statement is used. The basic syntax is:

Filename fileref='DDE DO-triplet';

Where fileref is the alias that is given to the link. This must be a valid SAS name.

DDE is a special device-type key word.

DDE-triplet is the identifier for the link.

The DDE triplet.

The DDE triplet is generally made up of three parts being:

application-name|topic|item

• application-name is the name of the server application,
• topic is the topic of conversation,
• item is the range of conversation specified between the client and server applications.

The | and ! are special characters that separate the different parts of the triplet.

The values of the DDE triplet are determined by the server application.

For Microsoft Excel:

An example of a triplet is: ‘Excel[a:\s\claims.xls]USAIR6C1:R12C2

• Application name: Excel
• Topic: For Excel version 4: the spreadsheet name eg, a:\sales.xls, or in Excel version 5 the workbook and tabbed sheet eg a:\claims.xls]USA.
• Item: The range of cells eg R6C1:R12C2 being Row 6 Column 1 to Row 12 column 2.

Reading data using DDE.

In this example, data will be read from specific cells in an Excel Version 5 spreadsheet. The spreadsheet will be the one as shown on the next page
The $AS program is as follows:

```
*set options for report;
   Options
   nodate nonumber;
*set up DDE link to read from Excel Spreadsheet;
   filename trills DOE
   Excel\C:\DATA\EXCEL\DDETESTS\[SUGI1.XLS1Trillsa
les! R6Cl: R12C2';
*read in data;
   data sales;
   length city $15,
   infile trills dlm='09'x notab dsd missover;
   input city sales;
*print report;
   proc print;
   var city sales;
   sum sales;
   format sales comma14.2;
   title1 'Trill Sales.';
   title2 '---------------';
run;
```

The DOE link must be established. In this example, the filename is trills. Next is the DOE key word and then the DOE triplet r6c1:r12c2'.

The data is then read in as list input as it would be for any external file. In this case, a dataset called Sales is being created. The city variable has been set to length $15. The input statement reads in the city variable (character) and the sales field.

Tab characters are inserted by Excel to separate the data. The notab option tells SAS not to convert the tab characters into blanks. The dlm='09'x option tells SAS that the delimiter used to separate the data is the tab character. '09'x is the ASCII hexadecimal value for the tab character. The dsd option tells SAS to treat two consecutive delimiters as representing a missing value, and the missover option prevents the SAS program from going to a new input line if there were no values on the current line.

Just to prove the code worked, a simple Proc Print will report on the data that were read in.

The results are listed in the output window:

```
   Trill Sales.
   ----------------
   OBS CITY       SALES
   1 Auckland     523,445.00
   2 Huntly      1,253.00
   3 Hamilton    123,455.00
   4 Tauranga   88,012.00
   5 Waitomo   5,234.00
   6 Rotorua 2,234.00
   7 Thames    593.00
   ----------------
   744,215.00
```

Writing data using DDE

In this example, data will be written from a SAS dataset to a specific cell range in an Excel Version 5 Spreadsheet. The spreadsheet is first loaded as shown on the following page.
The SAS Program is as follows:

```
*Set up DDE file allocation;
filename excel DDE;
'D:DATA\EXCEL\DDETESTS(SUGI2.XLS)Room15\Room15\R4C1:R25C4';
*read from dataset and write to EXCEL Spreadsheet;
data null;
set sasuser.class; •
file excel notab; •
put name '09'x age '09'x sex '09'x weight; •
run;
```

