

**PHUSE EU Connect
2025 – CT02**

How Does a Computer *Really* Process Your SAS Code?

Ben Barnaby-Pass, Phastar



Overview

- 1. Timing Code**
- 2. Keep/Drop Statements**
- 3. If/Where Statements**
- 4. Index Statements**
- 5. Conclusion**

Timing Code

Breakdown of the Timing Code Being Used

Code For Timing Code

- **Shows time running program**
- **Amount of memory it takes**
- **Number of steps taken in program**

```
options FULLTIMER;
```

Breakdown

NOTE: There were 8712200 observations read from the data set WORK.ADLB.
NOTE: The data set WORK.ADLB2 has 8712200 observations and 20 variables.
NOTE: Compressing data set WORK.ADLB2 decreased size by 69.02 percent.
Compressed is 29657 pages; un-compressed would require 95739 pages.
NOTE: DATA statement used (Total process time):

real time	18.42 seconds
user cpu time	15.73 seconds
system cpu time	2.67 seconds
memory	1601.87k
OS Memory	32244.00k
Timestamp	10/06/2025 04:32:36 PM
Step Count	56
Switch Count	29

Important Information

- **Real time – Time spent to process**
- **User CPU time – Time spent in user program**
- **System CPU time – Time spent to perform OS tasks**

NOTE: DATA statement used (Total process time):

real time	18.42 seconds
user cpu time	15.73 seconds
system cpu time	2.67 seconds

Process Of Checking

- 1. Run a program of ~9million observations**
- 2. Use this to then run the given statements**
- 3. Compare results to see which is 'quicker'**

Keep/Drop Statements

Comparing between and finding the difference in efficiency

Keep Statement (Data)

- **Keeping on data line**
- **Keeping 20 variables**

```
data ad1b2(keep = STUDYID -- ACTARMN);  
    set ad1b;  
run;
```

Keep Statement (Set)

- **Simple keep statement**
- **Keeping on set statement**

```
data ad1b2;  
  set ad1b(keep = STUDYID -- ACTARMN);  
run;
```

Timing Difference

NOTE: DATA statement used (Total process time):

real time	18.34 seconds
user cpu time	15.90 seconds
system cpu time	2.43 seconds

Keep (Data)

Keep (Set)

NOTE: DATA statement used (Total process time):

real time	17.94 seconds
user cpu time	15.23 seconds
system cpu time	2.70 seconds

Drop Statement (Data)

- On the data statement
- Dropping all but 20 variables

```
data ad1b2(drop = TRT01P -- GLOBALFL);  
  set ad1b;  
run;
```

Drop Statement (Set)

- **On the set statement**
- **Dropping all but 20 variables**

```
data adlb2;  
  set adlb(drop = TRT01P -- GLOBALFL);  
run;
```

Timing Difference

NOTE: DATA statement used (Total process time):

real time	18.25 seconds
user cpu time	15.56 seconds
system cpu time	2.68 seconds

Drop (Data)

Drop (Set)

NOTE: DATA statement used (Total process time):

real time	18.27 seconds
user cpu time	15.46 seconds
system cpu time	2.78 seconds

Timing Difference (Overall)

NOTE: DATA statement used (Total process time):

real time	18.34 seconds
user cpu time	15.90 seconds
system cpu time	2.43 seconds

Keep (Data)

NOTE: DATA statement used (Total process time):

real time	17.94 seconds
user cpu time	15.23 seconds
system cpu time	2.70 seconds

Keep (Set)

NOTE: DATA statement used (Total process time):

real time	18.25 seconds
user cpu time	15.56 seconds
system cpu time	2.68 seconds

Drop (Data)

NOTE: DATA statement used (Total process time):

real time	18.27 seconds
user cpu time	15.46 seconds
system cpu time	2.78 seconds

Drop (Set)

If/Where Statements

Finding which is more efficient

If Statement

- **Very basic if statement**
- **Outputting 1 arm of a treatment**

```
data adlb2;  
  set adlb;  
  if trtan = 1 then output;  
run;
```

Where Statement

- **Basic where statement**
- **Same desired output as if statement**

```
data ad1b2;  
  set ad1b;  
  where trtan = 1;  
run;
```

Timing Difference

NOTE: DATA statement used (Total process time):

real time	12.80 seconds
user cpu time	10.43 seconds
system cpu time	2.39 seconds

If statement

Where statement

NOTE: DATA statement used (Total process time):

real time	11.72 seconds
user cpu time	10.15 seconds
system cpu time	1.56 seconds

Index Statements

**Seeing if Index
statements are time
efficient**

Index Statement

- **Basic Index Statement**
- **Want to find where a certain string shows up in an observation/variable**

```
data adlb2;  
  set adlb;  
  POSITION = index(APHASE, "Follow-up");  
run;
```

Timing

- **Time spent analysing every observation**
- **Index scans must read ALL the variable data which takes time!**

```
NOTE: DATA statement used (Total process time):  
      real time                38.80 seconds  
      user cpu time            33.23 seconds  
      system cpu time          5.54 seconds
```

Conclusion

Keep/Drop Statements

- **Be smart with usage**
- **Coding may be easier for one or the other**
- **Realistically doesn't matter which you use...**

If/Where Statements

- **Where is 100% more efficient**
- **Both statements have their uses**
- **Where tells SAS to not read all observations**
- **If statements execute only when a condition is satisfied**

Index Statements

- **Index statements take time to process the data**
- **They are efficient, but only really have their uses**
- **Overall, I would rather use a scan statement**

Thank You

Ben Barnaby-Pass

Phastar

WFH, UK

ben.pass@phastar.com



Q+A

Any Questions?