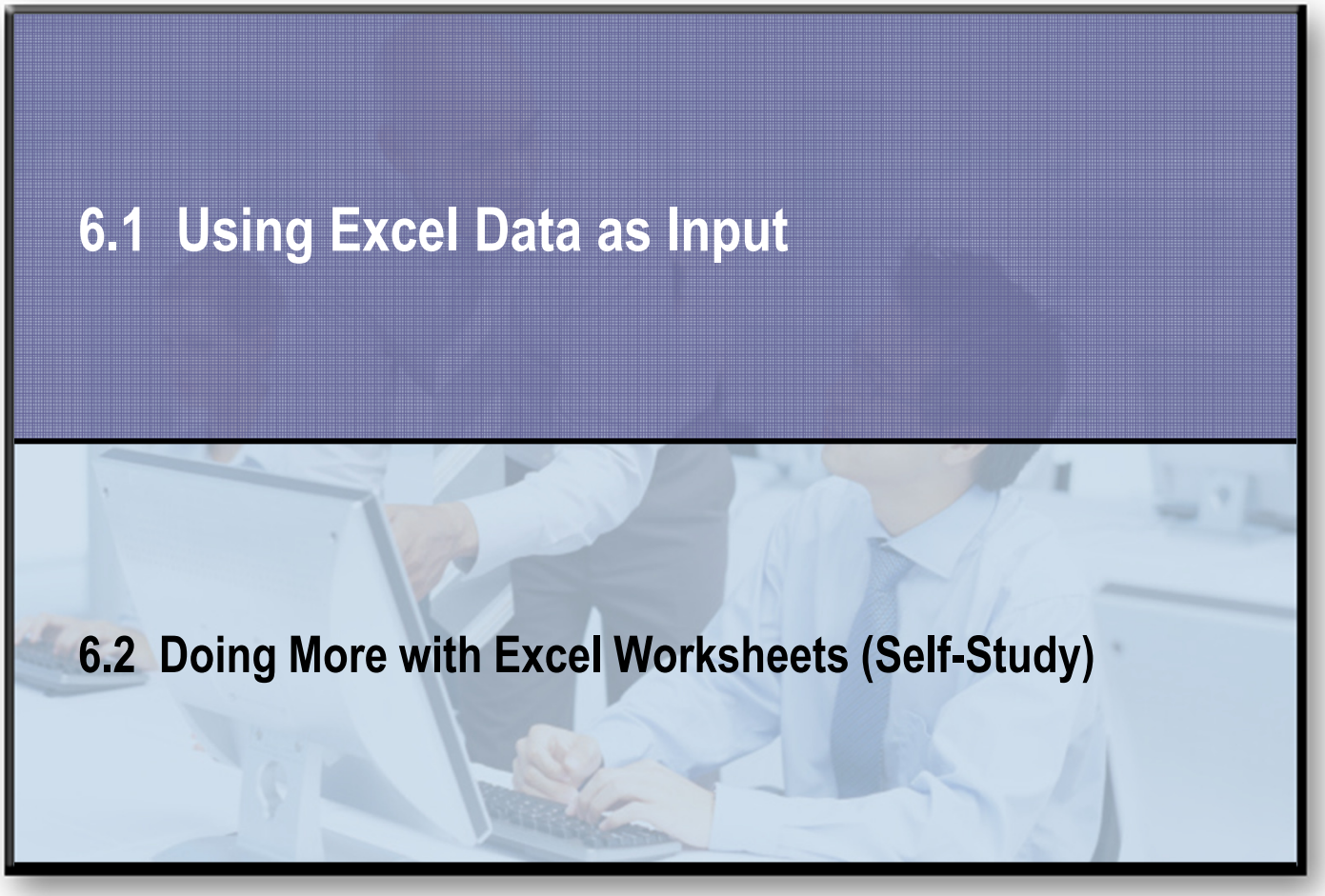
Chapter 6: Reading Excel Worksheets



Objectives

- Use the DATA step to create a SAS data set from an Excel worksheet.
- Use the SAS/ACCESS LIBNAME statement to read from an Excel worksheet as though it were a SAS data set.

Business Scenario

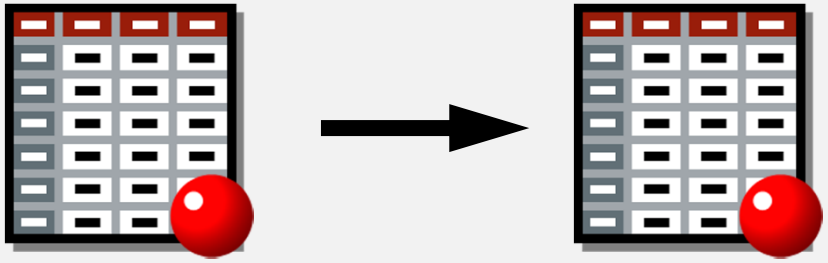
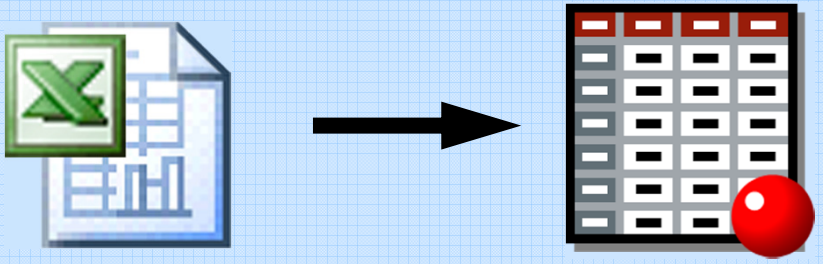
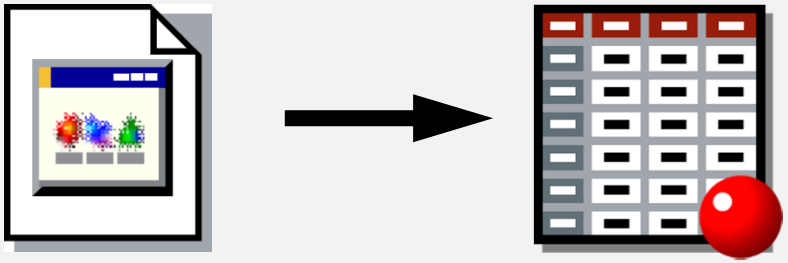
An existing data source contains information on Orion Star sales employees from Australia and the United States.

A new SAS data set needs to be created that contains a subset of this existing data source.

This new SAS data set must contain the following:

- only the employees from Australia who are Sales Representatives
- the employee's first name, last name, salary, job title, and hired date
- labels and formats in the descriptor portion

Business Scenario

Reading SAS Data Sets	
Reading Excel Worksheets	
Reading Delimited Raw Data Files	

Business Scenario

Reading SAS Data Sets	`libname _____;`
	`data _____;`
	`  set _____;`
	`  ...`
	`run;`

Reading Excel Worksheets	`libname _____;`
	`data _____;`
	`  set _____;`
	`  ...`
	`run;`

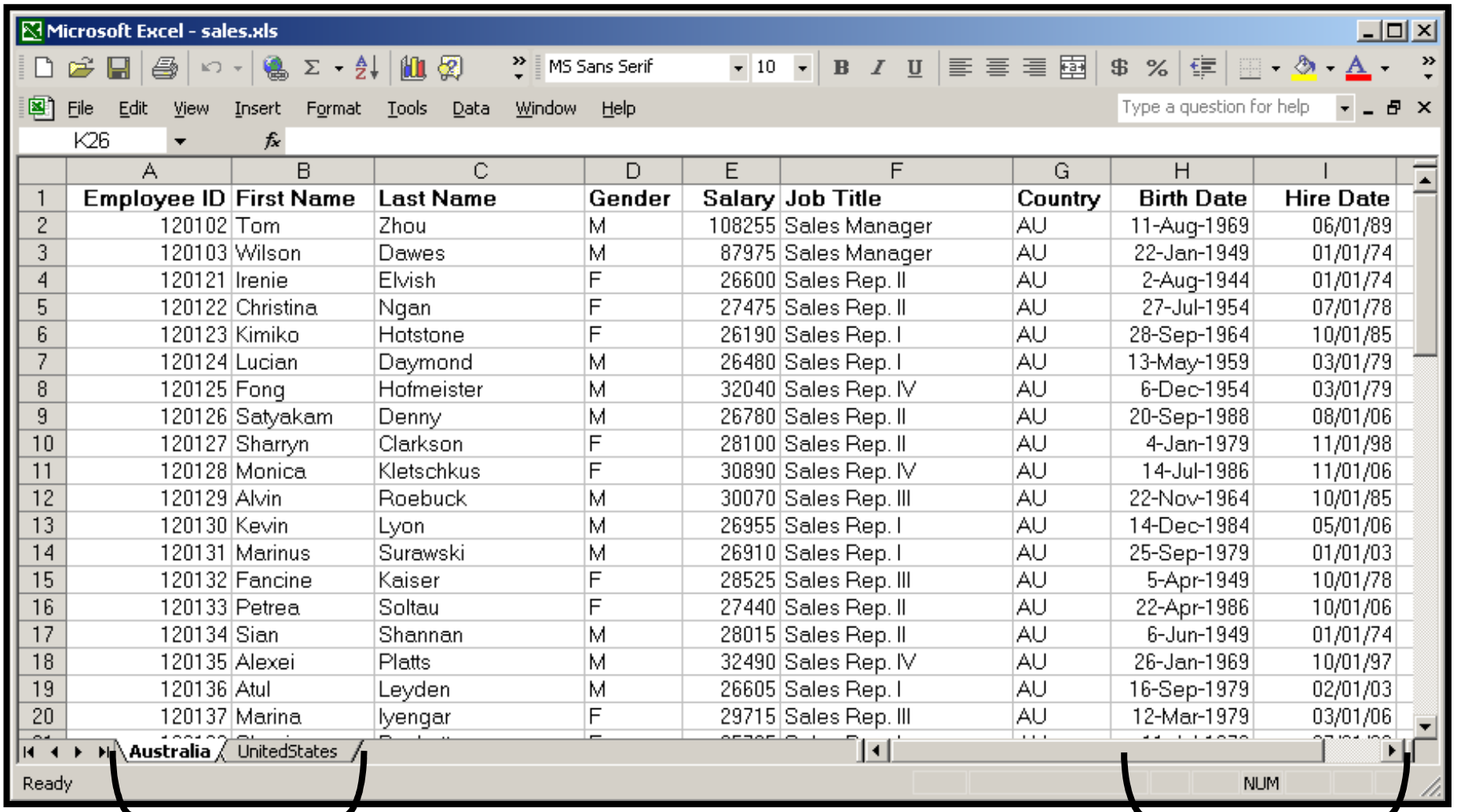
Reading Delimited Raw Data Files	`data _____;`
	`  infile _____;`
	`  input _____;`
	`  ...`
	`run;`

