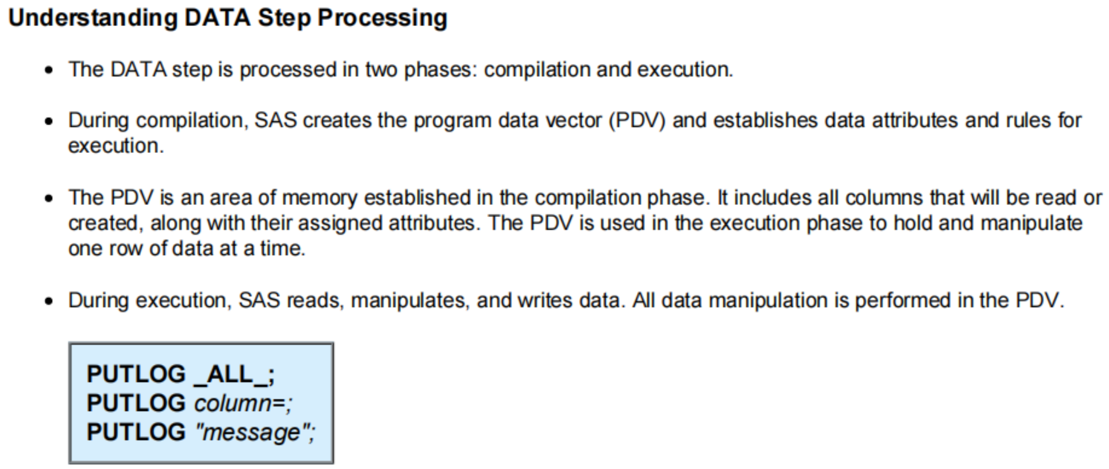
1.dataProcess

March 3, 2025

0.0.1 Data Step Processing

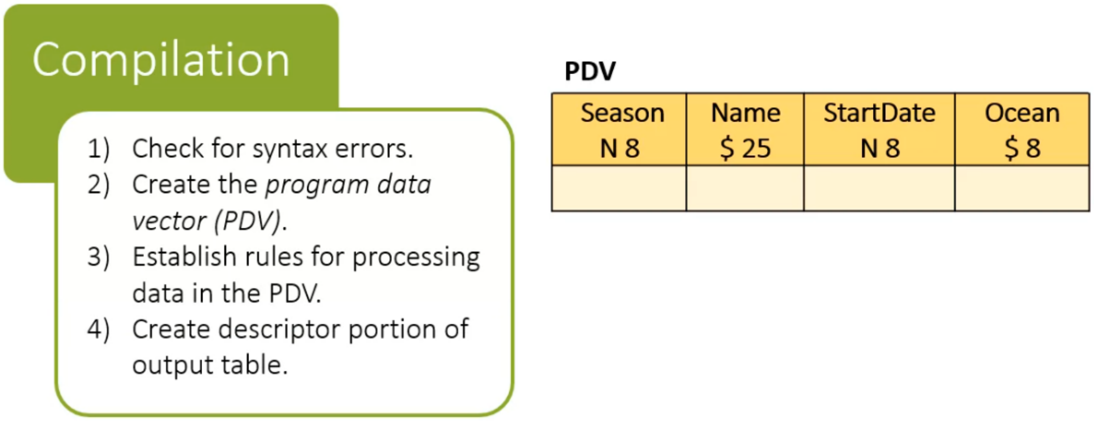
In Controlling Data Step Processing we dig deeper into the DATA step. You learn how the DATA step processes data behind the scenes. Then you use this knowledge to control when and where the DATA step outputs rows to new tables.



Understanding DATA Step Processing

- The DATA step is processed in two phases: compilation and execution.
- During compilation, SAS creates the program data vector (PDV) and establishes data attributes and rules for execution.
- The PDV is an area of memory established in the compilation phase. It includes all columns that will be read or created, along with their assigned attributes. The PDV is used in the execution phase to hold and manipulate one row of data at a time.
- During execution, SAS reads, manipulates, and writes data. All data manipulation is performed in the PDV.

```
PUTLOG _ALL_;
PUTLOG column=;
PUTLOG "message";
```

