Understanding Data Step Processing using PDV

Mohamed Mehatab
Hewlett-Packard Canada

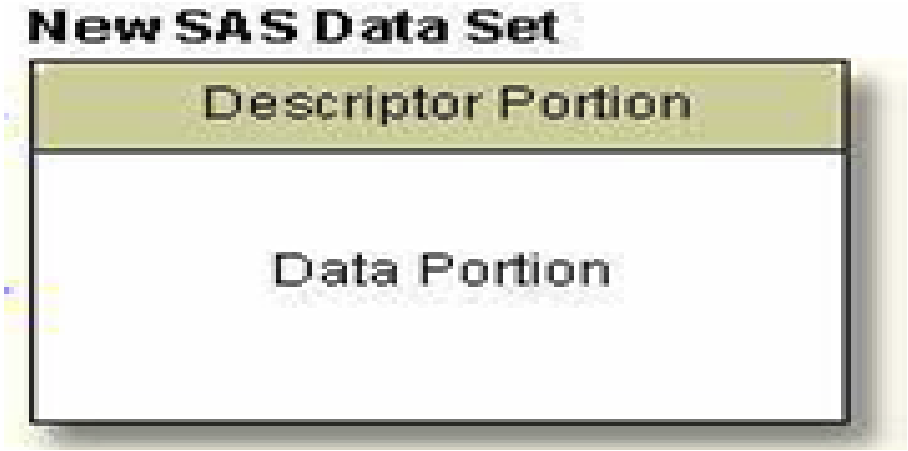
Introduction

Following are the key topics we will cover

- Data Step Processing
- PDV
- Drop/Keep SAS statements
- Drop/Keep dataset options
- Drop/Keep statements VS dataset options
- Where / If differences
- _Null_ SAS statements
- Understanding I/O and Dataset Size

- What is a SAS Data Set?

  A table, created in or for SAS, that SAS can recognize and knows how to process. It is usually created from datalines in one's code, or as the result of data extraction/manipulation from either a database, a SAS dataset, an external raw file or another program



- What is a SAS Data Step?

  A programming step used in SAS to perform data manipulation activities and as a result creates a SAS dataset

Data Step Processing

- Datastep processing consists of 2 phases.
  - Compilation Phase
  - Execution Phase

Compilation Phase

- During this phase, each of the statements within the data step are scanned for syntax errors.
- Descriptor portion of SAS dataset gets created at the end of compilation phase
- Following are the other 2 objects which get created at the end of compilation phase
  - Input Buffer
  - PDV

What is the PDV?