Business Scenario Syntax

Use the following statements to complete the scenario:

```
LIBNAME libref  'physical-file-name';

DATA output-SAS-data-set;
      SET input-SAS-data-set;
      WHERE where-expression;
      KEEP variable-list;
      LABEL variable = 'label'
                  variable = 'label'
                  variable = 'label';
      FORMAT variable(s) format ;
RUN;
```

sales.xls



	A	B	C	D	E	F	G	H	I
1	Employee ID	First Name	Last Name	Gender	Salary	Job Title	Country	Birth Date	Hire Date
2	120102	Tom	Zhou	M	108255	Sales Manager	AU	11-Aug-1969	06/01/89
3	120103	Wilson	Dawes	M	87975	Sales Manager	AU	22-Jan-1949	01/01/74
4	120121	Irenie	Elvish	F	26600	Sales Rep. II	AU	2-Aug-1944	01/01/74
5	120122	Christina	Ngan	F	27475	Sales Rep. II	AU	27-Jul-1954	07/01/78
6	120123	Kimiko	Hotstone	F	26190	Sales Rep. I	AU	28-Sep-1964	10/01/85
7	120124	Lucian	Daymond	M	26480	Sales Rep. I	AU	13-May-1959	03/01/79
8	120125	Fong	Hofmeister	M	32040	Sales Rep. IV	AU	6-Dec-1954	03/01/79
9	120126	Satyakam	Denny	M	26780	Sales Rep. II	AU	20-Sep-1988	08/01/06
10	120127	Sharryn	Clarkson	F	28100	Sales Rep. II	AU	4-Jan-1979	11/01/98
11	120128	Monica	Kletschkus	F	30890	Sales Rep. IV	AU	14-Jul-1986	11/01/06
12	120129	Alvin	Roebuck	M	30070	Sales Rep. III	AU	22-Nov-1964	10/01/85
13	120130	Kevin	Lyon	M	26955	Sales Rep. I	AU	14-Dec-1984	05/01/06
14	120131	Marinus	Surawski	M	26910	Sales Rep. I	AU	25-Sep-1979	01/01/03
15	120132	Fancine	Kaiser	F	28525	Sales Rep. III	AU	5-Apr-1949	10/01/78
16	120133	Petrea	Soltau	F	27440	Sales Rep. II	AU	22-Apr-1986	10/01/06
17	120134	Sian	Shannan	M	28015	Sales Rep. II	AU	6-Jun-1949	01/01/74
18	120135	Alexei	Platts	M	32490	Sales Rep. IV	AU	26-Jan-1969	10/01/97
19	120136	Atul	Leyden	M	26605	Sales Rep. I	AU	16-Sep-1979	02/01/03
20	120137	Marina	Iyengar	F	29715	Sales Rep. III	AU	12-Mar-1979	03/01/06

two worksheets

cells formatted as dates

The LIBNAME Statement (Review)

The *LIBNAME statement* assigns a library reference name (libref) to a SAS data library.

General form of the LIBNAME statement:

LIBNAME *libref 'SAS-data-library' <options>*;

Example:

```
libname orion 's:\workshop';
```

libref

physical location of SAS data library

The SAS/ACCESS LIBNAME Statement

The *SAS/ACCESS LIBNAME statement* extends the LIBNAME statement to support assigning a library reference name (libref) to Microsoft Excel workbooks.

General form of the SAS/ACCESS LIBNAME statement:

LIBNAME *libref* '*physical-file-name*' <*options*>**;**

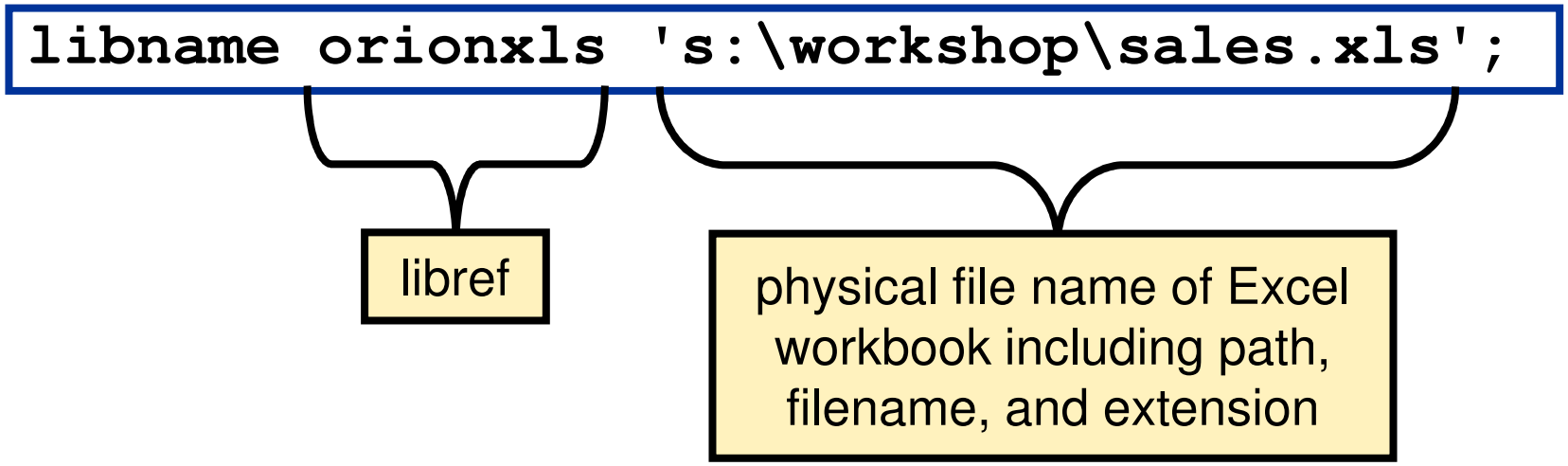
This enables you to reference worksheets directly in a DATA step or SAS procedure, and to read from and write to a Microsoft Excel worksheet as though it were a SAS data set.

The SAS/ACCESS LIBNAME Statement

SAS/ACCESS Interface to PC File Formats is required in order to use the SAS/ACCESS LIBNAME statement to access Excel workbooks.

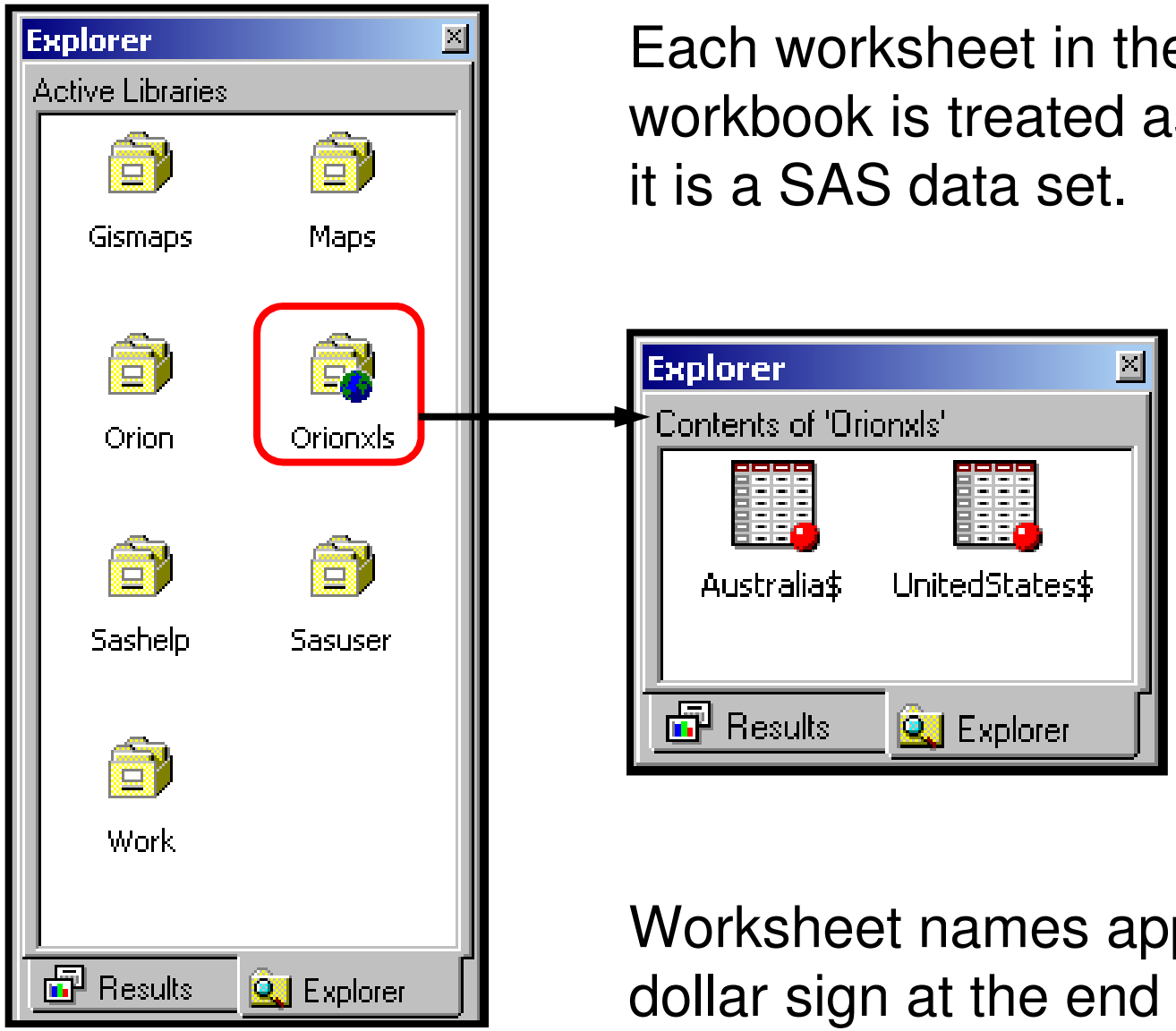
Example:

```
libname orionxls 's:\workshop\sales.xls';
```



p106d01

SAS Explorer Window



Each worksheet in the Excel workbook is treated as though it is a SAS data set.

Worksheet names appear with a dollar sign at the end of the name.

