Paper 238-30

# SAS Excels!

Gerald Frey, Systems Seminar Consultants, Madison, WI

## ABSTRACT

This paper will introduce several techniques for supplying data from SAS® to MS Excel®. SAS® can produce detailed, summarized, and categorized data or reports. Frequently; end users desire this information in an Excel spreadsheet. Fortunately, using SAS we can provide Excel information from various platforms using several different techniques:

This paper will examine the following techniques:

- Export Wizard
- New SAS Explorer options in SAS 9®
- Custom DATA step
- SSCFLAT macro
- Windows ODBC connection
- New in SAS 9®, MS Excel libname engine
- ODS

By using these or other techniques, the SAS programmer is able to take advantage of SAS for creating the reports while still meeting the end users' requirement.

## WHY SEND DATA OR REPORTS TO EXCEL?

Many clients are used to manipulating Excel spreadsheets. Also, by distributing the data in Excel it can be viewed by many people in your organization, not just SAS users. Thus, we can deliver business information to the people who need it in a common, widely used, easily distributed format.

## AN EXAMPLE USING SAS FILE EXPORT FACILITY

The SAS System for PCs has an easy, wizard driven export facility. A non-technical, almost automatic, menu driven process allows you to select the SAS data set to export, the type of file to create, and the destination.

This method is not available on all systems (Most notably not in batch jobs) and for other than simple file types (like CSV or tab delimited) we need SAS/Access to PC file Formats to be licensed. The wizard follows the same process for any file type:

**File Export Facility Wizard Example Steps**

- Select File → Export Data
- Identify the SAS library and data set you wish to export.
- Click Next
- Identify the type of file you wish to CREATE. (CSV is among the included file types. SAS/ACCESS to PC File Formats must be licensed to use options like Microsoft Excel97, 2000 or 2002 workbook)
- Click Next
- Identify the Microsoft Excel Workbook you wish to create in the Connect to MS Excel box. (type in the file path and name or use the Browse button)
- Click OK
- Enter the sheet name you wish to use
- Click Next
- Type in the name of the file you wish to save the PROC EXPORT code in, if desired.
- Click Finish

The workbook and spreadsheet name you entered will be created (note, the name is in the log), and the PROC EXPORT code is saved if you specified a file name.

## NEW IN SAS 9: USE THE SAS EXPLORER WINDOWING ENVIRONMENT

Starting with SAS 9.1 the SAS Explorer window now has the option to view SAS data set in Excel.

**SAS Explore Window View in ExcelExample Steps**

- In the SAS Explorer Window, Double click to select a Library.
- Right Click on a SAS data set to bring up a menu list.
- Click on View in Excel.
- The selected SAS data set will be sent to a file and Excel will open displaying that file.
- View the spreadsheet.

## USING THE DATA STEP TO CREATE A CUSTOM CSV

The DATA step is very powerful and flexible. The DATA step is available in all environments, and provides the ability to custom create a file from any data set. One disadvantage is that the name of each field in the data set must be known, and each data set needs a different, and possibly difficult, program. A sample program is shown below:

```
data _null_;                                   /* no sas ds needed */
 set softsale;                                 /* read input ds    */
 file ’c:\temp\sugi samples\softsale.csv’;     /* output flat file */
 if _n_ =1 then                                /* before first rec */
   put   ’"Name","Division",
          "Years","Sales",
          "Expense","State"’;
 put ’"’ name     +(-1) ’","’                  /* put out values   */
     ’"’ division +(-1) ’","’                  /* quoted and commas*/
     ’"’ years    +(-1) ’","’                  /* between          */
     ’"’ sales    +(-1) ’","’
     ’"’ expense  +(-1) ’","’
     ’"’ state    +(-1) ’"’ ;
run;                                           /* end of step      */
```

Results in:

```
"Name","Division","Years","Sales","Expense","State"
"CHRIS","H","2","233.11","94.12","WI"
"MARK","H","5","298.12","52.65","WI"
"SARAH","S","6","301.21","65.17","MN"
"PAT","H","4","4009.21","322.12","IL"
"JOHN","H","7","678.43","150.11","WI"
"WILLIAM","H","11","3231.75","644.55","MN"
"ANDREW","S","24","1762.11","476.13","MN"
"BENJAMIN","S","3","201.11","25.21","IL"
"JANET","S","1","98.11","125.32","WI"
"STEVE","H","21","6153.32","1507.12","WI"
"JENNIFER","S","1","542.11","134.24","IL"
"JOY","S","12","2442.22","761.98","WI"
"MARY","S","14","5691.78","2452.11","WI"
"TOM","S","5","5669.12","798.15","MN"
"BETH","H","12","4822.12","982.1","WI"
```