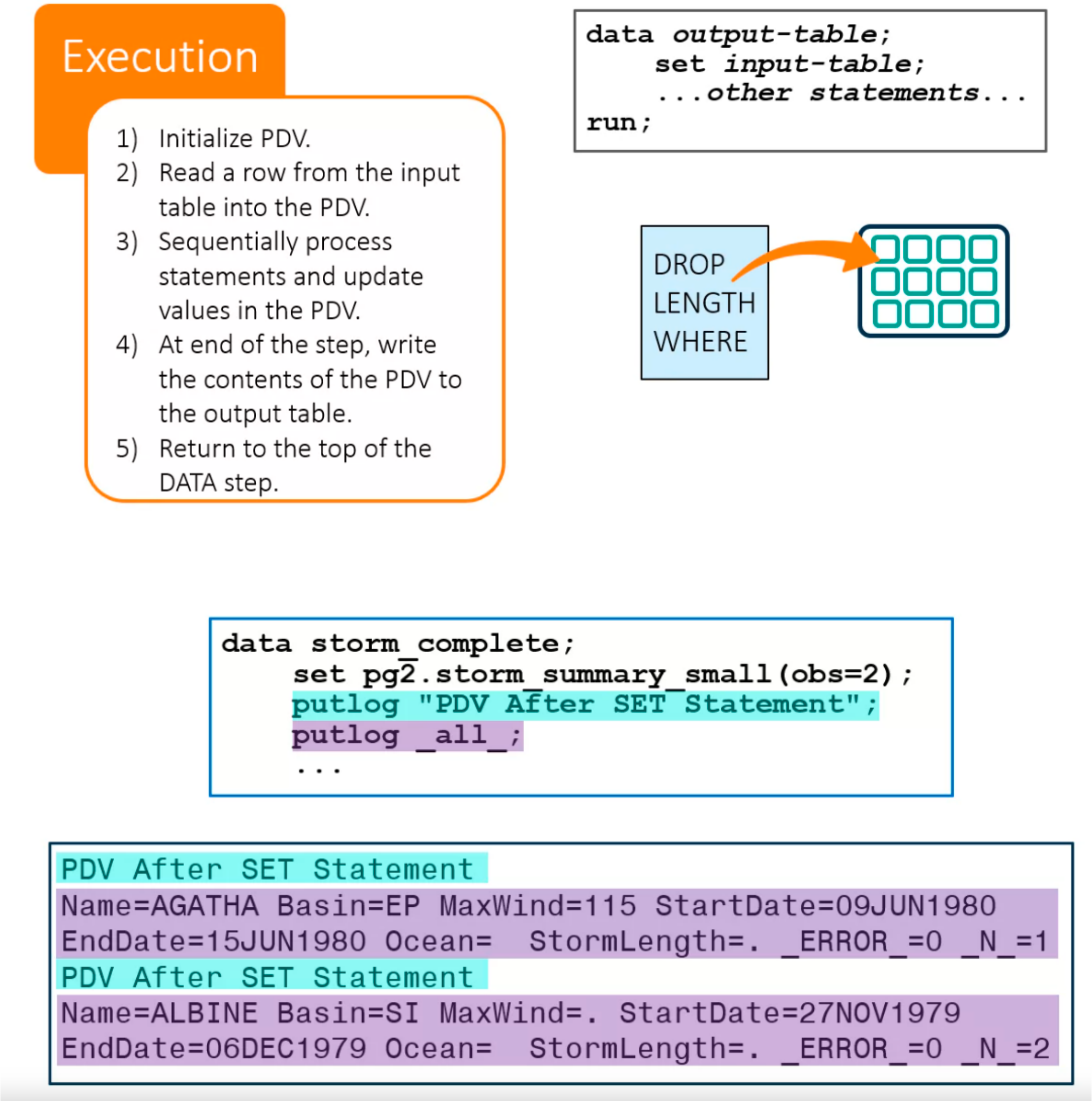


Compilation

1) Check for syntax errors.
2) Create the *program data vector (PDV).*
3) Establish rules for processing data in the PDV.
4) Create descriptor portion of output table.