The CONTENTS Procedure

```
proc contents data=orionxls._all_;
run;
```

```
                    The CONTENTS Procedure

                          Directory

                 Libref          ORIONXLS
                 Engine          EXCEL
                 Physical Name   sales.xls
                 User            Admin


                                          DBMS
                               Member     Member
            #   Name           Type       Type

            1   Australia$     DATA       TABLE
            2   UnitedStates$  DATA       TABLE
```

p106d01

The CONTENTS Procedure

Data Set Name	ORIONXLS.'Australia$'n	Observations	.
Member Type	DATA	Variables	9
Engine	EXCEL	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
8	Birth_Date	Num	8	DATE9.	DATE9.	Birth Date
7	Country	Char	2	$2.	$2.	Country
1	Employee_ID	Num	8			Employee ID
2	First_Name	Char	10	$10.	$10.	First Name
4	Gender	Char	1	$1.	$1.	Gender
9	Hire_Date	Num	8	DATE9.	DATE9.	Hire Date
6	Job_Title	Char	14	$14.	$14.	Job Title
3	Last_Name	Char	12	$12.	$12.	Last Name
5	Salary	Num	8			Salary

The CONTENTS Procedure

Data Set Name	ORIONXLS.'UnitedStates$'n	Observations	.
Member Type	DATA	Variables	9
Engine	EXCEL	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

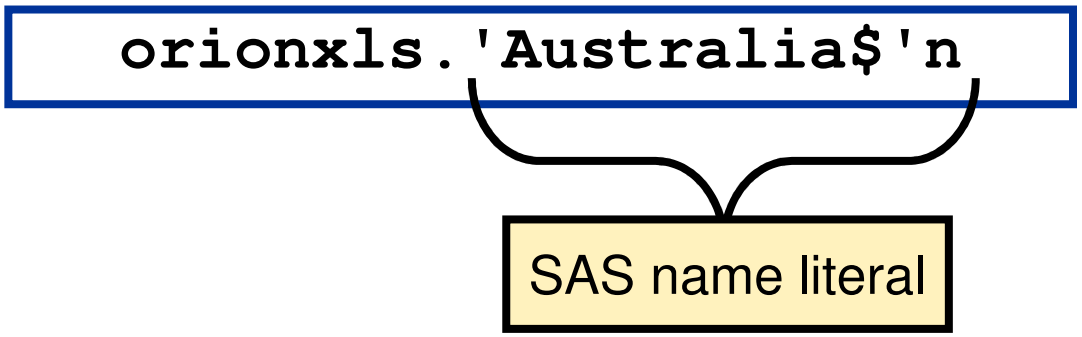
#	Variable	Type	Len	Format	Informat	Label
8	Birth_Date	Num	8	DATE9.	DATE9.	Birth Date
7	Country	Char	2	$2.	$2.	Country
1	Employee_ID	Num	8			Employee ID
2	First_Name	Char	12	$12.	$12.	First Name
4	Gender	Char	1	$1.	$1.	Gender
9	Hire_Date	Num	8	DATE9.	DATE9.	Hire Date
6	Job_Title	Char	20	$20.	$20.	Job Title
3	Last_Name	Char	18	$18.	$18.	Last Name
5	Salary	Num	8			Salary

SAS Name Literals

By default, special characters such as the $ are not allowed in data set names.

SAS name literals enable special characters to be included in data set names.

A *SAS name literal* is a name token that is expressed as a string within quotation marks, followed by the letter n.



The PRINT Procedure

```
proc print data=orionxls.'Australia$'n;
run;
```

Partial PROC PRINT Output

Obs	Employee_ ID	First_Name	Last_Name	Gender	Salary	Job_Title	Country	Birth_ Date	Hire_Date
1	120102	Tom	Zhou	M	108255	Sales Manager	AU	11AUG1969	01JUN1989
2	120103	Wilson	Dawes	M	87975	Sales Manager	AU	22JAN1949	01JAN1974
3	120121	Irenie	Elvish	F	26600	Sales Rep. II	AU	02AUG1944	01JAN1974
4	120122	Christina	Ngan	F	27475	Sales Rep. II	AU	27JUL1954	01JUL1978
5	120123	Kimiko	Hotstone	F	26190	Sales Rep. I	AU	28SEP1964	01OCT1985
6	120124	Lucian	Daymond	M	26480	Sales Rep. I	AU	13MAY1959	01MAR1979
7	120125	Fong	Hofmeister	M	32040	Sales Rep. IV	AU	06DEC1954	01MAR1979
8	120126	Satyakam	Denny	M	26780	Sales Rep. II	AU	20SEP1988	01AUG2006
9	120127	Sharryn	Clarkson	F	28100	Sales Rep. II	AU	04JAN1979	01NOV1998
10	120128	Monica	Kletschkus	F	30890	Sales Rep. IV	AU	14JUL1986	01NOV2006
11	120129	Alvin	Roebuck	M	30070	Sales Rep. III	AU	22NOV1964	01OCT1985
12	120130	Kevin	Lyon	M	26955	Sales Rep. I	AU	14DEC1984	01MAY2006
13	120131	Marinus	Surawski	M	26910	Sales Rep. I	AU	25SEP1979	01JAN2003
14	120132	Fancine	Kaiser	F	28525	Sales Rep. III	AU	05APR1949	01OCT1978
15	120133	Petrea	Soltau	F	27440	Sales Rep. II	AU	22APR1986	01OCT2006

p106d01

6.01 Quiz

Which PROC PRINT step displays the worksheet containing employees from the United States?

a.
```
proc print data=orionxls.'UnitedStates';
run;
```

b.
```
proc print data=orionxls.'UnitedStates$';
run;
```

c.
```
proc print data=orionxls.'UnitedStates'n;
run;
```

d.
```
proc print data=orionxls.'UnitedStates$'n;
run;
```

Business Scenario

Create a temporary SAS data set named **Work.subset2** from the Excel workbook named **sales.xls**.

```
libname orionxls 's:\workshop\sales.xls';

data work.subset2;
   set orionxls.'Australia$'n;
   where Job_Title contains 'Rep';
   keep First_Name Last_Name Salary
        Job_Title Hire_Date;
   label Job_Title='Sales Title'
         Hire_Date='Date Hired';
   format Salary comma10. Hire_Date weekdate.;
run;
```

Business Scenario

```
proc contents data=work.subset2;
run;
```

Partial PROC CONTENTS Output

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat	Label
1	First_Name	Char	10	$10.	$10.	First Name
5	Hire_Date	Num	8	WEEKDATE.	DATE9.	Date Hired
4	Job_Title	Char	14	$14.	$14.	Sales Title
2	Last_Name	Char	12	$12.	$12.	Last Name
3	Salary	Num	8	COMMA10.		Salary

Business Scenario

```
proc print data=work.subset2 label;
run;
```

Partial PROC PRINT Output

Obs	First Name	Last Name	Salary	Sales Title	Date Hired
1	Irenie	Elvish	26,600	Sales Rep. II	Tuesday, January 1, 1974
2	Christina	Ngan	27,475	Sales Rep. II	Saturday, July 1, 1978
3	Kimiko	Hotstone	26,190	Sales Rep. I	Tuesday, October 1, 1985
4	Lucian	Daymond	26,480	Sales Rep. I	Thursday, March 1, 1979
5	Fong	Hofmeister	32,040	Sales Rep. IV	Thursday, March 1, 1979
6	Satyakam	Denny	26,780	Sales Rep. II	Tuesday, August 1, 2006
7	Sharryn	Clarkson	28,100	Sales Rep. II	Sunday, November 1, 1998
8	Monica	Kletschkus	30,890	Sales Rep. IV	Wednesday, November 1, 2006
9	Alvin	Roebuck	30,070	Sales Rep. III	Tuesday, October 1, 1985
10	Kevin	Lyon	26,955	Sales Rep. I	Monday, May 1, 2006
11	Marinus	Surawski	26,910	Sales Rep. I	Wednesday, January 1, 2003
12	Fancine	Kaiser	28,525	Sales Rep. III	Sunday, October 1, 1978

Disassociating a Libref

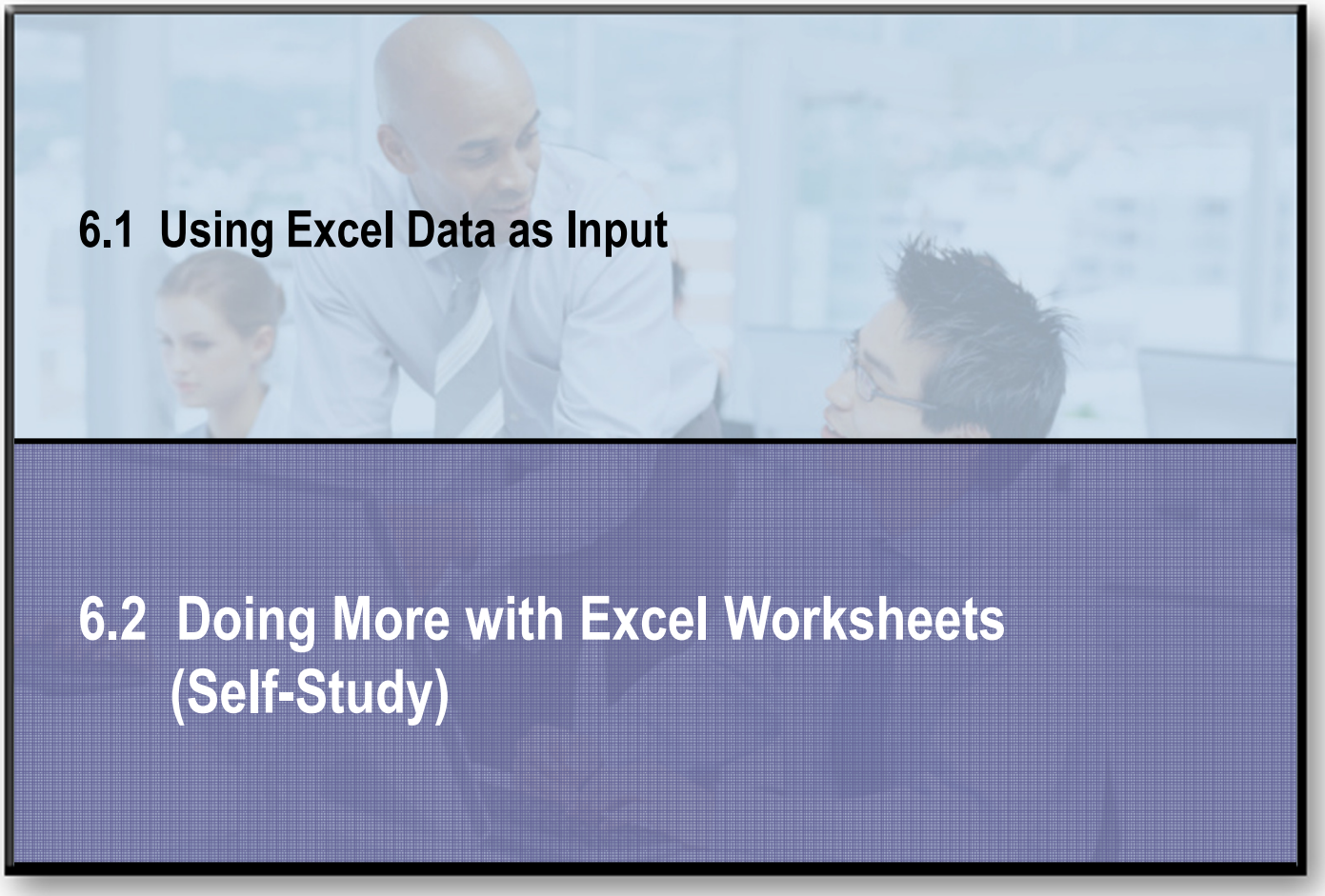
If SAS has a libref assigned to an Excel workbook, the workbook cannot be opened in Excel. To disassociate a libref, use a LIBNAME statement and specify the libref and the CLEAR option.

```
libname orionxls 's:\workshop\sales.xls';

data work.subset2;
   set orionxls.'Australia$'n;
   ...
run;

libname orionxls clear;
```

SAS disconnects from the data source and closes any resources that are associated with that libref's connection.

p106d02

Chapter 6: Reading Excel Worksheets



Objectives

- Use the DATA step to create an Excel worksheet from a SAS data set.
- Use the COPY procedure to create an Excel worksheet from a SAS data set.
- Use the IMPORT Wizard and procedure to read an Excel worksheet.
- Use the EXPORT Wizard and procedure to create an Excel worksheet.

Creating Excel Worksheets

In addition to reading an Excel worksheet, the SAS/ACCESS LIBNAME statement with the DATA step can be used to create an Excel worksheet.

```
libname orionxls
        's:\workshop\qtr2007a.xls';

data orionxls.qtr1_2007;
   set orion.qtr1_2007;
run;

data orionxls.qtr2_2007;
   set orion.qtr2_2007;
run;

proc contents data=orionxls._all_;
run;

libname orionxls clear;
```

Creating Excel Worksheets

Partial SAS Log

```
70   data orionxls.qtr1_2007;
71      set orion.qtr1_2007;
72
73   run;

NOTE: SAS variable labels, formats, and lengths are not written to DBMS tables.
NOTE: There were 22 observations read from the data set ORION.QTR1_2007.
NOTE: The data set ORIONXLS.qtr1_2007 has 22 observations and 5 variables.


74   data orionxls.qtr2_2007;
75      set orion.qtr2_2007;
76   run;

NOTE: SAS variable labels, formats, and lengths are not written to DBMS tables.
NOTE: There were 36 observations read from the data set ORION.QTR2_2007.
NOTE: The data set ORIONXLS.qtr2_2007 has 36 observations and 6 variables.
```

Creating Excel Worksheets

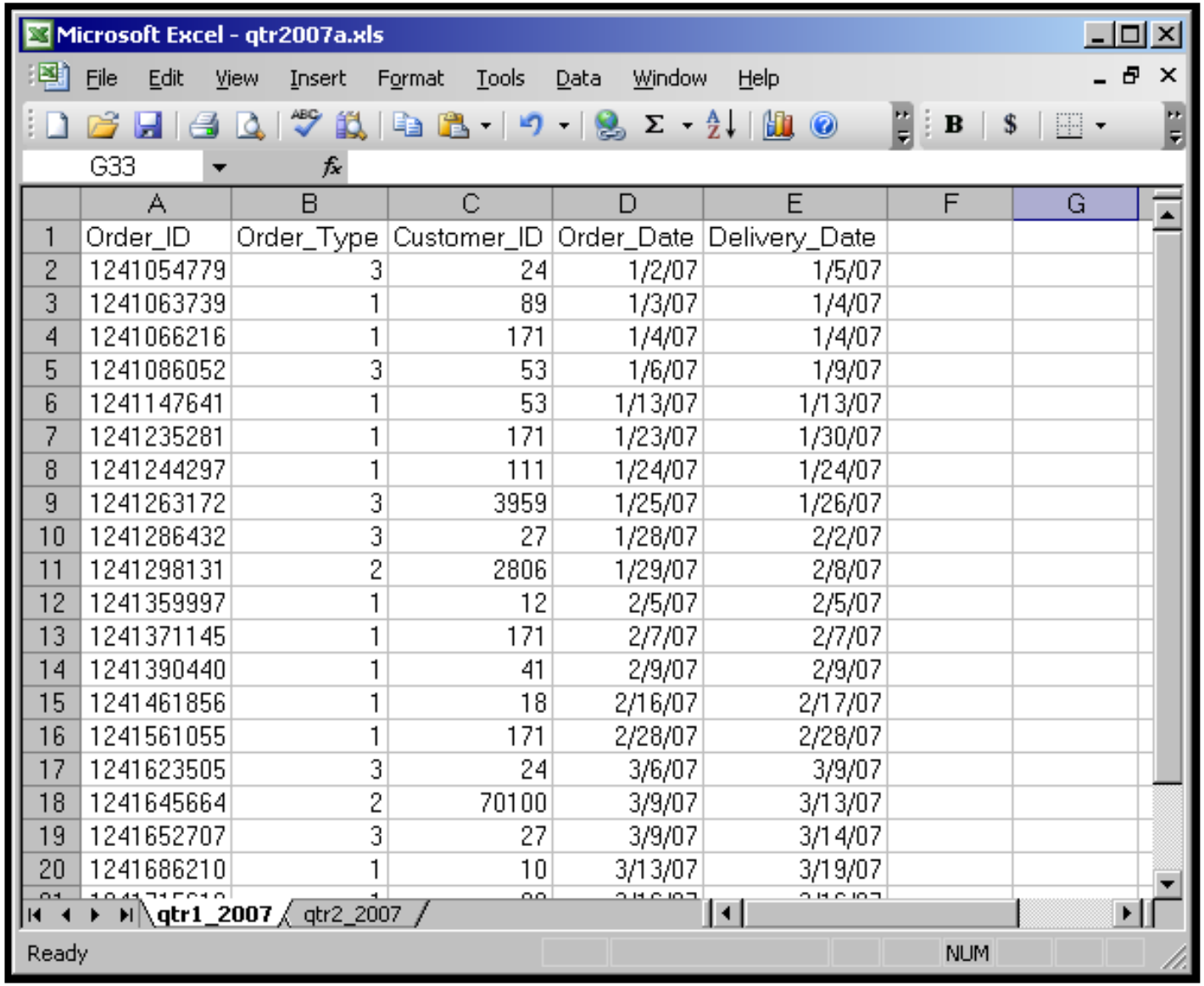
Partial PROC CONTENTS Output

```
                    The CONTENTS Procedure

                          Directory

                Libref          ORIONXLS
                Engine          EXCEL
                Physical Name   qtr2007a.xls
                User            Admin


                                          DBMS
                                Member    Member
               #  Name          Type      Type

               1  qtr1_2007     DATA      TABLE
               2  qtr1_2007$    DATA      TABLE
               3  qtr2_2007     DATA      TABLE
               4  qtr2_2007$    DATA      TABLE
```

worksheets

named ranges

Creating Excel Worksheets



	A	B	C	D	E	F	G
1	Order_ID	Order_Type	Customer_ID	Order_Date	Delivery_Date		
2	1241054779	3	24	1/2/07	1/5/07		
3	1241063739	1	89	1/3/07	1/4/07		
4	1241066216	1	171	1/4/07	1/4/07		
5	1241086052	3	53	1/6/07	1/9/07		
6	1241147641	1	53	1/13/07	1/13/07		
7	1241235281	1	171	1/23/07	1/30/07		
8	1241244297	1	111	1/24/07	1/24/07		
9	1241263172	3	3959	1/25/07	1/26/07		
10	1241286432	3	27	1/28/07	2/2/07		
11	1241298131	2	2806	1/29/07	2/8/07		
12	1241359997	1	12	2/5/07	2/5/07		
13	1241371145	1	171	2/7/07	2/7/07		
14	1241390440	1	41	2/9/07	2/9/07		
15	1241461856	1	18	2/16/07	2/17/07		
16	1241561055	1	171	2/28/07	2/28/07		
17	1241623505	3	24	3/6/07	3/9/07		
18	1241645664	2	70100	3/9/07	3/13/07		
19	1241652707	3	27	3/9/07	3/14/07		
20	1241686210	1	10	3/13/07	3/19/07		

Creating Excel Worksheets

As an alternative to the DATA step, the COPY procedure can be used to create an Excel worksheet.

```
libname orionxls
        's:\workshop\qtr2007b.xls';

proc copy  in=orion out=orionxls;
   select qtr1_2007 qtr2_2007;
run;

proc contents data=orionxls._all_;
run;

libname orionxls clear;
```

Creating Excel Worksheets

Partial SAS Log

```
82   proc copy  in=orion out=orionxls;
83      select qtr1_2007 qtr2_2007;
84   run;

NOTE: Copying ORION.QTR1_2007 to ORIONXLS.QTR1_2007 (memtype=DATA).
NOTE: SAS variable labels, formats, and lengths are not written to DBMS tables.
NOTE: There were 22 observations read from the data set ORION.QTR1_2007.
NOTE: The data set ORIONXLS.QTR1_2007 has 22 observations and 5 variables.
NOTE: Copying ORION.QTR2_2007 to ORIONXLS.QTR2_2007 (memtype=DATA).
NOTE: SAS variable labels, formats, and lengths are not written to DBMS tables.
NOTE: There were 36 observations read from the data set ORION.QTR2_2007.
NOTE: The data set ORIONXLS.QTR2_2007 has 36 observations and 6 variables.
```

Import/Export Wizards and Procedures

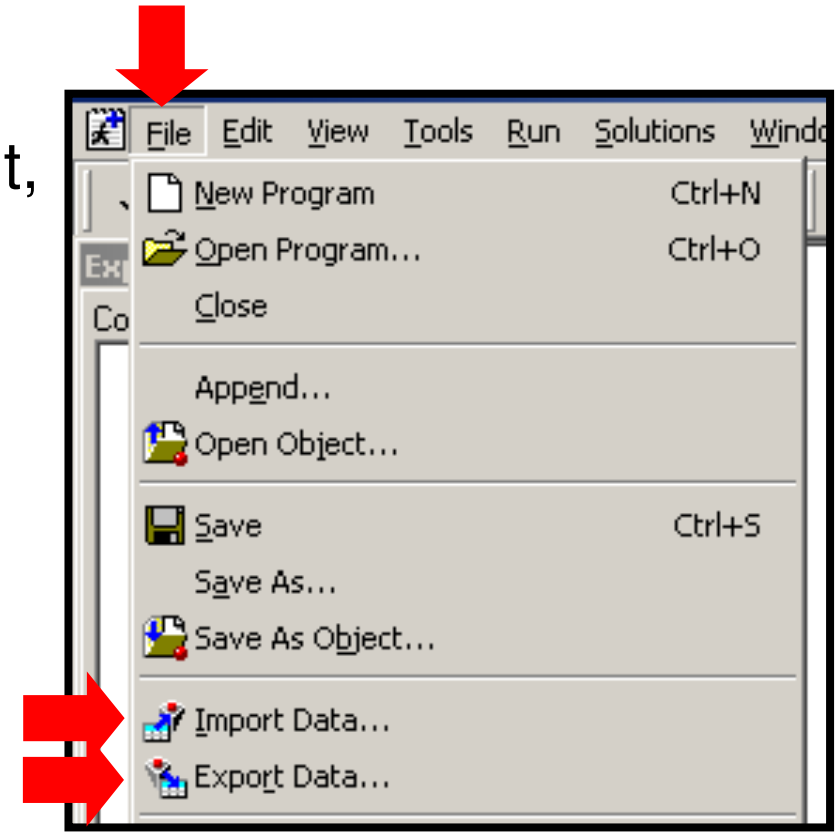
The Import/Export Wizards and IMPORT/EXPORT procedures enable you to read and write data between SAS data sets and external PC files.

The Import/Export Wizards and procedures are part of Base SAS and enable access to delimited files. If you have a license to SAS/ACCESS Interface to PC File Formats, you can also access Microsoft Excel, Microsoft Access, dBASE, JMP, Lotus 1-2-3, SPSS, Stata, and Paradox files.

Import/Export Wizards and Procedures

The wizards and procedures have similar capabilities; the wizards are point-and-click interfaces and the procedures are code-based.

To invoke the wizards from the SAS windowing environment, select **File** and **Import Data** or **Export Data**.



The Import Wizard

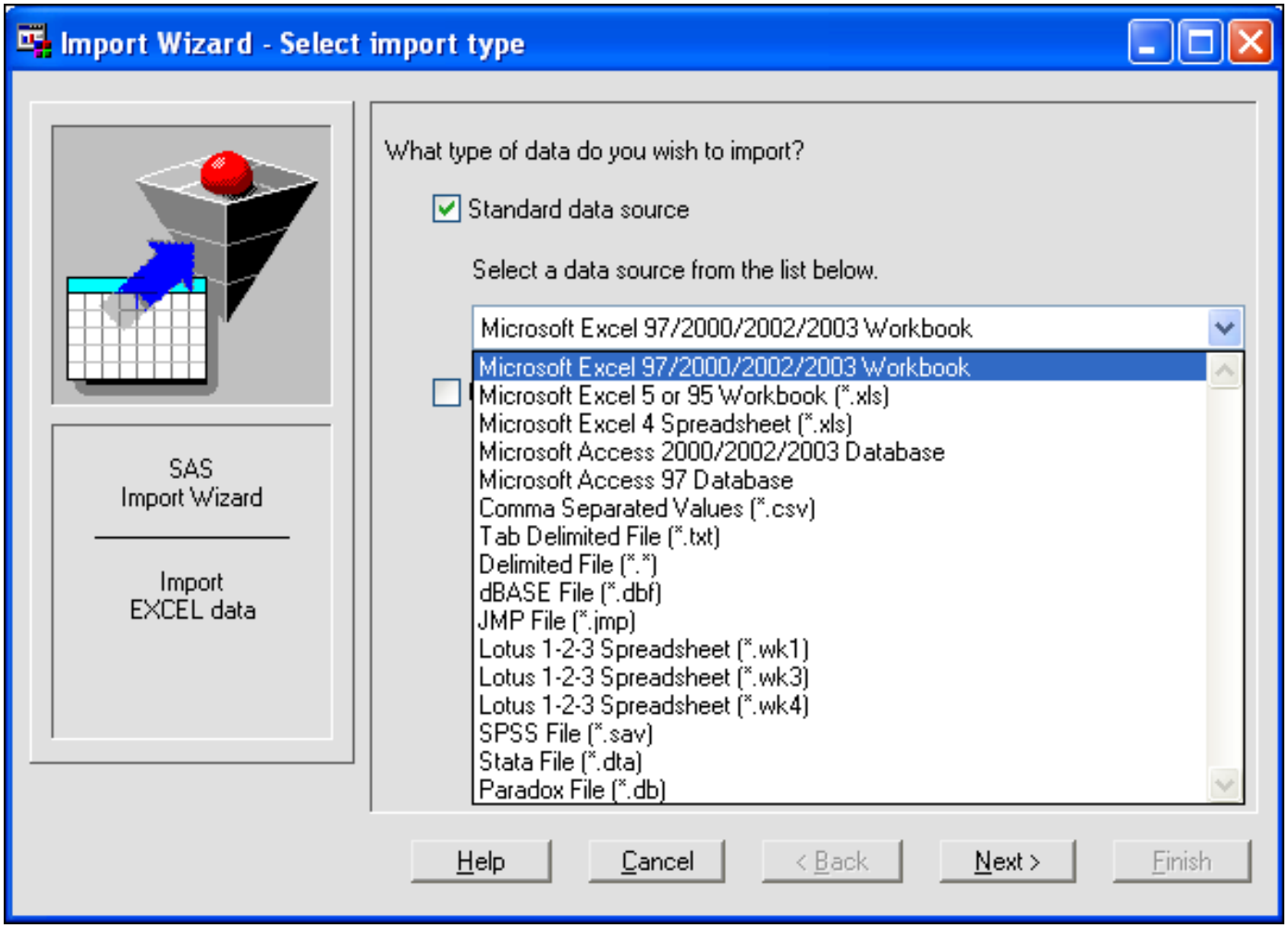
The Import Wizard enables you to read data from an external data source and write it to a SAS data set.

Steps of the Import Wizard:

1. Select the type of file you are importing.
2. Locate the input file.
3. Select the table range or worksheet from which to import data.
4. Select a location to store the imported file.
5. Save the generated PROC IMPORT code. (Optional)

