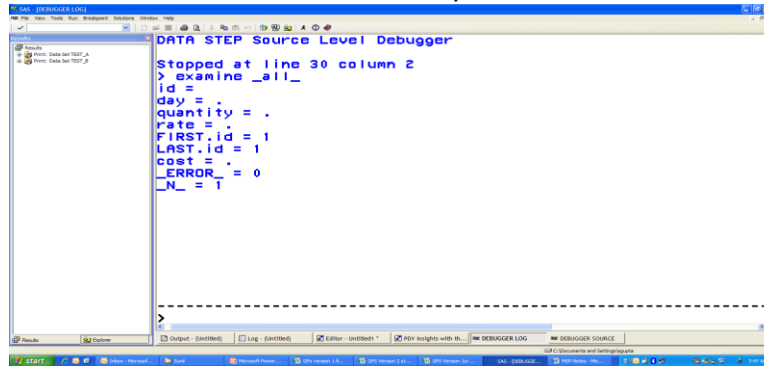


PDV Insights with the DATA Step Debugger: Debugger Log Window Commands

Model Syntax: data many_to_one/ debug; (SAS statements); run; (QUIT to end debugger session)

1. Display Current Variable Values

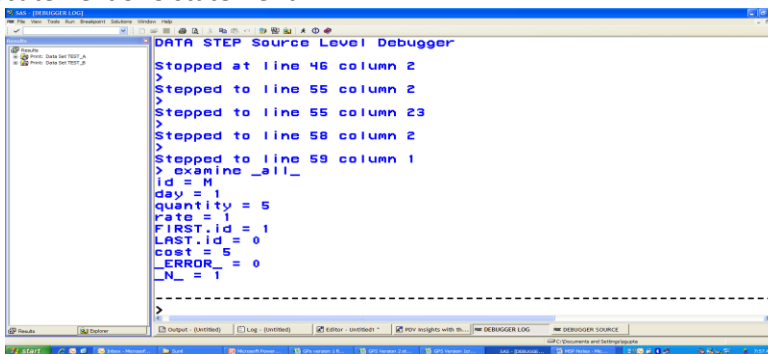
- ✓ **EXAMINE <_ALL_ or variable>** to display current values before the highlighted SAS statement is executed or the result after the previous SAS statement. **WATCH <variable>** to identify line and value of variable change.



```
DATA STEP Source Level Debugger
Stopped at line 30 column 2
> examine_all_
id = .
day = .
quantity = .
rate = .
FIRST.id = 1
LAST.id = 1
cost = .
_ERROR_ = 0
_N_ = 1
>
```

2. Set SAS Statement to Execute

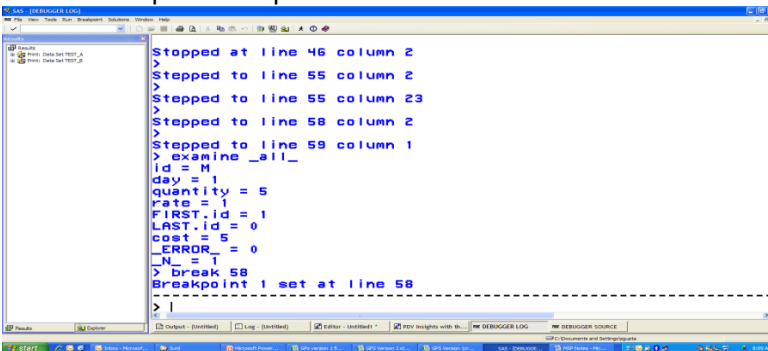
- ✓ **STEP** or return key to execute next SAS statement.



```
DATA STEP Source Level Debugger
Stopped at line 46 column 2
> Stepped to line 55 column 2
> Stepped to line 55 column 23
> Stepped to line 58 column 2
> Stepped to line 59 column 1
> examine_all_
id = M
day = 1
quantity = 5
rate = 1
FIRST.id = 1
LAST.id = 0
cost = 5
_ERROR_ = 0
_N_ = 1
>
```

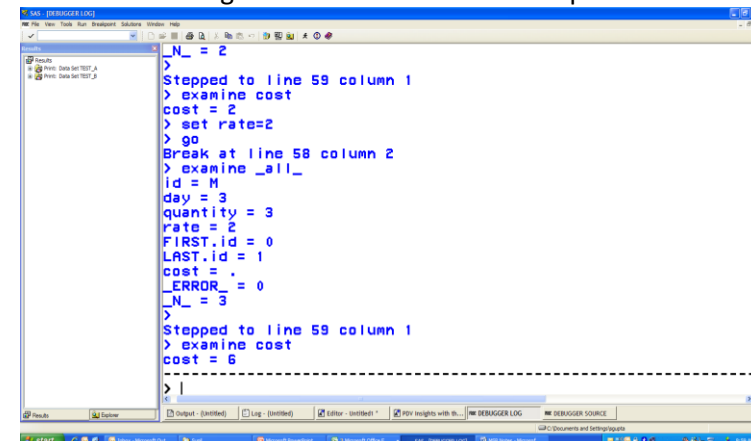
3. Set Variable Values for Testing Purpose

- ✓ **BREAK <line number>** to set execute up to line number in the DATA step.
- ✓ **GO** to continue execution of DATA step to breakpoint line number.



```
DATA STEP Source Level Debugger
Stopped at line 46 column 2
> Stepped to line 55 column 2
> Stepped to line 55 column 23
> Stepped to line 58 column 2
> Stepped to line 59 column 1
> examine_all_
id = M
day = 1
quantity = 5
rate = 1
FIRST.id = 1
LAST.id = 0
cost = 5
_ERROR_ = 0
_N_ = 1
> Break 58
Breakpoint 1 set at line 58
> |
>
```

- ✓ **SET <variable> = <variable or value>** to assign a value to a variable to help resolve issue.



```
DATA STEP Source Level Debugger
_N_ = 2
> Stepped to line 59 column 1
> examine cost
cost = 2
> set rate=2
> go
Break at line 58 column 2
> examine_all_
id = M
day = 3
quantity = 3
rate = 2
FIRST.id = 0
LAST.id = 1
cost = .
_ERROR_ = 0
_N_ = 3
> Stepped to line 59 column 1
> examine cost
cost = 5
> |
>
```