Another example using a file option called *DSD* or Data Set Delimited will automatically insert commas in the resulting file. We still need to know the names of the fields to output:

```
data _null_;                                   /* no sas ds needed */
 set softsale;                                 /* read input ds    */
 file ’c:\temp\sugi samples\softsale.csv’
       dsd;                                    /* data set delimited*/
 put name
     division
     years
     sales
     expense
     state ;
run;                                           /* end of step      */
```

Results in:

```
 CHRIS,H,2,233.11,94.12,WI
MARK,H,5,298.12,52.65,WI
SARAH,S,6,301.21,65.17,MN
PAT,H,4,4009.21,322.12,IL
JOHN,H,7,678.43,150.11,WI
```

```
WILLIAM,H,11,3231.75,644.55,MN
ANDREW,S,24,1762.11,476.13,MN
BENJAMIN,S,3,201.11,25.21,IL
JANET,S,1,98.11,125.32,WI
STEVE,H,21,6153.32,1507.12,WI
JENNIFER,S,1,542.11,134.24,IL
JOY,S,12,2442.22,761.98,WI
MARY,S,14,5691.78,2452.11,WI
TOM,S,5,5669.12,798.15,MN
BETH,H,12,4822.12,982.1,WI
```

## A UTILITY MACRO

Systems Seminar Consultants has written a generic SAS Macro called %*SSCFLAT*. This macro will run in any environment and can convert a SAS data set to a CSV file. One advantage to this macro is we do not need to know the field names in a data set. This macro will honor most formats by default and can provide a header row with the variables names or variable labels. The macro needs to be defined to the system first, then can be called as shown:

```
%include ’c:\temp\sscflat.sas’;      /* location of macro code */

%sscflat(msasds=softsale,mflatout=c:\temp\sugi samples\softsale.csv,
         mlabel=YES)
```

Results in:

```
"Name","Division","Years","Sales","Expense","State"
"CHRIS","H",2,233.11,94.12,"WI"
"MARK","H",5,298.12,52.65,"WI"
"SARAH","S",6,301.21,65.17,"MN"
"PAT","H",4,4009.21,322.12,"IL"
"JOHN","H",7,678.43,150.11,"WI"
"WILLIAM","H",11,3231.75,644.55,"MN"
"ANDREW","S",24,1762.11,476.13,"MN"
"BENJAMIN","S",3,201.11,25.21,"IL"
"JANET","S",1,98.11,125.32,"WI"
"STEVE","H",21,6153.32,1507.12,"WI"
"JENNIFER","S",1,542.11,134.24,"IL"
"JOY","S",12,2442.22,761.98,"WI"
"MARY","S",14,5691.78,2452.11,"WI"
"TOM","S",5,5669.12,798.15,"MN"
"BETH","H",12,4822.12,982.1,"WI"
```

Contact Systems Seminar Consultants for a copy of the *%SSCFLAT* code.

## NEW IN SAS 9: USE THE SAS MACRO %DB2CSV

A new utility macro in SAS 9 is called %DB2CSV. It converts SAS data sets to comma-separated value (CSV) files. There are several arguments that can be specified when calling the macro, but to call from a program, or the program editor window, there are three arguments we will briefly look at:

data= Name of the input data set.
runmode=(S,B) Type of run, runmode = S is the default, and is only for Application Dispatcher or Stored Process servers.
Runmode = B is for when you submit from the SAS Program Editor or you call it from a SAS program.
csvfile= Name of CSV file to create.

An example call would look like this:

```
%ds2csv(data=sasuser.softsale,
   runmode=b,
   csvfile=c:\temp\sugi samples\ds2csv.csv)
```

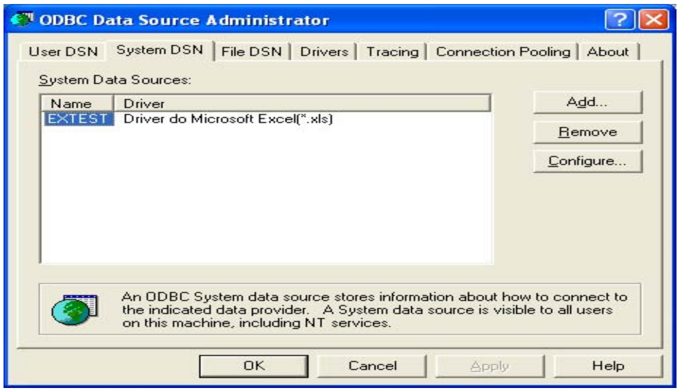
Results in:

```
"Name","Division","Years","Sales","Expense","State"
"CHRIS","H","2","233.11","94.12","WI"
```

```
"MARK","H","5","298.12","52.65","WI"
"SARAH","S","6","301.21","65.17","MN"
"PAT","H","4","4009.21","322.12","IL"
"JOHN","H","7","678.43","150.11","WI"
"WILLIAM","H","11","3231.75","644.55","MN"
"ANDREW","S","24","1762.11","476.13","MN"
"BENJAMIN","S","3","201.11","25.21","IL"
"JANET","S","1","98.11","125.32","WI"
"STEVE","H","21","6153.32","1507.12","WI"
"JENNIFER","S","1","542.11","134.24","IL"
"JOY","S","12","2442.22","761.98","WI"
"MARY","S","14","5691.78","2452.11","WI"
"TOM","S","5","5669.12","798.15","MN"
"BETH","H","12","4822.12","982.1","WI"
```

## USING ODBC IN WINDOWS

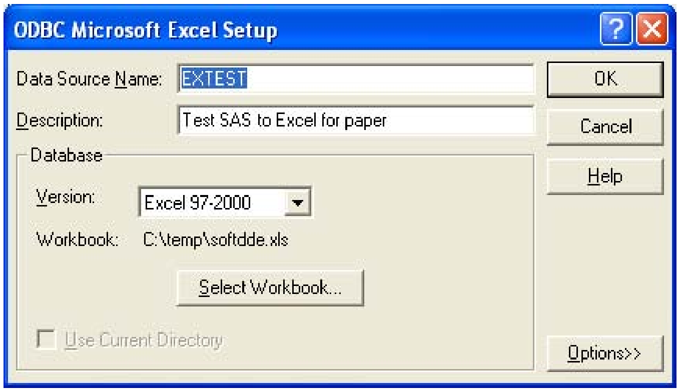
Windows has a method for managing files called Open Database Connectivity (ODBC). A simple description of ODBC would be using Windows to manage the filenames. This has an advantage that many types of files can be referenced, not just MS Excel. The connection can be a local or a global reference. However, this method is another layer to run through, and it may be slow for large file processing. To use an ODBC connection, the ODBC administrator must be run (a separate product, not the SAS system for PCs) which will establish a connection to a data source. An example of using the ODBC Administrator to connect to a spreadsheet:



System Data Sources:

| Name | Driver |
|---|---|
| EXTEST | Driver do Microsoft Excel(*.xls) |

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.



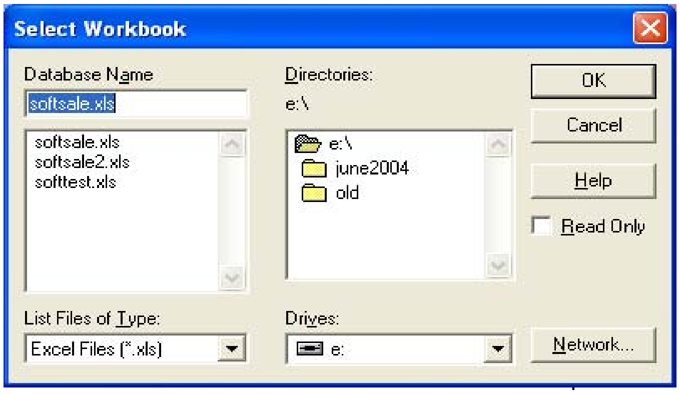
Data Source Name: EXTEST

Description: Test SAS to Excel for paper

Version: Excel 97-2000

Workbook: C:\temp\softdde.xls

□ Use Current Directory



Database Name: softsale.xls

softsale.xls
softsale2.xls
softtest.xls

Directories: e:\

e:\
june2004
old

□ Read Only

List Files of Type: Excel Files (*.xls)

Drives: e:

Then to use the ODBC connection in SAS code, we use the ODBC libname engine.

```
libname myodbc odbc dsn=extest;
Proc contents data=myodbc.softsale;
run;

proc print data=myodbc.softsale;
run;
```

Results in:

Proc Contents output:

```
                           The CONTENTS Procedure

                  Data Set Name        MYODBC.softsale    Observations          .
                  Member Type          DATA               Variables             6
                  Engine               ODBC               Indexes               0
                  Created              .                  Observation Length    0
                  Last Modified        .                  Deleted Observations  0
                  Protection                              Compressed            NO
                  Data Set Type                           Sorted                NO
                  Label
                  Data Representation  Default
                  Encoding             Default


                             Alphabetic List of Variables and Attributes

                   #    Variable    Type    Len    Format    Informat    Label

                   2    Division    Char    255    $255.     $255.       Division
                   5    Expense     Num       8                          Expense
                   1    Name        Char    255    $255.     $255.       Name
                   4    Sales       Num       8                          Sales
                   6    State       Char    255    $255.     $255.       State
                   3    Years       Num       8                          Years
```

Proc Print Output:

```
Obs  Name

       1  CHRIS
       2  MARK
       3  SARAH
       4  PAT
       5  JOHN
       6  WILLIAM
       7  ANDREW
       8  BENJAMIN
       9  JANET
      10  STEVE

     Obs  Division

       1  H
       2  H
       3  S
       4  H
       5  H
       6  H
       7  S
       8  S
       9  S
      10  H

     Obs  Years      Sales      Expense

       1     2      233.11       94.12
```

```
        2    5      298.12      52.65
        3    6      301.21      65.17
        4    4     4009.21     322.12
        5    7      678.43     150.11
        6   11     3231.75     644.55
        7   24     1762.11     476.13
        8    3      201.11      25.21
        9    1       98.11     125.32
       10   21     6153.32    1507.12
```

## NEW IN SAS 9: USE THE EXCEL LIBNAME ENGINE

Starting with SAS 9 SAS/ACCESS, you can use the Excel libname engine. This gives us direct access to Excel workbooks with the ability to address multiple sheets. We can store formatted values, but SAS formats, lengths and labels are not stored.  This method of access allows reading or creating a worksheet, but does not REPLACE an existing worksheet. An example that reads an Excel spreadsheet called softsale and filters just the division ‘H’ records to a new worksheet called harddiv in the same Excel workbook follows:

```
libname myExcelExcel’c:\temp\sugi samples\softsale.xls’;

data myexcel.harddiv;             /* new Excelsheet name  */
  set myexcel.softsale;           /* existing worksheet   */
  where upcase(division)=’H’;     /* filter input         */
run;                              /* end of step          */
proc print data=myexcel.harddiv;  /* print new data set   */
 title ’test print from excel’;   /* title of report      */
run;                              /* end of step          */
```

Results in:

```
                  test print from excel

Obs     Name       Division    Years     Sales     Expense    State

1    CHRIS          H            2      233.11      94.12     WI
2    MARK           H            5      298.12      52.65     WI
3    PAT            H            4     4009.21     322.12     IL
4    JOHN           H            7      678.43     150.11     WI
5    WILLIAM        H           11     3231.75     644.55     MN
6    STEVE          H           21     6153.32    1507.12     WI
7    BETH           H           12     4822.12     982.10     WI
```

## CAPTURE THE REPORT USING ODS

The SAS Output Delivery System (ODS) is a very easy method to capture reports and widely distribute them. The ODS system is available on all platforms, both batch and interactively. ODS can capture the *report* and *Excel* can display the *data* on the report with formatting. One disadvantage is that file sizes can get very large if care is not taken with formatting the reports. There are 3 general ways to use ODS to send a report to EXCEL.

Create a CSV file using the ODS CSV, or ODS CSVALL destination.
Create a HTML file using the ODS HTML destinations.
Create a XML file using the ODS MARKUP destination (Excel2002 and newer can open).

Once we have created a report in a HTML file, then we can use Excel file open to view our report in Excel. The report is not saved as an Excel workbook until it is saved in Excel.

Using the ODS CSV destination results in a CSV file similar to the DATA step examples above:

```
ods listing close;
ods csv body=’c:\temp\sugi samples\samplecsv.csv’;
proc print data=sasuser.softsale noobs;
  title ’ csv only’;
run;
ods csv close;ods listing;
```

Results in:

```
"Name","Division","Years","Sales","Expense","State"
"CHRIS","H",2,233.11,94.12,"WI"
"MARK","H",5,298.12,52.65,"WI"
"SARAH","S",6,301.21,65.17,"MN"
"PAT","H",4,4009.21,322.12,"IL"
"JOHN","H",7,678.43,150.11,"WI"
"WILLIAM","H",11,3231.75,644.55,"MN"
"ANDREW","S",24,1762.11,476.13,"MN"
"BENJAMIN","S",3,201.11,25.21,"IL"
"JANET","S",1,98.11,125.32,"WI"
"STEVE","H",21,6153.32,1507.12,"WI"
"JENNIFER","S",1,542.11,134.24,"IL"
"JOY","S",12,2442.22,761.98,"WI"
"MARY","S",14,5691.78,2452.11,"WI"
"TOM","S",5,5669.12,798.15,"MN"
"BETH","H",12,4822.12,982.10,"WI"
```

Using the ODS CSVALL destination results in a CSV file similar to the DATA step examples above, but we have captured the report, so titles and footnotes are also captured:

```
ods listing close;
ods csvall file="c:\temp\sugi samples\samplecsvall.csv";
proc print data=sasuser.softsale;
  title ’ csvall includes title’;
run;
ods csvall close;
ods listing;
```

Results in:

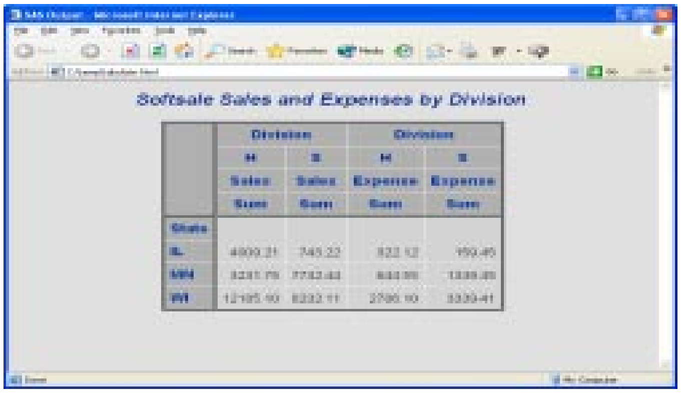
```
csvall includes title

"Name","Division","Years","Sales","Expense","State"
"CHRIS","H",2,233.11,94.12,"WI"
"MARK","H",5,298.12,52.65,"WI"
"SARAH","S",6,301.21,65.17,"MN"
"PAT","H",4,4009.21,322.12,"IL"
"JOHN","H",7,678.43,150.11,"WI"
"WILLIAM","H",11,3231.75,644.55,"MN"
"ANDREW","S",24,1762.11,476.13,"MN"
"BENJAMIN","S",3,201.11,25.21,"IL"
"JANET","S",1,98.11,125.32,"WI"
"STEVE","H",21,6153.32,1507.12,"WI"
"JENNIFER","S",1,542.11,134.24,"IL"
"JOY","S",12,2442.22,761.98,"WI"
"MARY","S",14,5691.78,2452.11,"WI"
"TOM","S",5,5669.12,798.15,"MN"
"BETH","H",12,4822.12,982.10,"WI"
```

Since EXCEL97 and newer versions can automatically convert HTML during the file open process, using ODS HTML to generate a HTML file and Excel to open it is one of the simplest methods for getting a report into Excel:

```
ods html body=’c:\temp\tabulate.html’;
proc tabulate data=softsale;
 title ’Softsale Sales and Expenses by Division’;
 class state division ;
 var sales expense;
 table state,division*sales*sum
             division*expense*sum;
run;
ods html close;
```

Results in (when viewed in a web browser) :