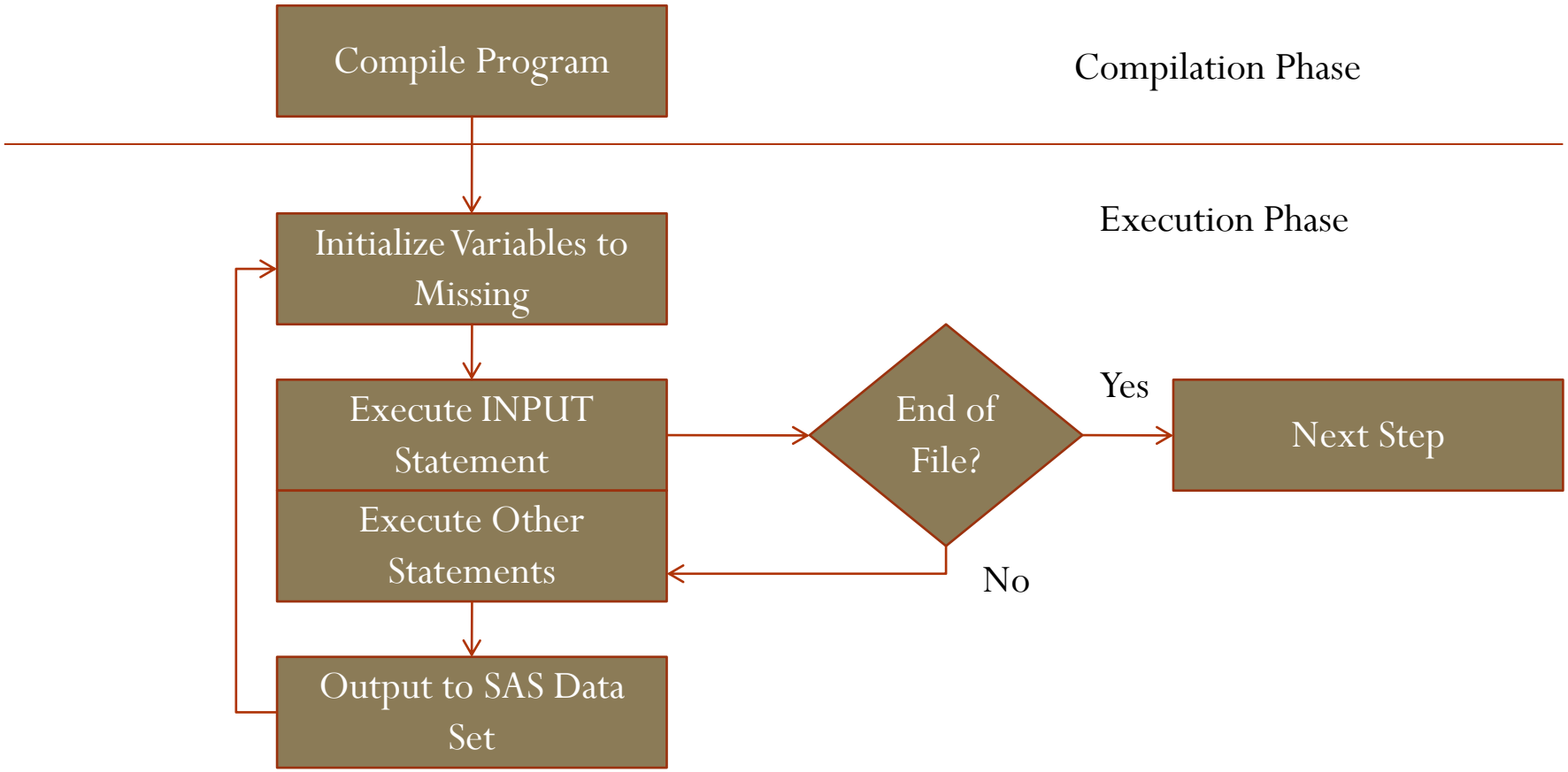
- The Program Data Vector is a logical area of memory that is created during the data step processing.
- SAS builds a SAS dataset by reading one observation at a time into the PDV and, unless given code to do otherwise, writes the observation to a target dataset.
- The program data vector contains two types of variables.
  - Permanent (data set and computed variables)
  - Temporary (automatic and option defined)

    Automatic (_N_ and _ERROR_)

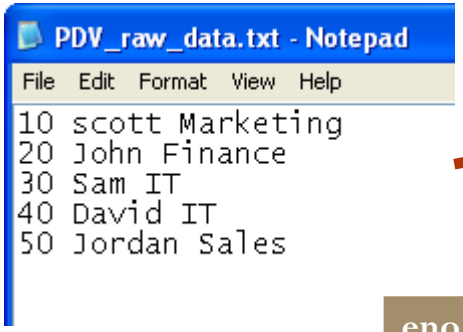
    Option defined (e.g., first.by-variable, last.by-variable, in=variable, end=variable)

Execution Phase

- During execution phase, a dataset's data portion is created.



Data Flow Process through PDV



Input Buffer

eno	ename	Dept	_N_	_ERROR_
10	scott	Marketing	1	0

PDV

eno	ename	Dept
10	Scott	Marketing
20	John	Finance
30	Sam	IT
40	David	IT
50	Jordan	Sales

SAS Dataset

Sample Program

```
115  data raw_data;
116       infile "C:\Documents and Settings\mohamed.mehatab\Desktop\Bell
116! Transition\MISC\TASS\PDV_raw_data.txt";
117       input eno ename $ Dept : $9.;
118       put _all_;
119  run;

NOTE: The infile "C:\Documents and Settings\mohamed.mehatab\Desktop\Bell
      Transition\MISC\TASS\PDV_raw_data.txt" is:

      Filename=C:\Documents and Settings\mohamed.mehatab\Desktop\Bell
      Transition\MISC\TASS\PDV_raw_data.txt,
      RECFM=V,LRECL=256,File Size (bytes)=76,
      Last Modified=November 17, 2012 22:13:23 o'c,
      Create Time=November 17, 2012 22:05:30 o'c

eno=10 ename=scott Dept=Marketing _ERROR_=0 _N_=1
eno=20 ename=John Dept=Finance _ERROR_=0 _N_=2
eno=30 ename=Sam Dept=IT _ERROR_=0 _N_=3
eno=40 ename=David Dept=IT _ERROR_=0 _N_=4
eno=50 ename=Jordan Dept=Sales _ERROR_=0 _N_=5
NOTE: 5 records were read from the infile "C:\Documents and Settings\mohamed.mehatab\Desktop\Bell
      Transition\MISC\TASS\PDV_raw_data.txt".
      The minimum record length was 9.
      The maximum record length was 18.
NOTE: The data set WORK.RAW_DATA has 5 observations and 3 variables.
NOTE: DATA statement used (Total process time):
      real time           0.15 seconds
      cpu time            0.00 seconds
```

Using different SAS statements to perform following actions

- Selecting Variables
  - Drop / Keep
- Selecting Observations
  - Where / If

Drop/Keep Statement

- Drop Statement
  - It Indicates which variables have to be dropped from output datasets
  - It applies to all the output datasets in a datastep
  - All variables listed on the DROP statement are on the program data vector and available for use in the current data step until the observation is output.
- Keep Statement
  - It Indicates which variables have to be dropped from output datasets
  - Variables **not** listed in the KEEP statement are on the program data vector and available for use in the current data step until the observation is output.

Drop/Keep Data set options

- Drop Data set Option
  - The DROP= data set option used on an input data set specifies the variables not to be read from the data set to the program data vector (on the way in)
  - The DROP= data set option used with an output data set it lists the variables on the program data vector that are not to be written to the output data set (on the way out).
- Keep Data set Option
  - The KEEP= data set option used on an input data set lists those variables to be read from the data set to the program data vector (on the way in).
  - The KEEP= data set options used with an output data set it specifies variables to be written from the program data vector to the output data set (on the way out).

Drop/Keep Statements Vs Drop/Keep Dataset Options

- Drop/Keep statements only affect the output datasets but Drop/Keep data set options affect output as well as input datasets
- Drop/Keep statements applies to all the output datasets but Drop/Keep data set options can be used to drop/ keep only on the specific output datasets
- Drop/Keep statements or Drop/Keep options, when used together, drop will take effect first

Where/IF Statement Differences

Where	IF
Where statement selects the observations before they are read into PDV	If statement selects the observations from the PDV
When the WHERE statement is used with a BY statement, the WHERE statement is executed first.	When a subsetting IF is used with a BY statement, the BY statement is processed first.
The WHERE statement selects observations in SAS data sets only so the variables created in a datastep cannot be used with Where	IF statement can select observations based on the variables created in a datastep
Where statement cannot be used to select records from external file that contains raw data	If statement can be used to select records from external file through PDV
Where statement improves the efficiency of SAS programs because SAS is not required to read all the observations from i/p dataset	

```
data where_test;
    set sashelp.cars;
    where make='Acura';
run;


data if_test;
    set sashelp.cars;
    if make='Acura';
run;
```

```
37
38   data where_test;
39       set sashelp.cars;
40       where make='Acura';
41   run;

NOTE: There were 7 observations read from the data set SASHELP.CARS.
      WHERE make='Acura';
NOTE: The data set WORK.WHERE_TEST has 7 observations and 15 variables.
NOTE: DATA statement used (Total process time):
      real time           0.01 seconds
      cpu time            0.00 seconds


42
43
44   data if_test;
45       set sashelp.cars;
46       if make='Acura';
47   run;

NOTE: There were 428 observations read from the data set SASHELP.CARS.
NOTE: The data set WORK.IF_TEST has 7 observations and 15 variables.
NOTE: DATA statement used (Total process time):
      real time           0.03 seconds
      cpu time            0.00 seconds
```

Accessing PDV variables in a where statement

```
78   data where_by;
79   set sashelp.cars;
80   by make;
81   where first.make;
           ----------
           180
ERROR: Syntax error while parsing WHERE clause.
ERROR 180-322: Statement is not valid or it is used out of proper order.

82   run;

NOTE: The SAS System stopped processing this step because of errors.
WARNING: The data set WORK.WHERE_BY may be incomplete.  When this step was stopped there were 0
         observations and 15 variables.
WARNING: Data set WORK.WHERE_BY was not replaced because this step was stopped.
NOTE: DATA statement used (Total process time):
      real time           0.03 seconds
      cpu time            0.03 seconds
```

Subsetting raw data using where statement

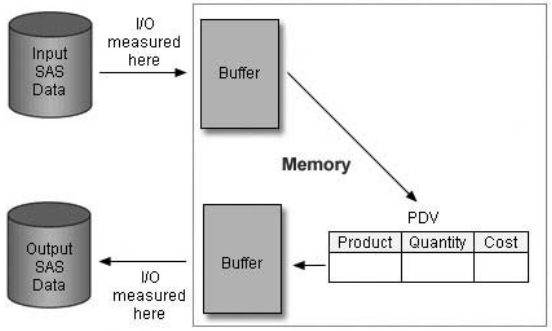
```
133  data raw_data;
134       infile "C:\Documents and Settings\mohamed.mehatab\Desktop\Bell
134! Transition\MISC\TASS\PDV_raw_data.txt";
135       input eno ename $ Dept : $9.;
136       put _all_;
137       where eno=10;
ERROR: No input data sets available for WHERE statement.
138  run;

NOTE: The SAS System stopped processing this step because of errors.
WARNING: The data set WORK.RAW_DATA may be incomplete.  When this step was stopped there were 0
         observations and 3 variables.
WARNING: Data set WORK.RAW_DATA was not replaced because this step was stopped.
NOTE: DATA statement used (Total process time):
      real time           0.03 seconds
      cpu time            0.00 seconds
```

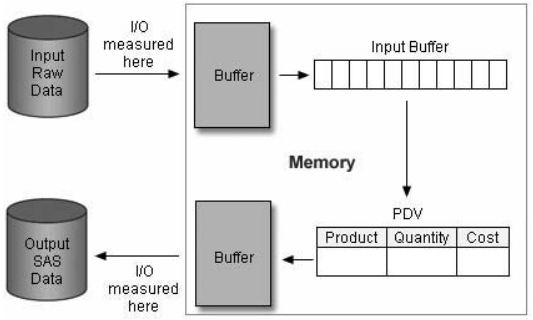
NULL SAS Statement

- This SAS statement is used to compile and execute data step without creating an output dataset.
- This statement is used for debugging datasteps.
- It is used to write data into external files from a SAS dataset
- It is also used to create macro variables containing values from the dataset variables using CALL SYMPUT statement
- Put statement along with _null_ is used to debug any programming / data errors

Understanding I/O

- I/O is a measurement of the read and write operations that are performed as data and programs are copied from a storage device to memory (input) or from memory to a storage or display device (output).





Understanding I/O

- The amount of data that can be transferred to one buffer in a single I/O operation is referred to as Page Size. Commonly referred a Buffer Size.(BUFSIZE)
- Larger page size can reduce execution time by reducing the number of times SAS has to read from or write to the storage medium. However, the improvement in execution time comes at the cost of increased memory consumption

Dataset Size

```
proc contents data=sashelp.cars;
run;
```

Dataset Size(bytes) = pagesize*number of pages

Engine/Host Dependent Information	
Data Set Page Size	16384
Number of Data Set Pages	5
First Data Page	1
Max Obs per Page	107
Obs in First Data Page	82
Number of Data Set Repairs	0
Filename	/apps/sas/SASFoundation/9.2/sashelp/cars.sas7bdat
Release Created	9.0202M0
Host Created	SunOS

References

- **The Use and Abuse of the Program Data Vector**
  Jim Johnson, Ephicacy Corporation, North Wales, PA, USA

  http://support.sas.com/resources/papers/proceedings12/255-2012.pdf

Thank You