**Chapter XX – Creating Multiple Observations from a Single Record**

1. Reading Repeating Blocks of Data

* Holding the Current Record with a Line-Hold Specifier

To hold the current record so the input statement can read, and SAS can output, repeated blocks of data on the same record.

* Two line-hold specifiers
  + The trailing at sign (@) holds the input record for the execution of the next **INPUT** statement.
  + The double trailing at sign (@@) holds the input record for the execution of the next **INPUT** statement, even across iterations of the DATA step. (Hold multiple values)
* 注意：@和@@必须在被**INPUT** statement specify的item末尾
* Using the Double Trailing At Sign (@@) to Hold the Current Record

一般情况下在每次**DATA** step执行后，**INPUT** statement会换行开始读取下一个record。但是在使用trailing @@后，**INPUT** statement会在同一行继续读取下一个record

Eg:

**INPUT** Score; **INPUT** Score @@;



* 注意：@@不可以和@ pointer control，column input，and MISSOVER option共同使用
* A record that is held by the double trailing at sign (@@) is not released until either of the following events occurs:
* The input pointer moves past the end of the record. Then the input pointer moves down to the next record.

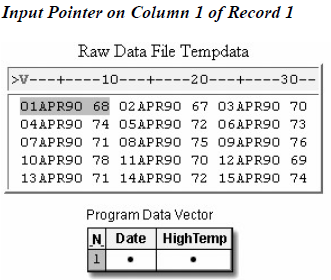


* An INPUT statement that has no trailing at sign executes.

Eg:

**INPUT** ID $ @@;

.

.

**INPUT** Department 5.;

* Eg:

**DATA** perm.april90;

**INFILE** tempdata;

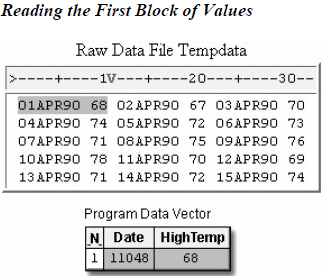
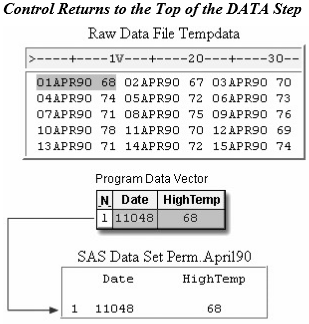
**INPUT** Date : **DATE**. HighTemp @@;

**FORMAT** date **DATE9**.;

**RUN**;

**PROC PRINT** data=perm.april90;

**RUN**;

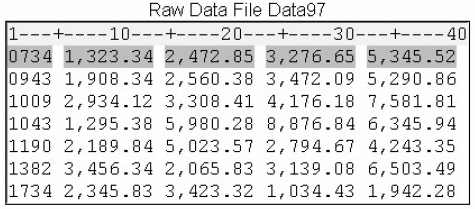
1. Reading the Same Number of Repeating Fields

Hold一个固定的variable，同时读取另一个record的值。此时，you must execute the DATA step once for each record, repetitively reading and writing values in one iteration。

* Different between double trailing at sign (@@) and single trailing at sign (@)
* Double trailing at sign (@@)可以hold多个iterations of the DATA step 直到读取到record的末尾
* Single trailing at sign (@) releases a record when control return to the top of the DATA step (Quiz 2)
* 如果在一个record被hold之后需要进行重复读取的话，我们可以使用DO loop来简化步骤增加运行效率：

Eg：

Each record in the file Data97 contains a sales representative's ID number, followed by four repeating fields that represent his or her quarterly sales totals for 1997. You want to pair each employee ID number with one quarterly sales total to produce a single observation. Four observations are generated from each record.



**DATA** perm.sales97;

**INFILE** data97;

Placing an explicit OUTPUT statement in a DATA step overrides the automatic output, and SAS adds an observation to a data set only when the explicit OUTPUT statement is executed.

