

Reading and Writing Data from Microsoft Excel/Word Using DDE

Destiny Corporation

Prepared by Dana Rafiee

Destiny Corporation • 100 Great Meadow Rd Suite
601• Wethersfield, CT 06109-2355

Phone: (860) 721-1684 • 1-800-7TRAINING

• Fax: (860) 721-9784

Email: info@destinycorp.com Web Site:
www.destinycorp.com

Copyright © 1999 Destiny Corporation. All Rights Reserved.

Dynamic Data Exchange to Excel

Dynamic Data Exchange is a means of transferring data between different Windows applications such as the SAS System and Microsoft Excel or Microsoft Word. The technique uses what is known as a Client / Server type relationship, where the Client application makes the requests and the Server application responds accordingly. When using DDE with the SAS System, SAS is always the Client.

The DDE technique may be used within a Data Step, a Macro or from inside a SAS/AF application.

Using the DDE technique involves specifying the following three components in what is called a DDE Triplet:

1. The path and name of the executable file for the server application. For example, with Microsoft Excel the name of the executable is EXCEL.
2. The full path and file name of the document or spreadsheet with which you wish to share data. This is referred to in SAS documentation as the Topic.
3. The location of data to be read or modified. With a spreadsheet application, this is a range of cells. In applications such as Microsoft Word, the location of data is defined by what are known as bookmarks.

An example of a DDE Triplet for an Excel Spreadsheet is as follows:

EXCEL | C:\COURSES\DATA\CLIN.XLS |
R4C17:R17C8

The DDE Triplet is then incorporated into a Filename statement of the following form:

```
FILENAME fileref DDE 'DDE-Triplet' |  
'CLIPBOARD' <DDE-Options>;
```

where:

fileref is a valid fileref

DDE is the keyword that primes the SAS System for Dynamic Data Exchange.

'DDE-Triplet' is the three part specification of the DDE external file and takes the following form: 'Executable-file-name|Topic|Data-range'. Note that the triplet is application dependent and may differ between different releases of the same application.

'CLIPBOARD' is an alternative to directly specifying the DDE triplet. To use this option, copy the required data from the Server application to the Clipboard and specify the 'CLIPBOARD' option in the Filename statement. SAS will then determine the Triplet for itself. You will be able to see the triplet definition by looking at the Log.

DDE-Options are as follows:

HOTLINK invokes the DDE HOTLINK facility, which causes the DDE link to be activated whenever data in the spreadsheet range is updated.

NOTAB makes the SAS System ignore tab characters between variables.

COMMAND allows remote commands to be sent to DDE server applications.

Example of using DDE to read data from Excel

Consider the following Excel Spreadsheet:

Patient	Sex	Age	Drug	Score 1	Score 2	Score 3	Total Score
1	M	56	CHG1	87	77	88	252
2	F	44	DFR4	45	23	87	155
4	F	34	GTR6	44	34	76	154
7	M	46	CHG1	88	76	79	243
9	M	43	GTR6	54	66	87	197
13	M	67	DFR4	78	87	83	248
15	F	78	CHG1	57	69	63	179
16	F	45	DFR4	68	72	78	218
17	M	78	GTR6	47	58	62	167
18	F	61	DFR4	65	68	73	206
21	F	71	GTR6	68	73	78	219
33	M	41	CHG1	73	81	91	245
36	F	70	DFR4	55	64	76	195
38	M	47	CHG1	85	75	82	242

With Excel running and the above spreadsheet open, a simple Data Step can be written to read the above data into a SAS Data Set. The DDE Triplet components for the above example are as follows:

Name of Executable File: EXCEL

Full Path and Name of spreadsheet file:
C:\COURSES\DATA\CLIN.XLS

Location of data (Row and Column coordinates):
R4C1:R17C8

Note how the two sets of coordinates that define the range of data are separated by a colon.