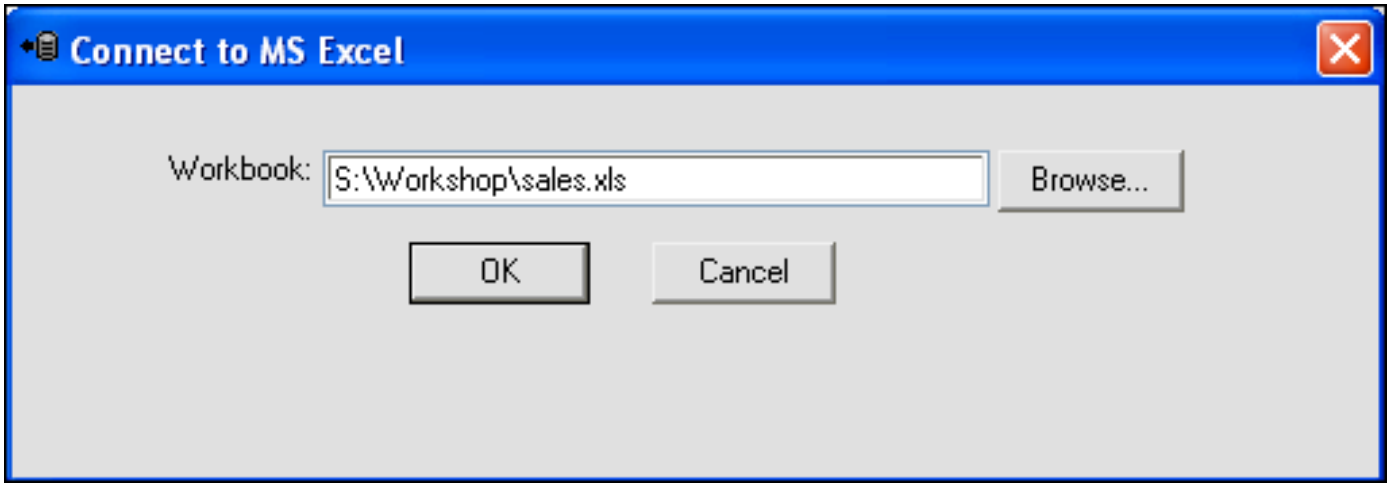
The Import Wizard

1. Select the type of file you are importing.



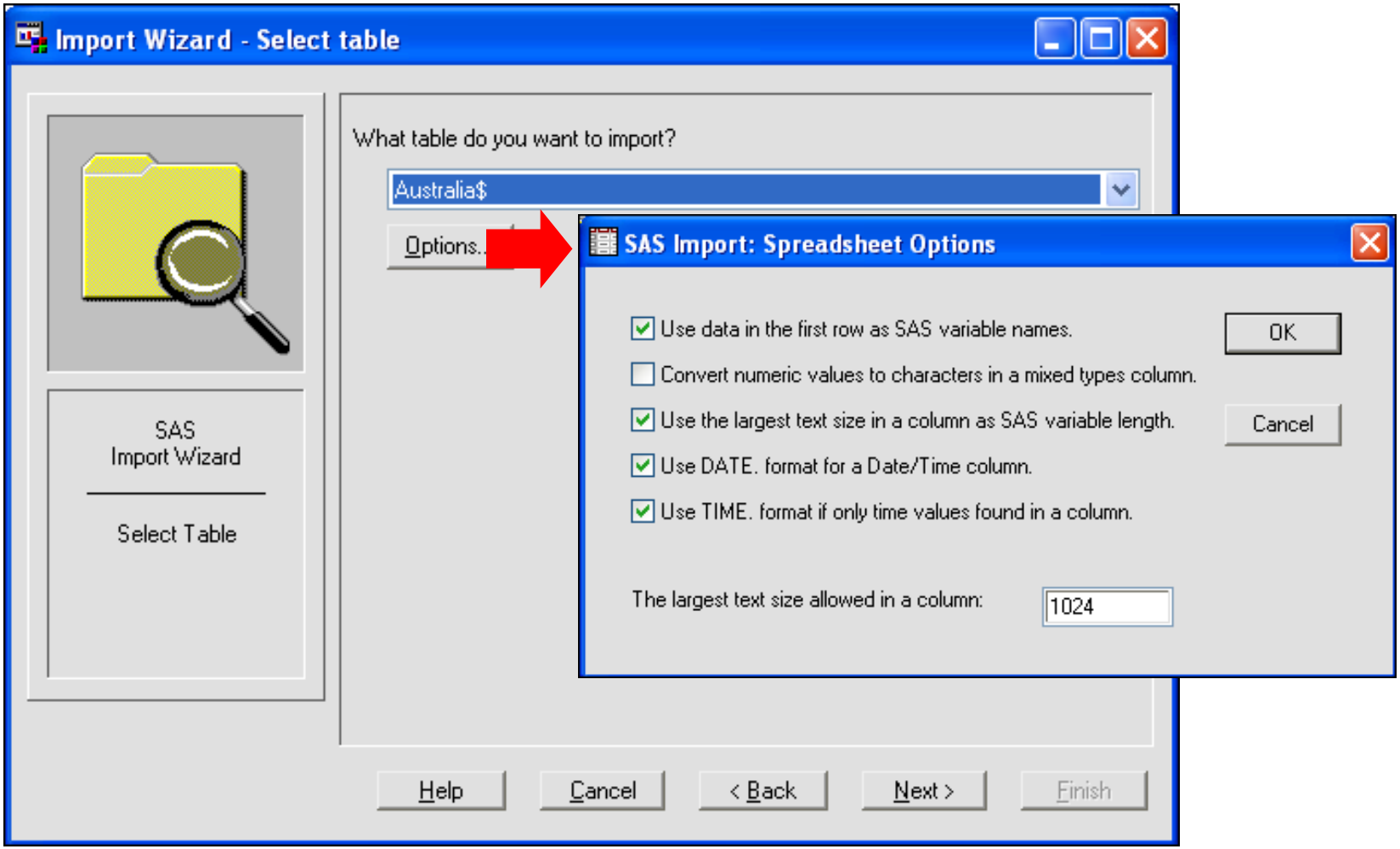
The Import Wizard

2. Locate the input file.



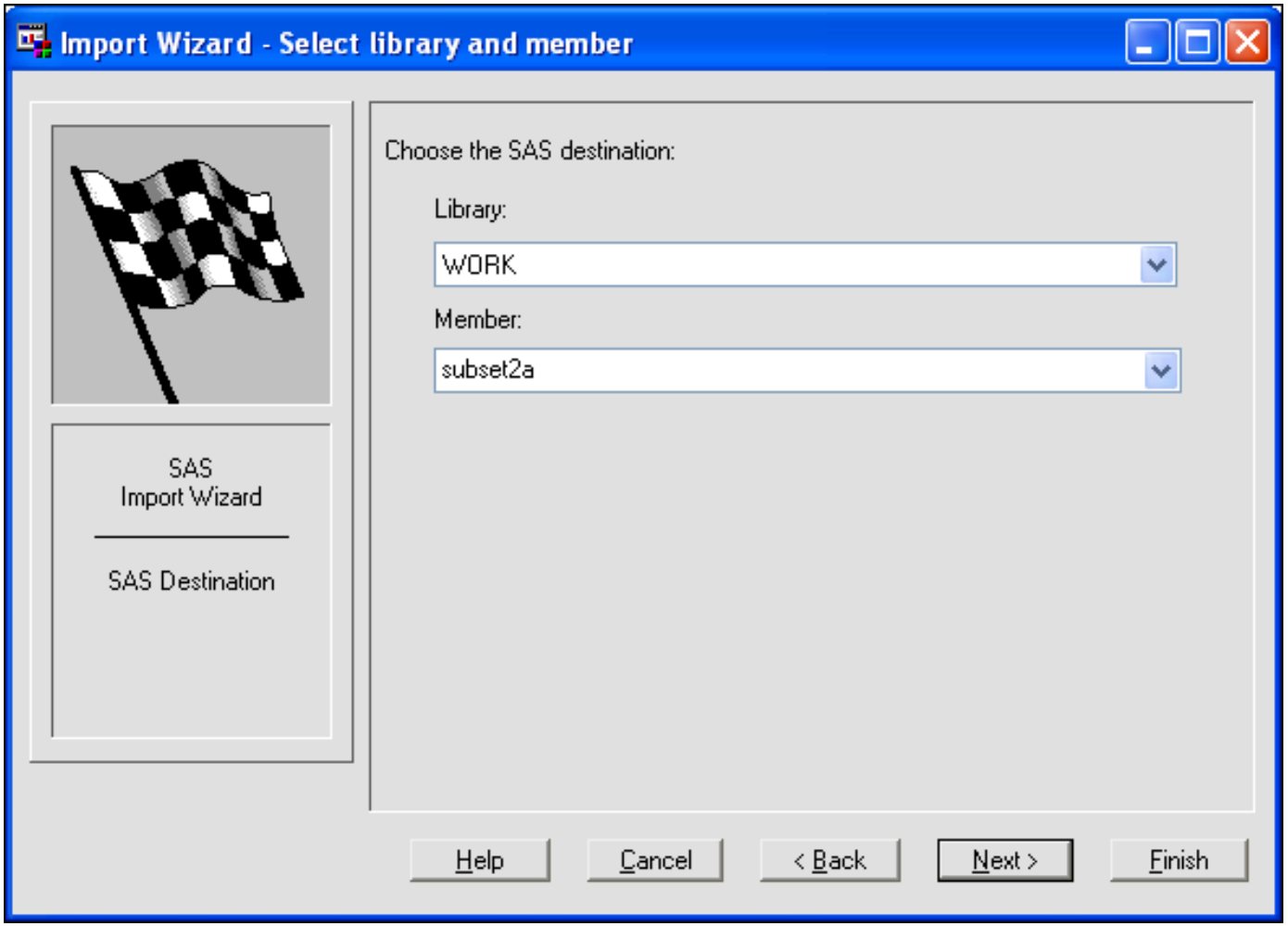
The Import Wizard

3. Select the table range or worksheet from which to import data.



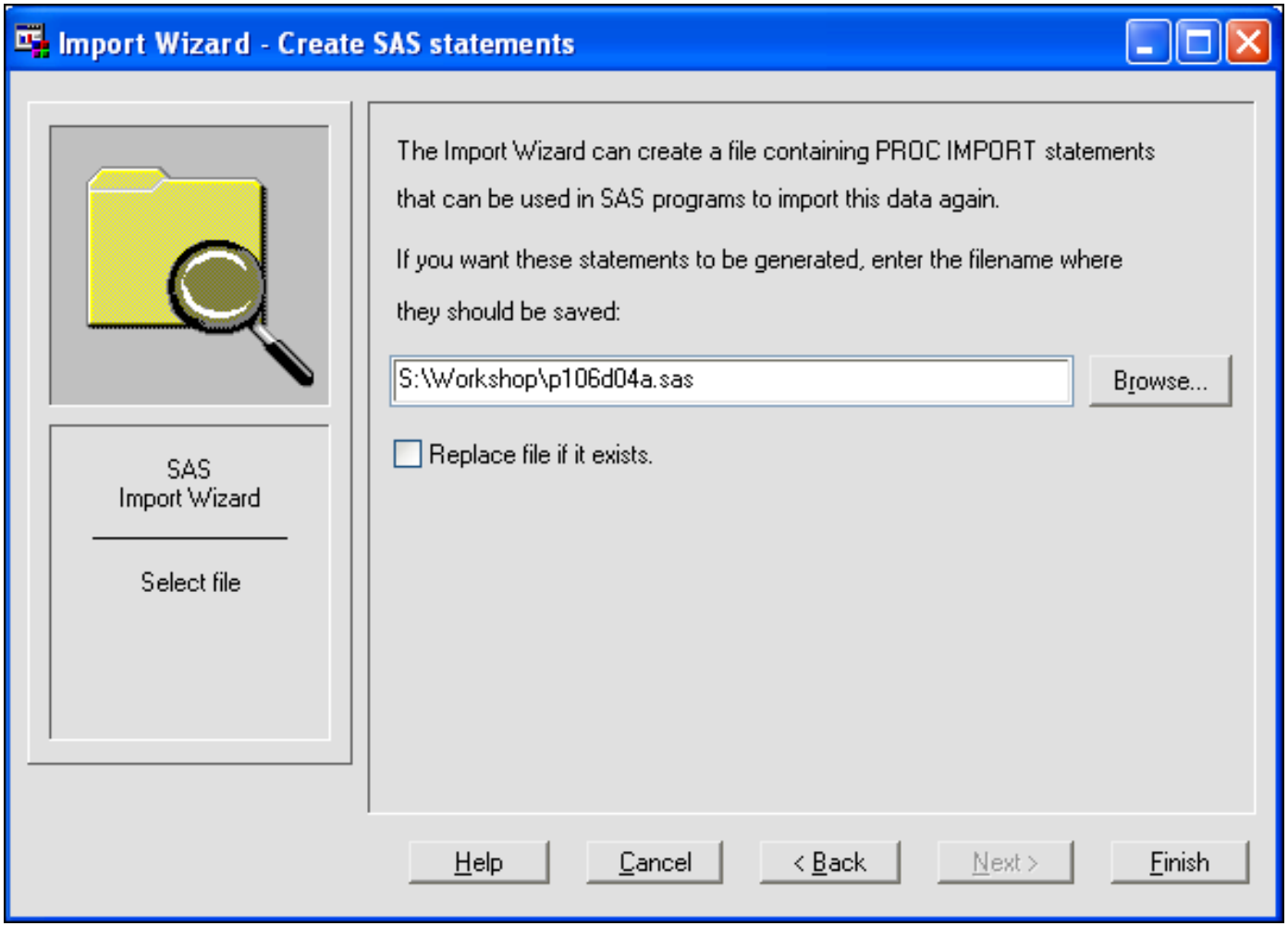
The Import Wizard

4. Select a location to store the imported file.



The Import Wizard

5. Save the generated PROC IMPORT code. (Optional)



The Import Wizard

SAS Log

```
NOTE: WORK.SUBSET2A data set was successfully created.
```

```
proc print data=work.subset2a;
run;
```

Partial PROC PRINT Output

Obs	Employee_ ID	First_Name	Last_Name	Gender	Salary	Job_Title	Country	Birth_ Date	Hire_Date
1	120102	Tom	Zhou	M	108255	Sales Manager	AU	11AUG1969	01JUN1989
2	120103	Wilson	Dawes	M	87975	Sales Manager	AU	22JAN1949	01JAN1974
3	120121	Irenie	Elvish	F	26600	Sales Rep. II	AU	02AUG1944	01JAN1974
4	120122	Christina	Ngan	F	27475	Sales Rep. II	AU	27JUL1954	01JUL1978
5	120123	Kimiko	Hotstone	F	26190	Sales Rep. I	AU	28SEP1964	01OCT1985

The Import Wizard

```
proc contents data=work.subset2a;
run;
```

Partial PROC CONTENTS Output

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat	Label
8	Birth_Date	Num	8	DATE9.	DATE9.	Birth Date
7	Country	Char	2	$2.	$2.	Country
1	Employee_ID	Num	8			Employee ID
2	First_Name	Char	10	$10.	$10.	First Name
4	Gender	Char	1	$1.	$1.	Gender
9	Hire_Date	Num	8	DATE9.	DATE9.	Hire Date
6	Job_Title	Char	14	$14.	$14.	Job Title
3	Last_Name	Char	12	$12.	$12.	Last Name
5	Salary	Num	8			Salary

p106d04

The IMPORT Procedure

The program **p106d04a** was created from the Import Wizard.

```
PROC IMPORT OUT= WORK.subset2a
            DATAFILE= "S:\Workshop\sales.xls"
            DBMS=EXCEL REPLACE;
     RANGE="Australia$";
     GETNAMES=YES;
     MIXED=NO;
     SCANTEXT=YES;
     USEDATE=YES;
     SCANTIME=YES;
RUN;
```

The Export Wizard

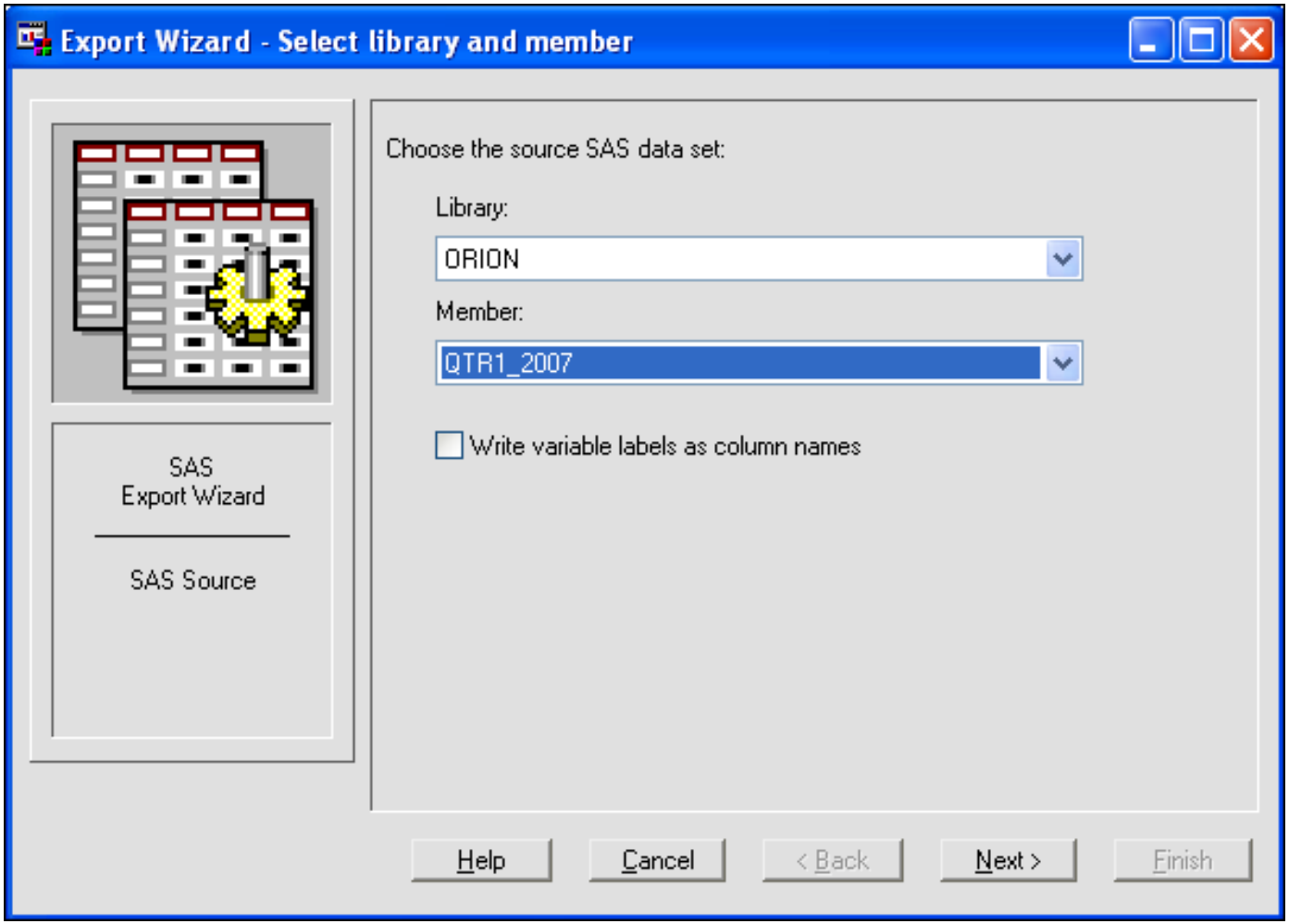
The Export Wizard reads data from a SAS data set and writes it to an external file source.

Steps of the Export Wizard:

1. Select the data set from which you want to export data.
2. Select the type of data source to which you want to export files.
3. Assign the output file.
4. Assign the table name.
5. Save the generated PROC EXPORT code. (Optional)

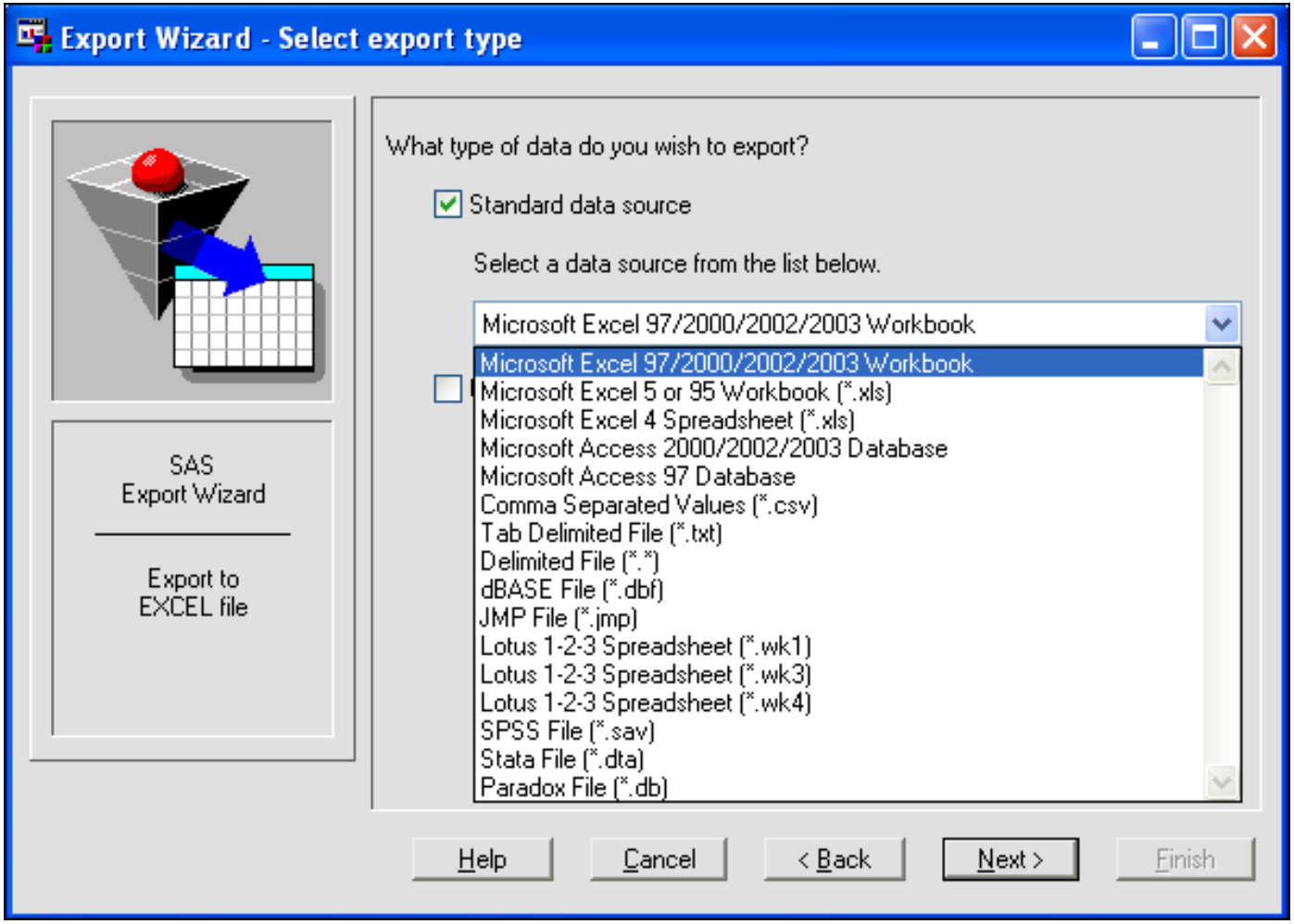
The Export Wizard

1. Select the data set from which you want to export data.



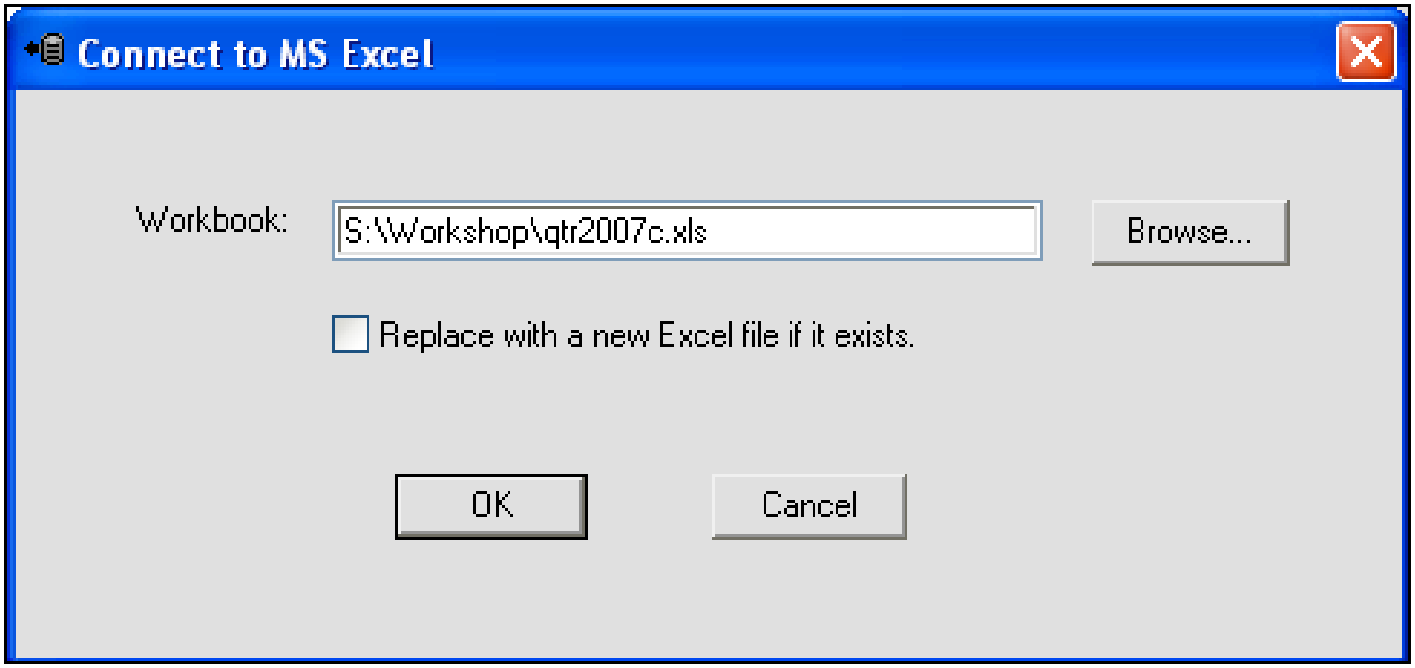
The Export Wizard

2. Select the type of data source to which you want to export files.



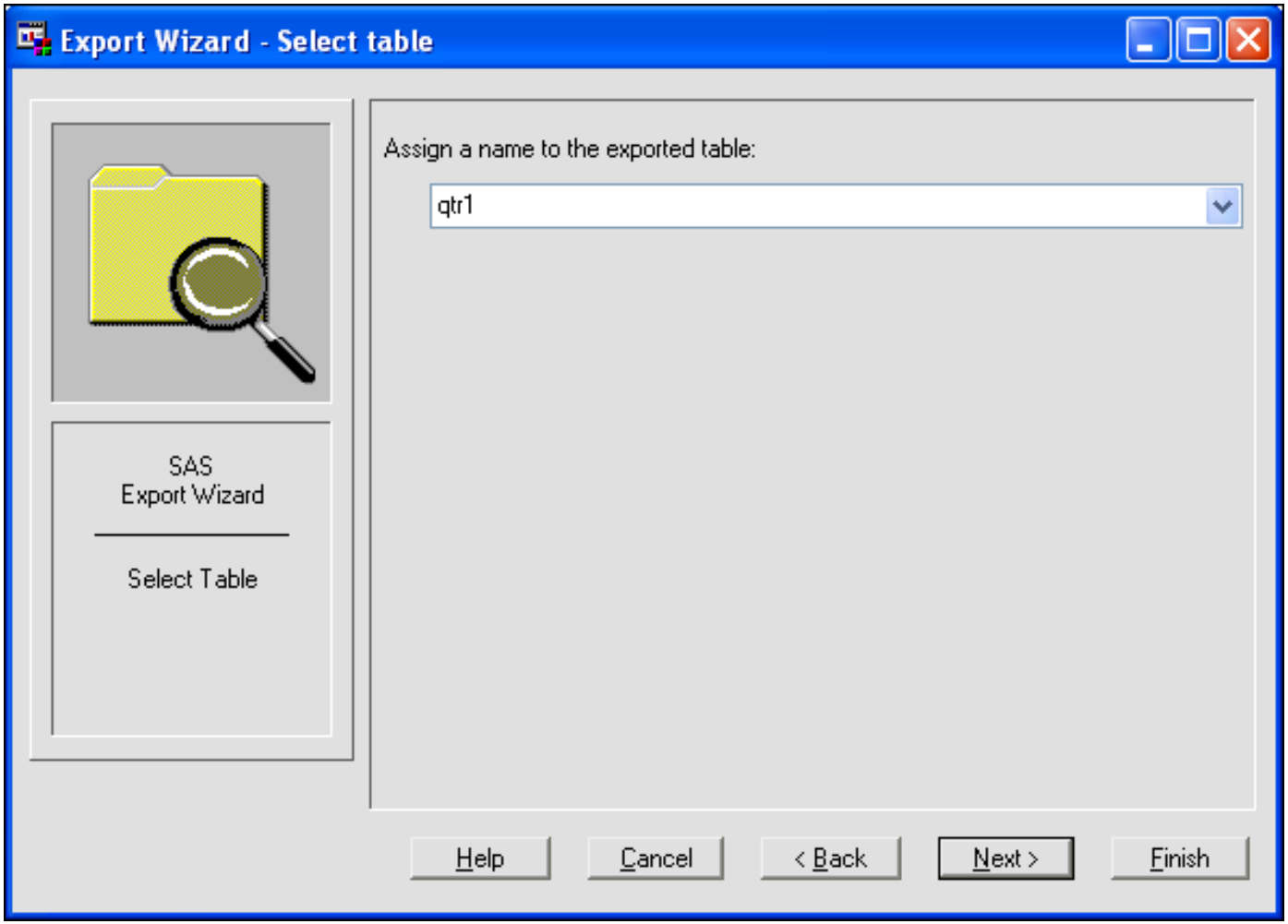
The Export Wizard

3. Assign the output file.



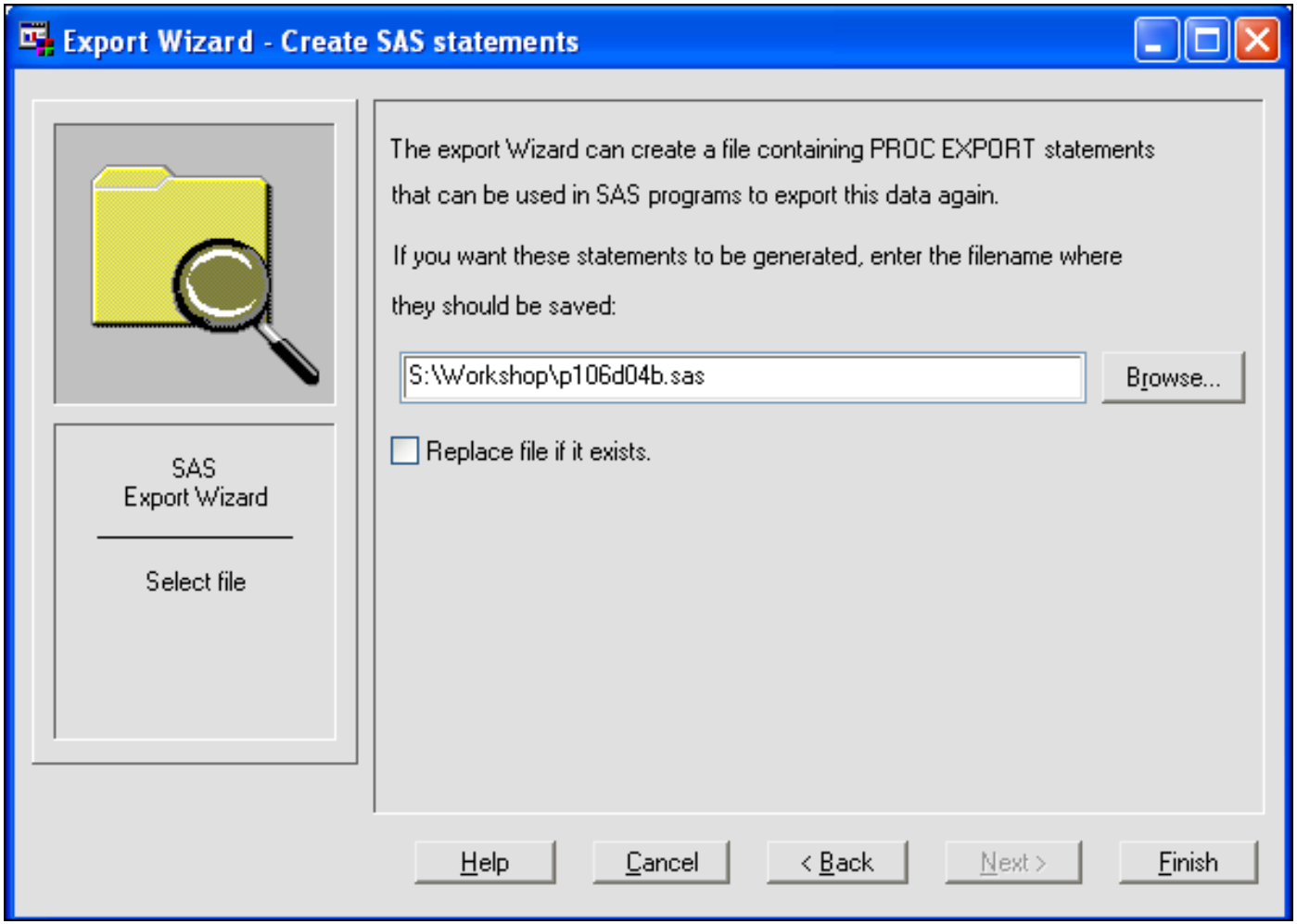
The Export Wizard

4. Assign the table name.



The Export Wizard

5. Save the generated PROC EXPORT code. (Optional)

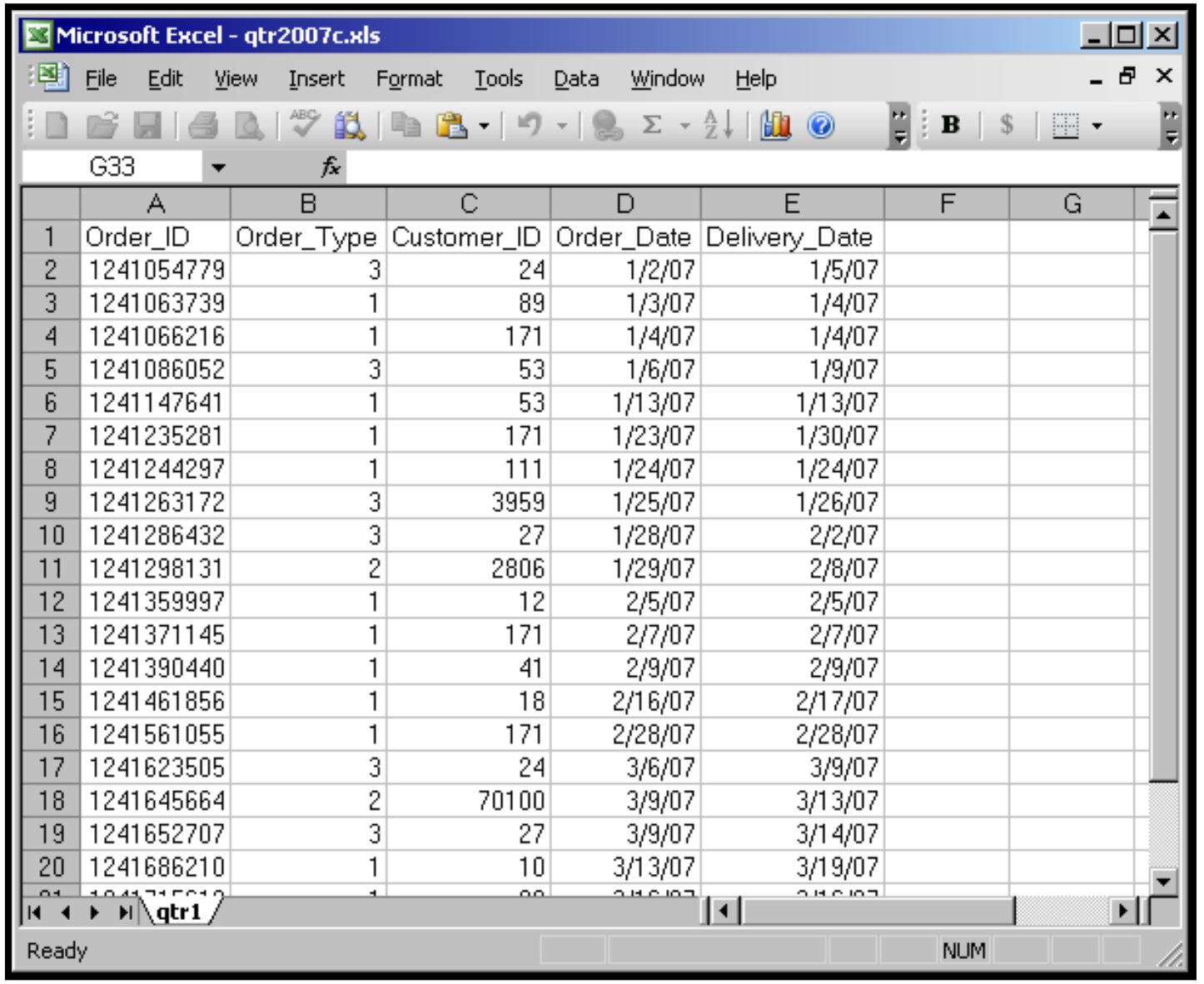


The Export Wizard

SAS Log

```
NOTE: File "S:\Workshop\qtr2007c.xls" will be created if the export
      process succeeds.
NOTE: "qtr1" table was successfully created.
```

The Export Wizard



	A	B	C	D	E	F	G
1	Order_ID	Order_Type	Customer_ID	Order_Date	Delivery_Date		
2	1241054779	3	24	1/2/07	1/5/07		
3	1241063739	1	89	1/3/07	1/4/07		
4	1241066216	1	171	1/4/07	1/4/07		