**INPUT** ID $ @;

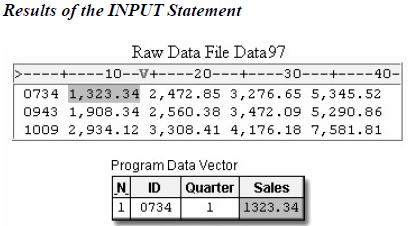
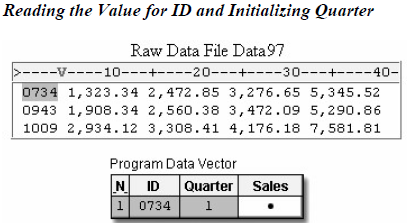
**DO** Quarter=1 **TO** 4;

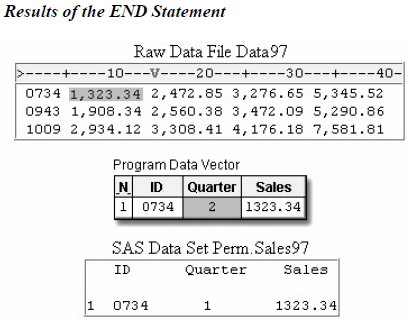
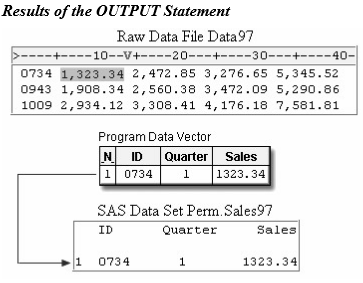
**INPUT** Sales : **COMMA**. @;

**OUTPUT**;

**END**;

**RUN**;







Eg:

**DATA** perm.sales97;

**INFILE** data97;

**INPUT** ID $ @;

**DO** Quarter=1 **TO** 4;

**INPUT** Sales : **COMMA**. @;

**OUTPUT**;

**END**;

**RUN**;

1. Reading a Varying Number of Repeating Fields

当records中的variable数量不同时，使用：或者&来帮助SAS读取数据。



* Using the **MISSOVER** Option

To prevent the input pointer from moving to the next record when there are missing Sales values.

* 在上个example中，因为每个record含有的variable数量不相等，可能会在末未出现missing value的现象，所以我们在程序中加入**MISSOVER**来防止误读或者错误

Eg:

**DATA** perm.sales97;

**INFILE** data97 **MISSOVER**;

**INPUT** ID $ Sales : **COMMA**. @;

Quarter=0;

**DO** **WHILE** (sales ne .);

quarter+1;

**OUTPUT**;

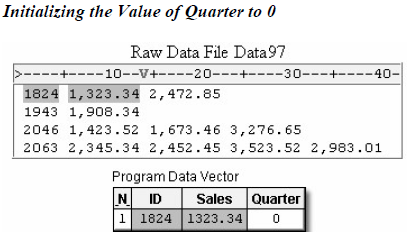
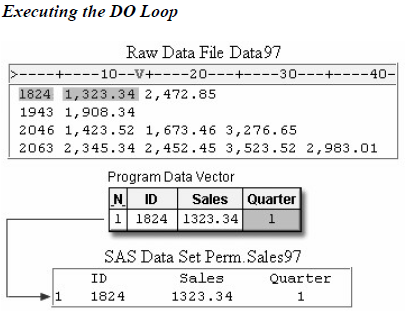
**INPUT** sales : **COMMA**. @;

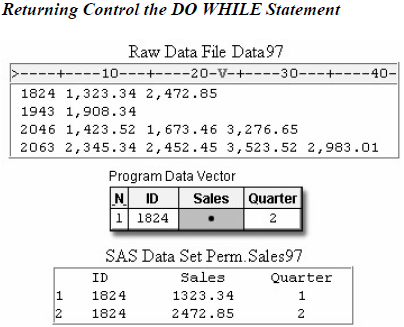
**END**;

