Moving Data and Results Between SAS® and Excel

Harry Droogendyk
Stratia Consulting Inc.
Introduction

- SAS can read (and write) anything
Introduction

- In the end… users want EVERYTHING in ….
Introduction

- SAS is a toolbox
- many options
  - what’s available ?
  - what’s best ?
  - what do I know how to use ?
- SAS $\leftarrow \rightarrow$ Excel
Introduction

- Text data
  - delimited (e.g. CSV, tab) and fixed width

- Native Excel formats
  - PROC IMPORT / EXPORT
  - Excel libname engine
  - Enterprise Guide
Introduction

- file types consumable by Excel
  - ODS
    - HTML, CSV, tagsets
    - EXCEL !!!!!

- Add-in for Microsoft Office

- Licensing requirements ?
  - see paper for summary
Text Data

- viewable in Notepad
  - delimited
    - CSV, tab-delimited
  - fixed width
data class;
    infile 'c:\temp\class.csv' delimiter = ' ,'
        missover dsd lrecl=32767 firstobs=2 ;

    informat Name $8. Sex $1. 
        Age Height Weight best32. ;

    input Name Sex Age Height Weight;
run;
PROC IMPORT

proc import datafile = 'c:\temp\class.csv'
  out = class
  dbms = csv replace;
getnames = yes;
run;
data WORK.CLASS;
%let _EFIERR_ = 0; /* set the ERROR detection macro variable */
infile 'c:temp\class.csv' delimiter = ',' MISSOVER DSD lrecl=32767 firstobs=2;
informat Name $8. ;
informat Sex $1. ;
informat Age best32. ;
informat Height best32. ;
informat Weight best32. ;
format Name $8. ;
format Sex $1. ;
format Age best12. ;
format Height best12. ;
format Weight best12. ;
input Name $ Sex $ Age Height Weight ;
if _ERROR_ then call symputx('_EFIERR_',1);
run;
filename fixed lrecl = 100 'c:\temp\class_fixed.txt';

data _null_;
  file fixed;
  set sashelp.class;
  put @1 name  @12 sex  @14 age  4.-L
       @19 height  5.1  @25 weight  5.1;
run;
Text Data – To Excel

- open .txt in Excel
- familiar import wizard
DDE – Dynamic Data Exchange

- Windows program communication - older than dirt
  - powerful
  - read / write data directly into cells
  - worksheet manipulation
  - Excel file commands
  - execute Excel macros
Native Excel Files – Not using EG

- requires SAS Access for PC Files license
- PROC IMPORT / EXPORT
  - many options
    - only some pertinent to Excel, confusing
- Excel libname engine
  - read Excel data like any other data
  - create NICE output
IMPORT / EXPORT - .xls(x)

- emphasis on functionality
  - limited formatting
  - use to move data in/out
- post-processing required
  - clean up imported data
  - link to formatted sheets for presentation
- SAS formats do not apply
data class;
   set sashelp.class;
   label sex  = 'Gender'
            name    = 'Given Name';
   format height weight 9.2;
run;
PROC EXPORT

proc export
  data = class
  outfile = 'c:\temp\class_label.xls'
  dbms = 'Excel97'
  sheet = 'SAS Paper'
  label replace;
run;
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Given Name</td>
<td>Gender</td>
<td>Age</td>
<td>Height</td>
</tr>
<tr>
<td>2</td>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69</td>
</tr>
<tr>
<td>3</td>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
</tr>
<tr>
<td>4</td>
<td>Barbara</td>
<td>F</td>
<td>13</td>
<td>65.3</td>
</tr>
<tr>
<td>5</td>
<td>Carol</td>
<td>F</td>
<td>14</td>
<td>62.8</td>
</tr>
<tr>
<td>6</td>
<td>Henry</td>
<td>M</td>
<td>14</td>
<td>63.5</td>
</tr>
<tr>
<td>7</td>
<td>James</td>
<td>M</td>
<td>12</td>
<td>57.3</td>
</tr>
<tr>
<td>8</td>
<td>Jane</td>
<td>F</td>
<td>12</td>
<td>59.8</td>
</tr>
</tbody>
</table>

SAS_Paper
PROC EXPORT - options

- some parameters are required
  - DATA, OUTFILE

- optional, ahhh.... options
  - DBMS
  - SHEET – dataset name
  - DBDSOPTS
    » cannot always use dataset options directly
  - REPLACE
PROC EXPORT - wizard
PROC EXPORT - wizard
The export Wizard can create a file containing PROC EXPORT statements that can be used in SAS programs to export this data again.