Softsale Sales and Expenses by Division

|  | Division | Division | Division | Division |
|---|---|---|---|---|
|  | H | S | H | S |
|  | Sales | Sales | Expense | Expense |
|  | Sum | Sum | Sum | Sum |
| State |  |  |  |  |
| IL | 4909.21 | 745.22 | 322.12 | 950.45 |
| MN | 3231.75 | 7732.44 | 644.95 | 1339.25 |
| WI | 12195.10 | 8232.11 | 2795.10 | 3339.41 |

Since Excel 2000 and newer versions recognize internal CSS (cascading style sheets), there is a new tagset called MSOffice2K  to take advantage of this. An example to generate a Excel file using the new tagset:

```
ODS tagsets.MSOffice2K body='c:\temp\softmark.xls';

proc print data=softsale;
 var state division sales expense years;
run;
ODS tagsets.MSOffice2K close;
```

Since Excel 2002 and newer versions can automatically convert XML during the file open process, Using ODS markup and a tagset like ExcelXP to generate a XML file and Excel 2002 to open the file is another method for getting a report into Excel:

```
ODS markup tagset=excelxp body='c:\temp\softmark.xls';

proc print data=softsale;
 var state division sales expense years;
run;
ODS markup close;
```

To get the best of both worlds, lets create a HTML file and allow the client to view it in a browser, and still have the option to save it as an Excel file! To do this we need to create a style using PROC TEMPLATE, create the HTML Report using the style we just created, Then we can display the report in a web browser (HTML) with a button labeled 'Download to Excel". For example this PROC TEMPLATE will create a style that displays a SAVE AS button on the HTML report  :

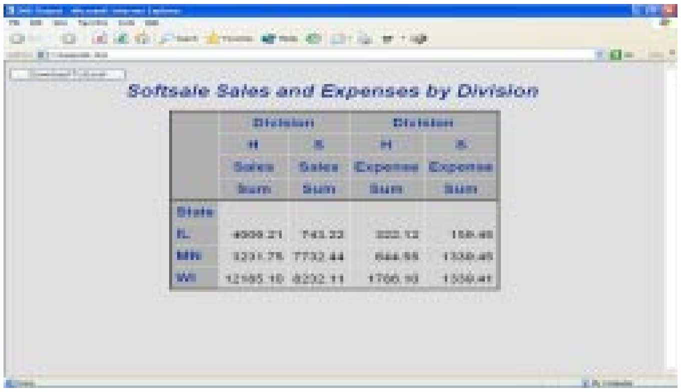
```
proc template;
  define style styles.test;
  parent=styles.default;
    style body from body /
    prehtml='<input onclick="document.execCommand(''SAVEAS'',true,
    ''c:\\temp\\test.xls'')"  value= "Download To Excel" type="button">';
  end;
run;
```

Now to create the HTML file using the style we just created:

```
ods html file='c:\temp\click.html' style=styles.test;
proc tabulate data=softsale;
 title 'Softsale Sales and Expenses by Division';
 class state division ;
 var sales expense;
 table state,division*sales*sum
             division*expense*sum;
```

```
     run;
     ods html close;
```

Now view the HTML file in a web browser, note the save as button labeled download to Excel:



### Softsale Sales and Expenses by Division

| | Division | | Division | |
|---|---|---|---|---|
| | H | S | H | S |
| | Sales | Sales | Expense | Expense |
| | Sum | Sum | Sum | Sum |
| State | | | | |
| IL | 4999.21 | 743.22 | 322.12 | 159.45 |
| MN | 3231.75 | 7732.44 | 644.55 | 1330.45 |
| WI | 12185.10 | 8232.11 | 1706.10 | 1559.41 |

## CONCLUSION

There are several techniques for moving Data or Reports from SAS to Excel. Options in the SAS system, and demonstrated techniques, make getting information to the people who need it, in a form they can easily manipulate, in a timely manner, easy to accomplish.

While each method has its advantages, the overall advantage of using SAS to generate our data file, or report remains the same, and these techniques will help distribute the data your organization needs to the people who need it.

## REFERENCES

1) “Using ODS to Generate Excel Files”, Chevell Parker,
SAS Institute

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Gerald Frey
Systems Seminar Consultants, Inc.
2997 Yarmouth Greenway Drive
Madison, WI 53711
Phone: (608) 278-9964
Fax: (608) 278-0065
Email: train@sys-seminar.com
Web: www.sys-seminar.com