The SAS statements required to read the data into a SAS data set are as follows:

```

1 INPUT PATNO SEX AGE DRUG SCORE1-SCORE5 TOTAL_SC ;
2 RUN;

NOTE: The infile XLFIL is:
      FILENAME='c:\courses\data\clin.xls';
      RECFM=V,LRECL=256

NOTE: 14 records were read from the infile XLFIL.
      The minimum record length was 25.
      The maximum record length was 25.
NOTE: The data set WORK.CLIN has 14 observations and 8 variables.
NOTE: The DATA statement used 8.71 seconds.

*****
***** /* Assign a Fileref to the Excel Spreadsheet and specify that
***** Dynamic Data Exchange is to be used */
*****
***** FILENAME xlfil DDE 'excel:c:\courses\data\clin.xls;R4C1:R17C8';
*****
***** /* Create a Data Set called CLIN in the Work Library and read in
***** data from the Excel Spreadsheet */
*****
***** DATA WORK.CLIN;
*****   INFILE XLFIL;
*****   INPUT PATNO SEX AGE DRUG SCORE1-SCORE5 TOTAL_SC ;
***** RUN;

```

Running the above code generates the following data set:

PATNO	SEX	AGE	DRUG	SCORE1	SCORE2	SCORE3	TOTAL_SC
1	M	56	CHG1	97	77	86	252
2	F	44	DFR4	45	23	67	135
4	F	34	GTR6	44	34	76	154
7	M	45	CHG1	66	75	79	220
8	M	43	GTR6	54	56	87	197
13	M	67	DFR4	78	87	83	248
15	F	76	CHG1	57	59	63	179
16	F	45	DFR4	68	72	78	218
17	M	70	GTR6	47	58	62	167
18	F	61	DFR4	65	68	73	206
21	F	71	GTR6	68	73	78	219
33	M	41	CHG1	73	91	91	255
36	F	70	DFR4	55	54	76	185
38	M	47	CHG1	65	75	82	222

Example of using DDE to write data to Excel

To output data to an Excel spreadsheet, define a DDE Triplet which contains the co-ordinates of the cells to which data is to be written.

Then use a null data step containing 'Put' statements, to write out the values of required variables to the spreadsheet.

For example, the following code uses a Proc Summary to calculate simple statistics on the data

that was originally read in from Excel. The statistics are output to a data set, which is used to update the Excel spreadsheet:

```

***** /* Service simple statistics for the three score variables */
*****
***** proc summary data=work.clin;
*****   var score1-score3;
*****   output out=work.summary;
***** run;

***** /* Assign a Fileref to the Excel Spreadsheet and specify that
***** Dynamic Data Exchange is to be used */
*****
***** FILENAME xlfil DDE 'excel:c:\courses\data\clin.xls;R4C1:R20C7';
*****
***** /* Output statistics for Score variables to Excel Spreadsheet */
*****
***** data _null_;
*****   file xlfil;
*****   put _stat_ score1 score2 score3;
***** run;

```

Note that the only change to the Filename statement compared with the previous example, is that the DDE Triplet points at a different range of cells in the spreadsheet.

A null data step is used to read the data set containing the statistics and 'Put' statements write out the values of selected variables to the Excel spreadsheet. This is no different to writing out data to a text file.

Note that a null data step is used when you wish to avoid creating another data set.

Running the above code updates the Excel spreadsheet accordingly:

The updated Excel spreadsheet complete with summary statistics:

Patient	Sex	Age	Drug	Score 1	Score 2	Score 3	Total Score
1	M	56	CHG1	97	77	86	252
2	F	44	DFR4	45	23	67	135
4	F	34	GTR6	44	34	76	154
7	M	45	CHG1	66	75	79	220
8	M	43	GTR6	54	56	87	197
13	M	67	DFR4	78	87	83	248
15	F	76	CHG1	57	59	63	179
16	F	45	DFR4	68	72	78	218
17	M	70	GTR6	47	58	62	167
18	F	61	DFR4	65	68	73	206
21	F	71	GTR6	68	73	78	219
33	M	41	CHG1	73	91	91	255
36	F	70	DFR4	55	54	76	185
38	M	47	CHG1	65	75	82	222
N				14	14	14	
MIN				44	23	62	
MAX				97	91	91	
MEAN				62.42857	63.78571	77.35714	
STD				12.71411	18.01359	9.663123	

Invoking an application from SAS

In the previous examples, the invocation of Excel and opening of a spreadsheet file has been done by clicking on the required icon and using the pull down menus within Excel.