If you want these statements to be generated, enter the filename where they should be saved:

c:\temp\class_wizard.sas

```
1   PROC EXPORT DATA= WORK.CLASS
2       OUTFILE= "c:\temp\class_wiz.xlsx"
3       DBMS=EXCEL REPLACE;
4       SHEET="class_wizard";
5   RUN;
6```

PROC IMPORT

- more options
  - data types, lengths must be determined
  - Excel columns are not “typed”
    » numeric values, “N/A”
  - GUESSINGROWS, MIXED
- some options DBMS dependent
- post-processing required?
### Given Names and Test Results

<table>
<thead>
<tr>
<th>Name</th>
<th>Surname</th>
<th>Math</th>
<th>History</th>
<th>English</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>George</td>
<td>Bell</td>
<td>67</td>
<td>72</td>
<td>74</td>
<td>Much improved</td>
</tr>
<tr>
<td>Henry</td>
<td>Manian</td>
<td>91</td>
<td>82</td>
<td>81</td>
<td>Excellent marks</td>
</tr>
<tr>
<td>John</td>
<td>Herbert</td>
<td>48</td>
<td>62</td>
<td>57</td>
<td>Better effort required</td>
</tr>
<tr>
<td>Chantal</td>
<td>Leveque</td>
<td>77</td>
<td>N/A</td>
<td>84</td>
<td>Good work</td>
</tr>
<tr>
<td>Kurt</td>
<td>Cobain</td>
<td>55</td>
<td>34</td>
<td>23</td>
<td>Much more expected</td>
</tr>
</tbody>
</table>

**Note:**

- George Bell: Much improved
- Henry Manian: Excellent marks
- John Herbert: Better effort required
- Chantal Leveque: Good work
- Kurt Cobain: Much more expected
PROC IMPORT

proc import
  datafile = 'c:\temp\import_2003.xls'
  out = school_xls
  dbms = 'xls'    replace;
  sheet = 'School';
  getnames = yes;
  guessingrows = 20;
run;
NOTE: Variable Name Change.
   Given Name -> Given_Name
NOTE: Variable Name Change.
   **Note** -> VAR6
NOTE: The import data set has 5 observations and 6 variables.
NOTE: WORK.SCHOOL_XLS data set was successfully created.
## PROC IMPORT

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Type</th>
<th>Length</th>
<th>Format</th>
<th>Informat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given_Name</td>
<td>Text</td>
<td>12</td>
<td>$12.</td>
<td>$12.</td>
</tr>
<tr>
<td>Surname</td>
<td>Text</td>
<td>10</td>
<td>$10.</td>
<td>$10.</td>
</tr>
<tr>
<td>Math</td>
<td>Number</td>
<td>8</td>
<td>BEST12.</td>
<td></td>
</tr>
<tr>
<td>English</td>
<td>Number</td>
<td>8</td>
<td>BEST12.</td>
<td></td>
</tr>
<tr>
<td>VAR6</td>
<td>Text</td>
<td>35</td>
<td>$35.</td>
<td>$35.</td>
</tr>
</tbody>
</table>
The data points below were captured during an extensive tour in the spring of 2012.

<table>
<thead>
<tr>
<th>Given Name</th>
<th>Surname</th>
<th>0-50</th>
<th>51-100</th>
<th>Score</th>
<th><strong>Note</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Rob</td>
<td>Bell</td>
<td>Y</td>
<td></td>
<td>48</td>
<td>Erroneous exegsis</td>
</tr>
<tr>
<td>John</td>
<td>Piper</td>
<td></td>
<td>87</td>
<td>87</td>
<td>Well developed theme</td>
</tr>
<tr>
<td>Eric</td>
<td>Alexander</td>
<td>Y</td>