The DDE link is set up in the same way as for the read example. Note that the DDE Triplet reflects the different range and selection spreadsheet name. For the purposes of the example, data from the dataset SASUSER.CLASS will be used. This contains some data about pupils. The statement to write the data to a file is in the same format as with any file write. The filename is excel as defined in the filename statement. By default SAS sends a tab delimiter between each word sent to the DDE link. As Excel uses the tab character to delimit each cell, this means that each word is placed into a separate cell. In some cases a variable may have more than one word as the character value, for example if a person has two christian names ‘Mary Jane’ or ‘Jean Paul’. By default, SAS will put a tab character between the words and Excel will place these into different columns. To overcome this, the NOTAB option is specified. This instructs SAS not to put tab delimiters by default. The data is then transferred to the Excel sheet. The fields name, age, sex and weight are put to the file (DDE link). As SAS is not putting tab delimiters (because of the NOTAB option), the tab delimiter character must be put between each field that is sent to Excel. This means that a tab delimiter is sent between each variable rather than each word that is sent to Excel. 09x is the ASCII Hexadecimal value for the tab delimiter.

The program is run and the data is transferred to the Excel sheet. The results are shown on the next page.
## Weights of pupils in Room 15

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Sex</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>13</td>
<td>F</td>
<td>84</td>
</tr>
<tr>
<td>Becca</td>
<td>13</td>
<td>F</td>
<td>96</td>
</tr>
<tr>
<td>Gail</td>
<td>14</td>
<td>F</td>
<td>70</td>
</tr>
<tr>
<td>Karen</td>
<td>12</td>
<td>F</td>
<td>112</td>
</tr>
<tr>
<td>Kathy</td>
<td>12</td>
<td>F</td>
<td>94.5</td>
</tr>
<tr>
<td>Mary</td>
<td>15</td>
<td>F</td>
<td>112</td>
</tr>
<tr>
<td>Sandy</td>
<td>11</td>
<td>F</td>
<td>70</td>
</tr>
<tr>
<td>Sharon</td>
<td>15</td>
<td>F</td>
<td>112</td>
</tr>
<tr>
<td>Tammy</td>
<td>14</td>
<td>F</td>
<td>102.5</td>
</tr>
<tr>
<td>Alfred</td>
<td>14</td>
<td>M</td>
<td>112</td>
</tr>
<tr>
<td>Duke</td>
<td>14</td>
<td>M</td>
<td>102.5</td>
</tr>
<tr>
<td>Guido</td>
<td>15</td>
<td>M</td>
<td>133</td>
</tr>
<tr>
<td>James</td>
<td>12</td>
<td>M</td>
<td>133</td>
</tr>
<tr>
<td>Jeffrey</td>
<td>13</td>
<td>M</td>
<td>70</td>
</tr>
<tr>
<td>John</td>
<td>12</td>
<td>M</td>
<td>96.5</td>
</tr>
<tr>
<td>Philip</td>
<td>16</td>
<td>M</td>
<td>150</td>
</tr>
<tr>
<td>Robert</td>
<td>12</td>
<td>M</td>
<td>121</td>
</tr>
<tr>
<td>Thomas</td>
<td>11</td>
<td>M</td>
<td>85</td>
</tr>
<tr>
<td>William</td>
<td>15</td>
<td>M</td>
<td>122</td>
</tr>
</tbody>
</table>

**Average Weight:** 100.03

### Conclusion

Passing data to and from Excel from within the SAS system is a straightforward task. This article briefly discussed some simple examples. DDE can also be used to control Excel from within the SAS System, providing the ability to automate applications. Operations such as opening, saving, formatting and printing Excel spreadsheets can be controlled from within the SAS System. This topic will be covered in depth in issue nineteen or twenty of Observations.

### References


### Contact details

Sunken Treasure Software Systems Limited  
Specialising in SAS Software Consultancy for the Asia-Pacific Region  
73 Pine Street Mt Eden Auckland New Zealand  
Phone 025 725 384  Fax +64-9-620-9079  
INTERNET MARKBOTT@STSS.CO.NZ

* SAS and Observations are registered trademarks of SAS Institute Inc. in the USA and other countries. OS/2 is a registered trademark of International Business Machines Corporation. Microsoft, Microsoft Excel, Microsoft Word, Windows and Windows NT are registered trademarks of Microsoft Corporation. © indicates USA registration.