**RUN**;

**PROC** **PRINT** data=perm.sales97;

**RUN**;

1. Sample Programs

* Repeating Blocks of Data

**LIBNAME** perm 'c:\records\weather';

**FILENAME** tempdata 'c:\records\weather\tempdata';

**DATA** perm.april90;

**INFILE** tempdata;

**INOUT** Date : **DATE**. HighTemp @@;

**FORMAT** date **DATE9**.;

**RUN**;

* An ID Field Followed by the Same Number of Repeating Fields

**LIBNAME** perm 'c:\records\sales';

**FILENAME** data97 'c:\records\sales\1997.dat';

**DATA** perm.sales97;

**INFILE** data97;

**INPUT** ID $ @;

**DO** Quarter=1 **TO** 4;

**INPUT** Sales : **COMMA**. @;

**OUTPUT**;

**END**;

**RUN**;

* An ID Field Followed by a Varying Number of Repeating Fields

**LIBNAME** perm 'c:\records\sales';

**FILENAME** data97 'c:\records\sales\1997.dat';

**DATA** perm.sales97;

**INFILE** data97 **MISSOVER**;

**INPUT** ID $ Sales : **COMMA**. @;

Quarter=0;

**DO** **WHILE** (sales ne .);

quarter+1;

**OUTPUT**;

**INPUT** sales : **COMMA**. @;

**END**;

**RUN**;

练习

1. A record that is being held by a single trailing at sign (@) is automatically released when
2. the input pointer moves past the end of the record.
3. the next iteration of the DATA step begins.
4. another INPUT statement that has an @ executes.
5. another value is read from the observation.
6. Which SAS program reads the values for ID and holds the record for each value of Quantity, so that three observations are created for each record?



1. **DATA** work.sales;

**INFILE** unitsold;

**INPUT** ID $;

**DO** week=1 **TO** 3;

**INPUT** Quantity : **COMMA**.;

**OUTPUT**;

**END**;

**RUN**;

1. **DATA** work.sales;

**INFILE** unitsold;

**INPUT** ID $ @@;

**DO** week=1 **TO** 3;

**INPUT** Quantity : **COMMA**.;

**OUTPUT**;

**END**;

**RUN**;

1. **DATA** work.sales;

**INFILE** unitsold;

**INPUT** ID $ @;

**DO** week=1 **TO** 3;

**INPUT** Quantity : **COMMA**.;

**OUTPUT**;

**END**;

**RUN**;

1. **DATA** work.sales;

**INFILE** unitsold;

**INPUT** ID $ @;

**DO** week=1 **TO** 3;

**INPUT** Quantity : **COMMA**. @;

**OUTPUT**;

**END**;

**RUN**;

1. Which program creates the SAS data set Perm.Topstore from the raw data file shown below?





1. **DATA** perm.topstores;

**INFILE** sales98 **MISSOVER**;

**INPUT** Store Sales : **COMMA**. @;

**DO WHILE** (sales ne .);

month + 1;

**OUTPUT**;

**INPUT** sales : **COMMA**. @;

**END**;

**RUN**;

1. **DATA** perm.topstores;

**INFILE** sales98 **MISSOVER**;

**INPUT** Store Sales : **COMMA**. @;

**DO WHILE** (sales ne .);

Month=0;

month + 1;

**OUTPUT**;

**INPUT** sales : **COMMA**. @;

**END**;

**RUN**;

1. **DATA** perm.topstores;

**INFILE** sales98 **MISSOVER**;

**INPUT** Store Sales : **COMMA**.

Month @;

**DO WHILE** (sales ne .);

month + 1;

**INPUT** sales : **COMMA**. @;

**END**;

**OUTPUT**;

**RUN**;

1. **DATA** perm.topstores;

**INFILE** sales98 **MISSOVER**;

**INPUT** Store Sales : **COMMA**. @;

Month=0;

**DO WHILE** (sales ne .);

month + 1;

**OUTPUT**;

**INPUT** sales : **COMMA**. @;

**END**;

**RUN**;

1. How many observations are produced by the DATA step that reads this external file?



**DATA** perm.choices;

**INFILE** icecream **MISSOVER**;

**INPUT** Day $ Flavor : $10. @;

**DO WHILE** (flavor ne ' ');

**OUTPUT**;

**INPUT** flavor : $10. @;

**END**;

**RUN**;

1. 3
2. 5
3. 12
4. 15