<td></td>
<td>94</td>
<td>Outstanding</td>
</tr>
<tr>
<td>John</td>
<td>MacArthur</td>
<td>Y</td>
<td></td>
<td>85</td>
<td>Eschatological</td>
</tr>
<tr>
<td>Benjamin</td>
<td>Henn</td>
<td>13</td>
<td></td>
<td>13</td>
<td>Complete mess</td>
</tr>
</tbody>
</table>

**2012**

**School**
DATA;
    datafile = 'c:\temp\import.xlsx';
    out = import;
    dbms = excel2007 replace;
    sheet = '2012';
    getnames = yes;
    DBDSOPTS = 'firstobs=3';
RUN;
PROC IMPORT

- **DBMS value differences**
  - options not available with all DBMS values
  - log messages differ

- **NOTE:** WORK.IMPORT data set was successfully created.

- `DBDSOPTS='firstobs=3'`
- non-standard column names
The data points below were captured during an extensive tour in the spring of 2012.

<table>
<thead>
<tr>
<th>Given Name</th>
<th>Surname</th>
<th>0-50</th>
<th>51-100</th>
<th>Score</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rob</td>
<td>Bell</td>
<td>Y</td>
<td></td>
<td>48</td>
<td>Erroneous exegesis</td>
</tr>
<tr>
<td>John</td>
<td>Piper</td>
<td></td>
<td>87</td>
<td>87</td>
<td>Well developed theme</td>
</tr>
<tr>
<td>Eric</td>
<td>Alexander</td>
<td>Y</td>
<td></td>
<td>94</td>
<td>Outstanding</td>
</tr>
<tr>
<td>John</td>
<td>MacArthur</td>
<td></td>
<td></td>
<td>85</td>
<td>Eschatological</td>
</tr>
<tr>
<td>Benjamin</td>
<td>Henn</td>
<td>13</td>
<td></td>
<td>13</td>
<td>Complete mess</td>
</tr>
</tbody>
</table>

**Note:**
### PROC IMPORT

<table>
<thead>
<tr>
<th>Given_Name</th>
<th>Surname</th>
<th>Score</th>
<th><strong>Note</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Rob</td>
<td>Bell</td>
<td>48</td>
<td>Erroneous exegesis</td>
</tr>
<tr>
<td>John</td>
<td>Piper</td>
<td>87</td>
<td>Well developed theme</td>
</tr>
<tr>
<td>Eric</td>
<td>Alexander</td>
<td>94</td>
<td>Outstanding</td>
</tr>
<tr>
<td>John</td>
<td>MacArthur</td>
<td>85</td>
<td>Eschatological</td>
</tr>
<tr>
<td>Benjamin</td>
<td>Henn</td>
<td>13</td>
<td>Complete mess</td>
</tr>
</tbody>
</table>

- **column names**
  - 0 - replaced with __
  - 51 replaced with _1
  - _1_100 is character
  - __Note__ -- different than VAR6
PROC IMPORT - options

- DBMS
- GETNAMES – defaults to YES
- DBDSOPTS – not valid with DBMS='XLS'
- GUESSINGROWS – only valid with DBMS='XLS'
- see paper and SAS Online Docs for more
PROC IMPORT - wizard

Import Wizard - Select table

What table do you want to import?

class_wizard

Options...

SAS Import: Spreadsheet Options

- Use data in the first row as SAS variable names.
- Convert numeric values to characters in a mixed types column.
- Use the largest text size in a column as SAS variable length.
- Use DATE. format for a Date/Time column.
- Use TIME. format if only time values found in a column.

The largest text size allowed in a column: 1024
PROC IMPORT - wizard

- customize code or include in batch process

```sas
PROC IMPORT OUT= WORK.CLASS_WIZARD_IMPORT
DATAFILE= "c:\temp\class_wiz.xlsx"
DBMS=EXCEL REPLACE;
RANGE="class_wizard";
GETNAMES=YES;
MIXED=NO;
SCANTEXT=YES;
USEDATE=YES;
SCANTIME=YES;
RUN;
```

Excel LIBNAME Engine

- SAS Access to PC Files required
- treats Excel worksheets / ranges like any other dataset
- use with DATA or PROC steps
- familiar syntax

```sas
libname xls <engine> physical-file-name;
libname xls excel 'c:\temp\report.xls';
... SAS code ...
libname xls clear; *very important !! ;
```
Excel LIBNAME Engine

- SAS will create Excel files
- libref available in SAS Explorer window
- caveats
  - Excel row and column limits apply
  - Excel sheet / column names can contain non-standard SAS characters
    » use validvarname = ANY

```
data xls.'sheet1'$'n';  * worksheet;
data xls.range_out;  * named range range_out;
```
data xls.class;
  set sashelp.class;
run;

proc copy in = sashelp
  out = xls;
  select prdsale shoes;
run;
Excel LIBNAME Engine – writing to Excel

<table>
<thead>
<tr>
<th>Dept</th>
<th>No. Employees</th>
<th>Total Salaries</th>
<th>Minimum Salary</th>
<th>Maximum Salary</th>
<th>Average Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>$578,739.58</td>
<td>$66,650.17</td>
<td>$117,091.21</td>
<td>$96,457</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>$384,170.95</td>
<td>$61,163.06</td>
<td>$87,065.84</td>
<td>$76,834</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>$343,696.27</td>
<td>$75,471.54</td>
<td>$97,062.81</td>
<td>$85,924</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>$406,973.24</td>
<td>$84,081.02</td>
<td>$115,985.15</td>
<td>$101,743</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>$614,940.33</td>
<td>$78,630.07</td>
<td>$116,899.44</td>
<td>$102,490</td>
</tr>
</tbody>
</table>

**Departmental Salaries**

2008-05-03
Excel LIBNAME Engine – reading from Excel

<table>
<thead>
<tr>
<th>Given Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69</td>
<td>112.5</td>
</tr>
<tr>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
<td>84</td>
</tr>
<tr>
<td>Barbara</td>
<td>F</td>
<td>13</td>
<td>65.3</td>
<td>98</td>
</tr>
<tr>
<td>Carol</td>
<td>F</td>
<td>14</td>
<td>62.8</td>
<td>102.5</td>
</tr>
<tr>
<td>Henry</td>
<td>M</td>
<td>14</td>
<td>63.5</td>
<td>102.5</td>
</tr>
<tr>
<td>James</td>
<td>M</td>
<td>12</td>
<td>57.3</td>
<td>83</td>
</tr>
<tr>
<td>Jane</td>
<td>F</td>
<td>12</td>
<td>59.8</td>
<td>84.5</td>
</tr>
</tbody>
</table>
Excel LIBNAME Engine – reading from Excel

- analogous to previous examples