5	1241086052	3	53	1/6/07	1/9/07		
6	1241147641	1	53	1/13/07	1/13/07		
7	1241235281	1	171	1/23/07	1/30/07		
8	1241244297	1	111	1/24/07	1/24/07		
9	1241263172	3	3959	1/25/07	1/26/07		
10	1241286432	3	27	1/28/07	2/2/07		
11	1241298131	2	2806	1/29/07	2/8/07		
12	1241359997	1	12	2/5/07	2/5/07		
13	1241371145	1	171	2/7/07	2/7/07		
14	1241390440	1	41	2/9/07	2/9/07		
15	1241461856	1	18	2/16/07	2/17/07		
16	1241561055	1	171	2/28/07	2/28/07		
17	1241623505	3	24	3/6/07	3/9/07		
18	1241645664	2	70100	3/9/07	3/13/07		
19	1241652707	3	27	3/9/07	3/14/07		
20	1241686210	1	10	3/13/07	3/19/07		

The EXPORT Procedure

The program **p106d04b** was created from the Export Wizard.

```
PROC EXPORT DATA= ORION.QTR1_2007
            OUTFILE= "S:\Workshop\qtr2007c.xls"
            DBMS=EXCEL REPLACE;
     RANGE="qtr1";
RUN;
```

- The RANGE statement is not supported and is ignored in the EXPORT procedure.

Chapter Review

1. What statement is used to point to a physical filename including the path, filename, and extension of an Excel workbook ?

2. What character appears at the end of an Excel worksheet name in the SAS Explorer?

3. What is an example of a SAS name literal?

4. How do you disassociate a libref?