HANDS ON WORKSHOPS

The application remains running in the background until the user actively closes it down.

Ideally, you only want an application running for the time that it takes to access the data. Otherwise, the idle application ties up valuable memory that can be better used. The solution to this problem is to open and close applications from your SAS program.

An application may be called from SAS by using the following statements:

```
OPTIONS NOXWAIT NOXSYNC;
```

```
X 'C:\EXCEL\EXCEL';
```

```
DATA _NULL_;  
  T=SLEEP(5);  
RUN;
```

NOXWAIT causes control to be returned to SAS as soon as the specified 'X' command has been executed. In the above example, as soon as Excel has been loaded, control will return to SAS.

NOXSYNC causes the application called from SAS to execute asynchronously so that it does not interfere with the SAS session.

Remember to reset the above options to XWAIT and XSYNC when the application is closed. The X statement contains the path and name of the executable file that calls the application. The **Sleep** statement inside the null data step simply gives the called application time to load. In the above example, 5 seconds has been allowed.

Issuing Commands from SAS

Having invoked an application like Excel using the above statements, you need to issue commands that carry out tasks such as opening the required spreadsheet file and selecting data. This is done through what is called the **SYSTEM** Topic:

Assign a Fileref that includes the DDE topic **SYSTEM** to enable Excel or Word commands to be issued from SAS.

```
FILENAME CMDS DDE 'Excel|SYSTEM';
```

Commands may now be sent to Excel by using 'Put' statements inside a null data step. Note that the syntax of the commands issued is application dependent.

```
DATA _NULL_;
```

```
FILE CMDS;
```

```
PUT
```

```
'[FILE-
```

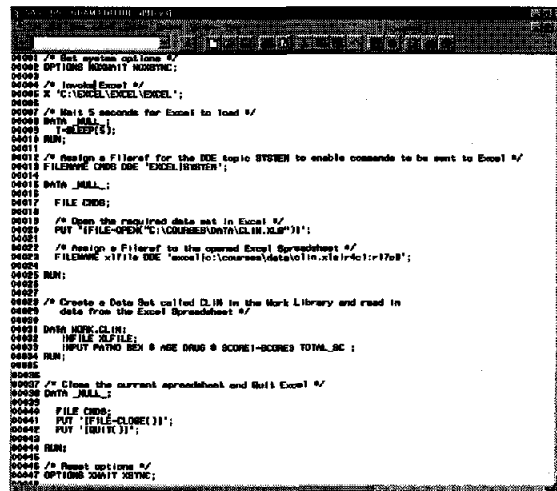
```
OPEN('C:\COURSES\DATA\CLIN.XLS=)']';
```

```
RUN;
```

Example of invoking and issuing commands to Excel

The example program does the following:

1. Invokes Excel.
2. Opens a Spreadsheet.
3. Reads spreadsheet data into a SAS data set.
4. Closes the Spreadsheet.
5. Closes Excel.

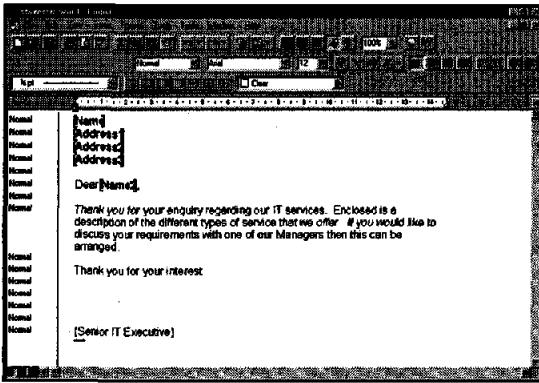


```
0001 /* Set system options */  
0002 OPTIONS NOXWAIT NOXSYNC;  
0003  
0004 /* Invoke Excel */  
0005 X 'C:\EXCEL\EXCEL';  
0006  
0007 /* Wait 5 seconds for Excel to load */  
0008 DATA _NULL_;  
0009   T=SLEEP(5);  
0010 RUN;  
0011  
0012 /* Assign a Fileref for the DDE topic SYSTEM to enable commands to be sent to Excel */  
0013 FILENAME CMDS DDE 'Excel|SYSTEM';  
0014  
0015 DATA _NULL_;  
0016  
0017   FILE CMDS;  
0018  
0019   /* Open the required data set in Excel */  
0020   PUT 'FILE-OPEN "C:\COURSES\DATA\CLIN.XLS"' ;  
0021  
0022   /* Assign a Fileref to the opened Excel Spreadsheet */  
0023   FILENAME xFILE DDE 'Excel|C:\COURSES\DATA\CLIN.XLS|<A1:B100';  
0024  
0025 RUN;  
0026  
0027 /* Create a Data Set called CLIN in the Work Library and read in  
0028 data from the Excel Spreadsheet */  
0029  
0030 DATA WORK.CLIN;  
0031   INFILE xFILE;  
0032   INPUT PATNO SEX AGE DRUG SCORE1-SCORE5 TOTAL_SC ;  
0033 RUN;  
0034  
0035 /* Close the current spreadsheet and quit Excel */  
0036 DATA _NULL_;  
0037  
0038   FILE CMDS;  
0039   PUT 'FILE-CLOSE';  
0040   PUT 'QUIT';  
0041  
0042 RUN;  
0043  
0044 /* Reset options */  
0045 OPTIONS XWAIT XSYNC;
```

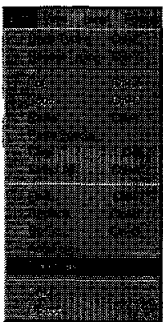
Dynamic Data Exchange to Word

Using DDE to access data stored in a Microsoft Word document is the almost the same procedure as used for Excel. The only difference is that the location of data in Word is specified by referencing Bookmarks rather than a cell range.

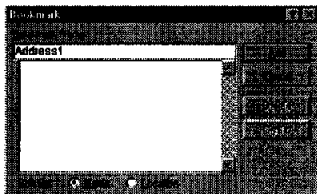
A bookmark is simply a named marker placed within the Word document. The following Word Document uses bookmarks to denote the position of Name and Address fields:



To create a bookmark within a Word document, highlight the text which you want to use as a bookmark, then select **Edit** and **Bookmark** from the pull down menu:



Enter a name for the bookmark and click on **Add**



The components of the DDE Triplet for the above example are as follows:

Name of Executable File: WINWORD

Full Path and Name of Word document:
C:\COURSES\DATA\ENQUIRY.DOC

Location of data (Name of Bookmark): Address1

Invoking Word from SAS

The procedure used to call Word from SAS is the same procedure used to call Excel or any other Windows application from SAS. Set the appropriate options, use an 'X' statement to call Word, and allow Word a few seconds to load.

```
OPTIONS NOXWAIT NOXSYNC;
```

```
X 'C:\WINWORD\WINWORD';
```

```
DATA _NULL_;
```

```
T=SLEEP(5);
```

```
RUN;
```

Remember to reset the above Options to XWAIT and XSYNC when the application is closed.