```sas
libname xls excel 'c:\temp\class.xlsx';
data my_class;
  set xls.'class$n';
run;
proc print data = xls.'class$n'
  label noobs;
run;
```
<table>
<thead>
<tr>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69.0</td>
<td>112.5</td>
</tr>
<tr>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
<td>84.0</td>
</tr>
<tr>
<td>Barbara</td>
<td>F</td>
<td>13</td>
<td>65.3</td>
<td>98.0</td>
</tr>
<tr>
<td>#</td>
<td>Variable</td>
<td>Format</td>
<td>Label</td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>--------------</td>
<td>--------</td>
<td>------------------</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Age</td>
<td></td>
<td>Age</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Given_Name</td>
<td>$7.</td>
<td>Given Name</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Height</td>
<td></td>
<td>Height</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sex</td>
<td>$1.</td>
<td>Sex</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Weight</td>
<td></td>
<td>Weight</td>
<td></td>
</tr>
</tbody>
</table>
Excel LIBNAME Engine - abilities

- can create
  - workbooks, sheets, named ranges
- delete data within existing named range
- populate *empty*, named range
- append data to named range
- read data from sheet or named range
Excel LIBNAME Engine - limitations

- cannot
  - formatting, font, color
  - delete entire workbook, sheet or named range
  - delete cells containing a formula
  - write Excel formulas
Excel LIBNAME Engine - options

- **DSLABEL**: SAS labels become Excel column headers
- **DBSASTYPE**: define column type and length
- **VERSION**: only necessary when creating Excel files
- **SCAN_TEXT**: set to ‘NO’ to append data
does not require SAS ACCESS for PC Files license
  - if present, can use it

uses “local Windows components”
  - reads the spreadsheet
  - transforms to text format
  - reads text into dataset

cannot create standalone batch process
SAS Enterprise Guide – import from Excel
SAS Enterprise Guide – import from Excel
SAS Enterprise Guide – import from Excel

Select columns and define attributes:

- **PERIOD_DT**
  - Type: Date
  - Source Informat: DATE9
  - Output Informat: DATE9
  - Length: 8

- **BAN**
  - Type: Number
  - Source Informat: BEST12
  - Output Informat: BEST12
  - Length: 8

- **SUBSCRIBER_ID**
  - Type: String
  - Source Informat: $S
  - Output Informat: $S

- **MIGR_CNT**
  - Type: Number
  - Source Informat: BEST12
  - Output Informat: BEST12
  - Length: 8

- **REASON_CD**
  - Type: String
  - Source Informat: $S
  - Output Informat: $S

- **ORIG_DEALER_CD**
  - Type: String
  - Source Informat: $S
  - Output Informat: $S

- **DEALER_CD**
  - Type: String
  - Source Informat: $S
  - Output Informat: $S

Advanced Options:

- Embed the data within the generated SAS code.
- Import the data using SAS/ACCESS Interface to PC Files whenever possible.
- Remove characters that can cause transmission errors from text-based data files.
SAS Enterprise Guide – import from Excel
SAS Enterprise Guide – export to Excel

- multiple methods
- File menu
SAS Enterprise Guide – export to Excel

- while viewing data
SAS Enterprise Guide – export to Excel

- right-click on data on palette
Output Delivery System (ODS)

- SAS output can be routed to various “destinations”
  - PDF, RTF, HTML, tagsets etc
  - lots of flexibility and nice results
  - PROC TEMPLATE

- ODS destinations consumable by Excel
  - CSV, HTML
  - tagsets
    » ExcelXP, MSOffice2K
  - EXCEL !!!!!
    » can also be IMPORTED by SAS
Output Delivery System (ODS)

ods csv file = 'c:\temp\ods_csv.csv';

ods html file = 'c:\temp\ods_html.xlsm'
  style=seaside;

ods tagsets.excelxp file =
  'c:\temp\tagsets_excelxp.xls'
  options ( autofilter='all' ) style=ocean;

ods excel file = 'c:\temp\excel.xlsx'
  options ( autofilter='all' ) style=ocean;
proc print data = sashelp.class
  noobs label;
run;

ods csv close;
ods html close;
ods tagsets.excelxp close;
ods excel close;
Output Delivery System (ODS)

- CSV

<table>
<thead>
<tr>
<th></th>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69</td>
<td>112.5</td>
</tr>
<tr>
<td>3</td>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
<td>84</td>
</tr>
<tr>
<td>4</td>
<td>Barbara</td>
<td>F</td>
<td>13</td>
<td>65.3</td>
<td>98</td>
</tr>
<tr>
<td>5</td>
<td>Carol</td>
<td>F</td>
<td>14</td>
<td>62.8</td>
<td>102.5</td>
</tr>
<tr>
<td>6</td>
<td>Henry</td>
<td>M</td>
<td>14</td>
<td>63.5</td>
<td>102.5</td>
</tr>
</tbody>
</table>
Output Delivery System (ODS)

- HTML (.xls suffix)

<table>
<thead>
<tr>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69</td>
<td>112.5</td>
</tr>
<tr>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
<td>84</td>
</tr>
</tbody>
</table>
Output Delivery System (ODS)

- tagsets.ExcelXP
**Output Delivery System ( ODS )**

- **EXCEL**

![Excel screenshot](image_url)
Output Delivery System - tagsets.ExcelXP

- traffic lighting
- specify viewing / printing options
  - frozen headers, filters, fit to page, gridlines, multi-sheet
- format Excel columns via tagattr
  - "\#,\#,\#0.000" thousands with 3 dec
  - maintain leading zeroes
- Excel formulas
- Vince delGobbo / Eric Gebhart papers
Output Delivery System - EXCEL

- many of the same options as tagsets.ExcelXP
  - not all
- much smaller files
- no warning dialog box to disturb user peace of mind

Chevell Parker
Add-in for Microsoft Office ( AMO )

- SAS Business Intelligence license required
- data sources defined in metadata
  - remote RDBMS
  - SAS datasets
  - text files
- menu bar option - “SAS”
  - open data – detail or pivot
  - run Stored Processes
  - full set of SAS tasks
Add-in for Microsoft Office ( AMO )

- Finally... Excel can do analytics!
Add-in for Microsoft Office (AMO)

- browse to data source
- local or metadata resources
- Filter & Sort
Add-in for Microsoft Office (AMO)

- pivot table wizard
Add-in for Microsoft Office (AMO) - update

- open detail data

![Excel spreadsheet with data](image-url)
Add-in for Microsoft Office (AMO) - update

- End Edit / Commit
Wrap

- SAS ↔ Excel  MANY avenues
- degrees of flexibility and utility
- various license requirements
  - Base SAS
  - SAS Access PC Files
  - SAS Business Intelligence
- see paper for summary
- links in the paper
Contact

Harry Droogendyk

harry@stratia.ca

www.stratia.ca/papers