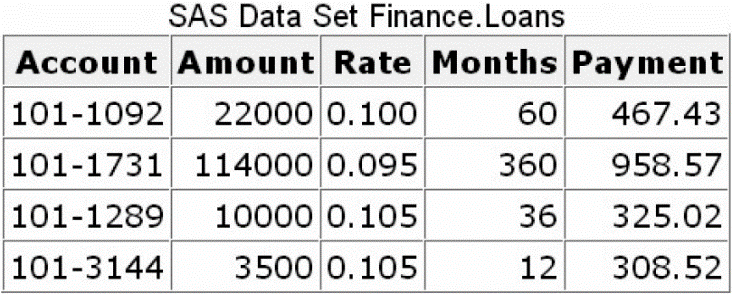
**Chapter XI & Chapter VI附加 - Understanding How Data Sets Are Read**

Example:

**DATA** finance.duejan;

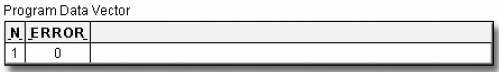
**SET** finance.loans;

Interest=amount\*(rate/12);

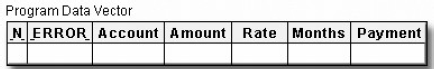
**RUN**;

**Compilation Phase**

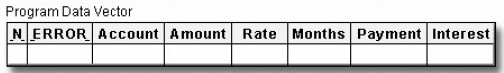
1. The program data vector is created and contains the automatic variables \_N\_ and \_ERROR\_.



1. SAS also scans each statement in the DATA step, looking for syntax errors.
2. When the SET statement is compiled, a slot is added to the program data vector for each variable in the input data set. The input data set supplies the variable names, as well as attributes such as type and length.



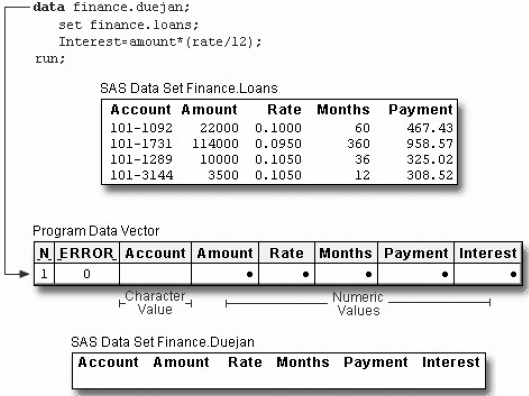
1. Any variables that are created in the DATA step are also added to the program data vector. The attributes of each of these variables are determined by the expression in the statement.



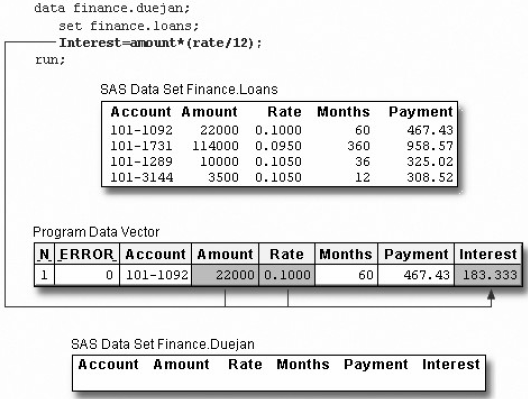
1. At the bottom of the DATA step, the compilation phase is complete, and the descriptor portion of the new SAS data set is created. There are no observations because the DATA step has not yet executed.

**Execution Phase**

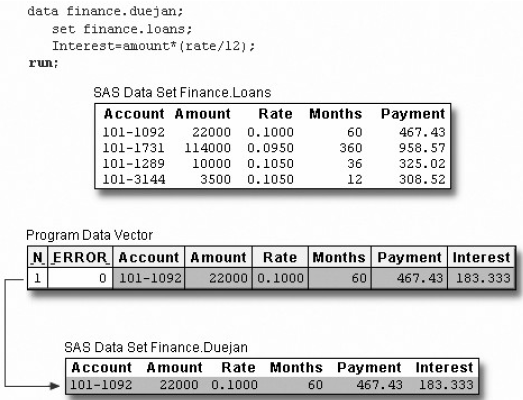
1. The DATA step executes once for each observation in the input data set. For example, this DATA step will execute four times because there are four observations in the input data set Finance.Loans.
2. At the beginning of the execution phase, the value of \_N\_ is 1. Because there are no data errors, the value of \_ERROR\_ is 0. The remaining variables are initialized to missing. Missing numeric values are represented by a period, and missing character values are represented by a blank.



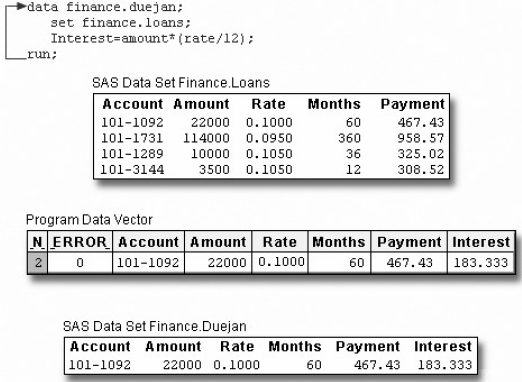
1. The SET statement reads the first observation from the input data set into the program data vector.
2. Then, the assignment statement executes to compute the value for Interest.

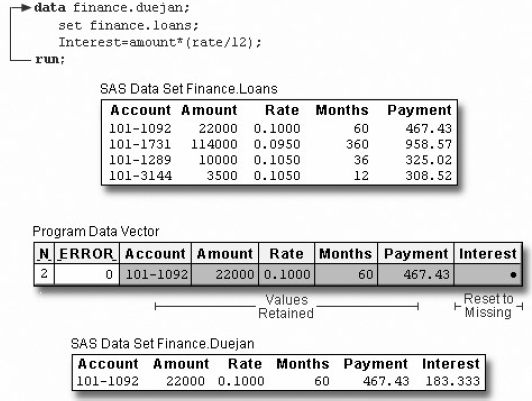


1. At the end of the first iteration of the DATA step, the values in the program data vector are written to the new data set as the first observation



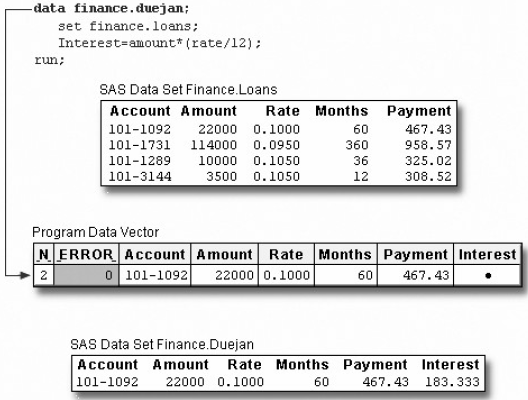
1. The value of \_N\_ increments from 1 to 2, and control returns to the top of the DATA step. Remember, the automatic variable \_N\_ keeps track of how many times the DATA step has begun to execute.



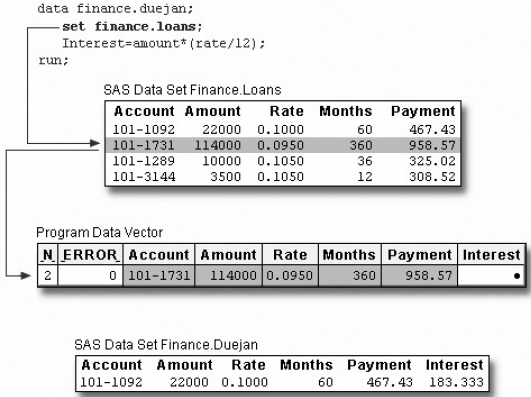
1. SAS retains the values of variables that were read from a SAS data set with the SET statement, or that were created by a sum statement. All other variable values, such as the values of the variable Interest, are set to missing.

* When SAS reads raw data, the situation is different. In that case, SAS sets the value of each variable in the DATA step to missing at the beginning of each iteration, with these exceptions:
* variables named in a RETAIN statement
* variables created in a sum statement
* data elements in a \_TEMPORARY\_ array
* any variables created by using options in the FILE or INFILE statements
* automatic variables.

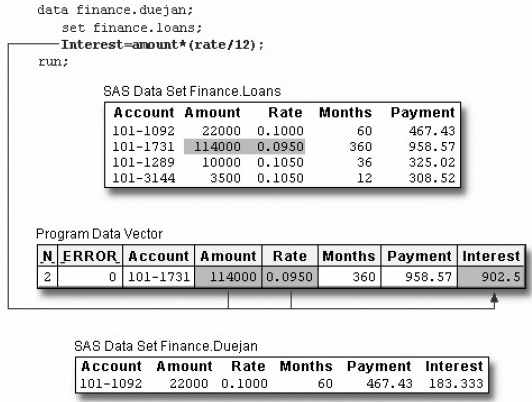
1. At the beginning of the second iteration, the value of \_ERROR\_ is reset to 0.



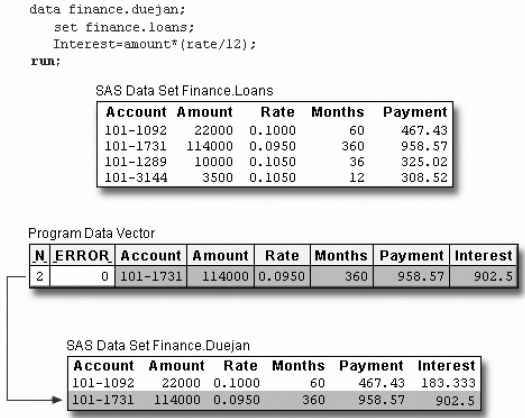
1. As the SET statement executes, the values from the second observation are read into the program data vector.



1. The assignment statement executes again to compute the value for Interest for the second observation



1. At the bottom of the DATA step, the values in the program data vector are written to the data set as the second observation.



1. The value of \_N\_ increments from 2 to 3, and control returns to the top of the DATA step. SAS retains the values of variables that were read from a SAS data set with the SET statement, or that were created by a sum statement. All other variable values, such as the values of the variable Interest, are set to missing.
2. This process continues until all of the observations are read.