Issuing Commands to Word from SAS

As with Excel, this is done by assigning a Fileref to the DDE topic SYSTEM:

```
FILENAME CMDS DDE 'WINWORD|SYSTEM';
```

Once the above Fileref is assigned, commands can be sent to Word by using 'Put' statements inside a null data step.

```
DATA _NULL_;
```

```
FILE CMDS;
```

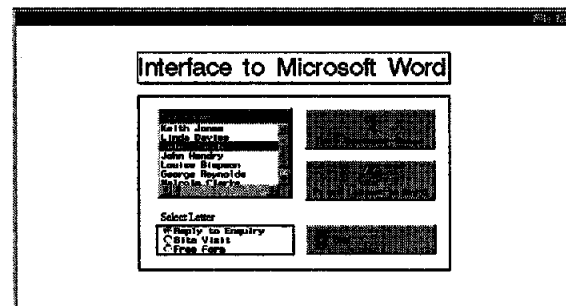
```
PUT
```

```
'[FILEOPEN(\\C:\COURSES\DATA\ENQUIRY.DOC=)]';
```

```
RUN;
```

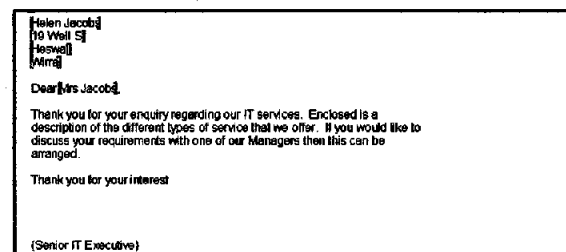
Example of using DDE to access a Word document

The following SAS/AF Application automatically inserts the name and address of a customer into the selected Word document and prints it:



The name and address information is stored in a SAS data set that may be accessed and updated by clicking on the 'Edit Customer Names' icon.

The name and address information of the selected customer is inserted into the Word document as follows:

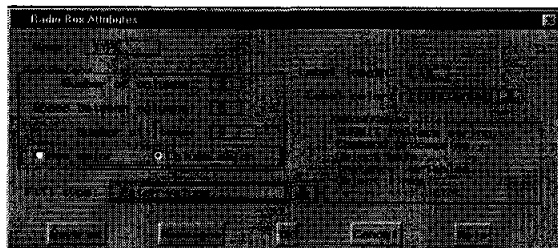


HANDS ON WORKSHOPS

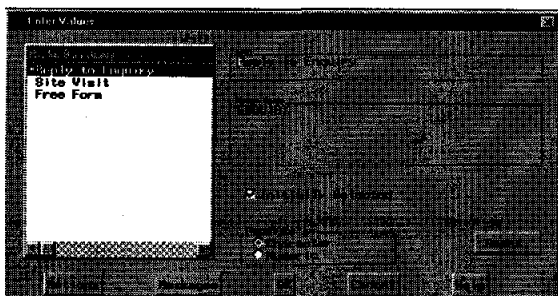
Note how the name and address details have been written to their respective bookmarks. The square brackets, which denote the boundary of a bookmark, do not appear in the printed document.

Note the following about the SCL Program that controls the Frame:

1. The physical paths for the Word executable file and Word documents, are assigned to Macro variables in the Autoexec file which runs before the application is called. Call Symget routines are used in the INIT section to assign the values of these Macro variables to SCL variables. This approach avoids having to hard code path names in the application.
2. The name of the Word document to open is derived by concatenating the document path, letter type, and the suffix '.DOC'.
3. The letter type is obtained by sending the `_GET_TEXT_` method to the Radio Box, which returns the Character value assigned to the selected item. Each Radio Box item has a Character return value specified in the attributes window:



Click on  to access Return values for items on



Clicking on a Radio Box Item displays its Return value.

4. A Submit Block contains all of the SAS code used to load Word, access the selected

document and print it. Recall that an ampersand within a Submit Block references an SCL variable.

5. Five different Filrefs are assigned to point at the various name and address bookmarks in the Word document.

The SCL Program is as follows (see next column):

```

0001 /* Declare the length of the Character variables */
0002 length volume $20 lettertype $4;
0003
0004 INIT:
0005 /* Set the physical paths to the Word Documents and to the Word
0006    document which are stored as Macro variables */
0007 path=symget('path');
0008 wordpath=symget('wordpath');
0009
0010 return;
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250
0251
0252
0253
0254
0255
0256
0257
0258
0259
0260
0261
0262
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347
0348
0349
0350
0351
0352
0353
0354
0355
0356
0357
0358
0359
0360
0361
0362
0363
0364
0365
0366
0367
0368
0369
0370
0371
0372
0373
0374
0375
0376
0377
0378
0379
0380
0381
0382
0383
0384
0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
0535
0536
0537
0538
0539
0540
0541
0542
0543
0544
0545
0546
0547
0548
0549
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0821
0822
0823
0824
0825
0826
0827
0828
0829
0830
0831
0832
0833
0834
0835
0836
0837
0838
0839
0840
0841
0842
0843
0844
0845
0846
0847
0848
0849
0850
0851
0852
0853
0854
0855
0856
0857
0858
0859
0860
0861
0862
0863
0864
0865
0866
0867
0868
0869
0870
0871
0872
0873
0874
0875
0876
0877
0878
0879
0880
0881
0882
0883
0884
0885
0886
0887
0888
0889
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940
0941
0942
0943
0944
0945
0946
0947
0948
0949
0950
0951
0952
0953
0954
0955
0956
0957
0958
0959
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0970
0971
0972
0973
0974
0975
0976
0977
0978
0979
0980
0981
0982
0983
0984
0985
0986
0987
0988
0989
0990
0991
0992
0993
0994
0995
0996
0997
0998
0999
1000

```

Workshop Session

Create an Excel Spreadsheet and enter the text 'Demographic Data' in cell A1. Save the spreadsheet and make a note of the path and file name used.

Write a SAS program to do the following:

1. Invoke Excel.
2. Open the spreadsheet file that you have created for this exercise.
3. Write out values of Demographic Data Set variables to the Excel spreadsheet.
4. Save the spreadsheet.
5. Close the spreadsheet file.
6. Close Excel.

Demographic Data									
Gene Cherry	22	M	84	80	S	0	1	13882	low
Julia Panderbury	28	F	67	74	M	1	1	8870	low
Norman Henry	100	M	55	82	D	2	1	12572	low
Harold Hicks	28	M	64	74	M	3	1	23780	high
Mary Messersmith	23	F	64	68	P	1	1	80512	low
Bob Beekington	30	M	87	76	S	0	1	7520	low
Pete Sanford	34	F	78	72	M	2	1	28512	high
Julia Todd	39	F	66	66	M	2	1	14840	low
Helen Corderford	56	F	52	80	M	3	2	47620	high
Mary Chapel	52	F	66	69	M	5	1	23460	high
Lemuel Deereest	60	F	60	71	M	3	2	20882	high
Luke Jennings	66	F	61	61	M	0	2	13760	low
Shirley Walters	66	F	63	76	W	3	1	47520	high
Chane Duller	66	F	67	81	M	2	1	13840	low
Dean Dunt	23	F	76	80	S	0	1	12840	low

The resulting spreadsheet should appear as follows:

Hints:

Use a separate Fileref to define the range of cells into which people's names are written and include the NOTAB option i.e.:

```
FILENAME                XLF1                DDE
'EXCEL|C:\COURSES\DATA\DEMOG.XLS!r3c1:
r52c1' NOTAB;
```

The NOTAB option makes SAS ignore Tab Characters between variables. In this example, the space between the first name and surname is taken as a Tab Character, which results in the two parts of the name being put into different columns in Excel unless NOTAB is specified. Use a second Fileref to define the range of cells for the remaining variables but do not include the NOTAB option as non of the variables include space characters:

```
FILENAME                XLF2                DDE
'EXCEL|C:\COURSES\DATA\DEMOG.XLS!r3c1:r52c1';
```

Within a null Data Step, use separate File and Put statements of the following form to output the data:

```
DATA _NULL_;
  FILE XLF1;
  PUT NAME;
  FILE XLF2;
  PUT AGE GENDER etc;
RUN;
```

Remember to reset options NOXWAIT and NOXSYNC to XWAIT and XSYNC respectively.