PDV

Season N 8	Name $ 25	StartDate N 8	Ocean $ 8



Execution

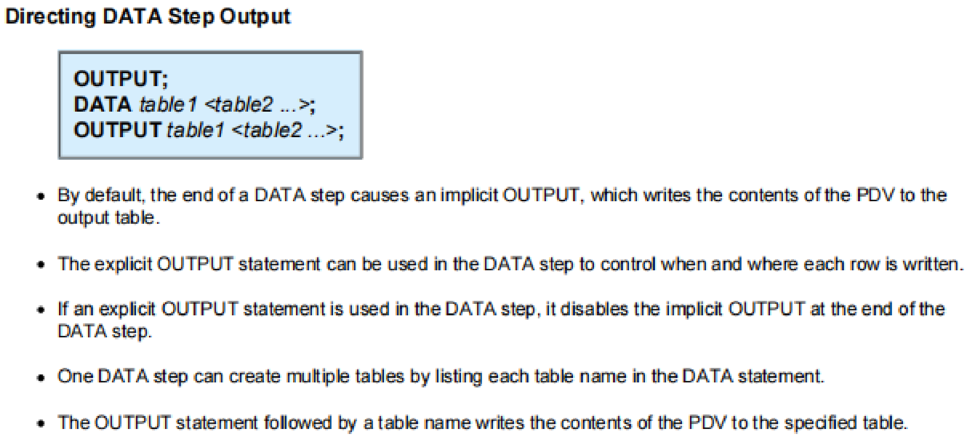
1) Initialize PDV.
2) Read a row from the input table into the PDV.
3) Sequentially process statements and update values in the PDV.
4) At end of the step, write the contents of the PDV to the output table.
5) Return to the top of the DATA step.

```
data output-table;
     set input-table;
     ...other statements...
run;
```

```
data storm_complete;
     set pg2.storm_summary_small(obs=2);
     putlog "PDV After SET Statement";
     putlog _all_;
     ...
```

```
PDV After SET Statement
Name=AGATHA Basin=EP MaxWind=115 StartDate=09JUN1980
EndDate=15JUN1980 Ocean=  StormLength=. _ERROR_=0 _N_=1
PDV After SET Statement
Name=ALBINE Basin=SI MaxWind=. StartDate=27NOV1979
EndDate=06DEC1979 Ocean=  StormLength=. _ERROR_=0 _N_=2
```

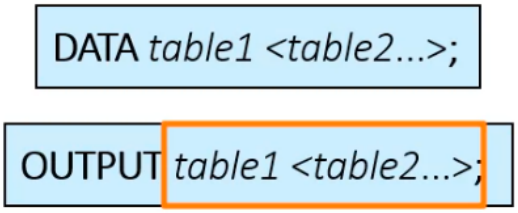
0.0.2 Output Control



Directing DATA Step Output

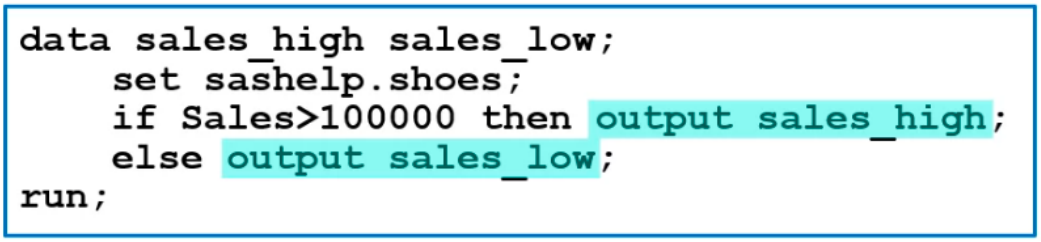
```
OUTPUT;
DATA table1 <table2 ...>;
OUTPUT table1 <table2 ...>;
```

- By default, the end of a DATA step causes an implicit OUTPUT, which writes the contents of the PDV to the output table.
- The explicit OUTPUT statement can be used in the DATA step to control when and where each row is written.
- If an explicit OUTPUT statement is used in the DATA step, it disables the implicit OUTPUT at the end of the DATA step.
- One DATA step can create multiple tables by listing each table name in the DATA statement.
- The OUTPUT statement followed by a table name writes the contents of the PDV to the specified table.



```
DATA table1 <table2...>;
```

```
OUTPUT table1 <table2...>;
```



```
data sales_high sales_low;
    set sashelp.shoes;
    if Sales>100000 then output sales_high;
    else output sales_low;
run;
```

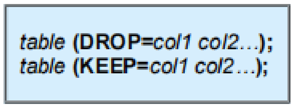
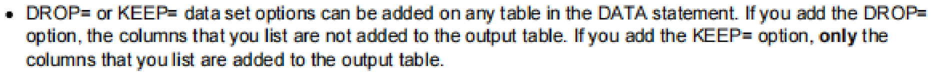
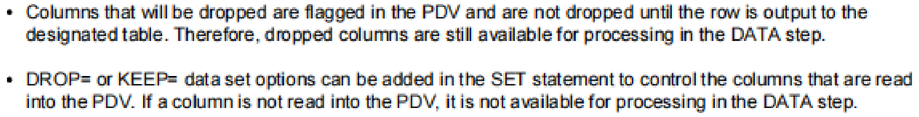


table **(DROP=***col1 col2…***);**
table **(KEEP=***col1 col2…***);**



- DROP= or KEEP= data set options can be added on any table in the DATA statement. If you add the DROP= option, the columns that you list are not added to the output table. If you add the KEEP= option, **only** the columns that you list are added to the output table.



- Columns that will be dropped are flagged in the PDV and are not dropped until the row is output to the designated table. Therefore, dropped columns are still available for processing in the DATA step.
- DROP= or KEEP= data set options can be added in the SET statement to control the columns that are read into the PDV. If a column is not read into the PDV, it is not available for processing in the DATA step.

[ ]: