# Using the KEY= Option for Lookup Tables

**Zeph Stemle, Qualex Consulting Services, Inc.**
**Martinsville, VA**

Many of us work on projects where we have the need to reference lookup tables. These lookup tables can contain code descriptions, addresses, part information, or any other detail about key items in a master table.

For the most part, a DATA step using MERGE statements or PROC SQL joins are adequate. However, when the master dataset starts to get large, two problems start to occur. The time to perform the lookup operations can get to be quite long (as well as any PROC SORT operations performed before a merge), and the amount of space required in the WORK library can be as much as three times larger than the size of the master dataset.

You can use the KEY= option with one or more SET statements to perform lookup operations. This method is normally much faster than a DATA step that uses MERGE statements and several times faster than PROC SQL. Additionally, less WORK space is required when you use this method.

## Requirements

There are some requirements that must be met when you use the KEY= option:

- All lookup tables must contain unique keys. However, keys can be comprised of a combination of variables.
- Lookup tables must be indexed by the key variables. This can be either a simple or composite index.
- You must use the KEEP= dataset option to specify the variables to keep from each lookup table. You must also keep the key variables.
- If any key values exist in the master dataset that do not exist in the lookup tables, additional code must be included to prevent lookup values from being retained from the previous observation.
- If any key values repeat in successive observations in the master data set, the UNIQUE option must be used. The UNIQUE forces SAS to begin at the top of the lookup table each time it does a lookup operation.

## SAS Code

### Create Indexes for Lookup Tables

You must create an index for all lookup tables. You can create a simple index if one variable is used to form the key. If multiple variables are used for the key, a composite index must be created. To prevent errors in the SAS Log, check for the existence of the index from previous runs before you create it.

The following code example creates a simple index in one lookup table, and a composite index in another:

```
proc datasets lib = tables
nolist;

  modify buyer;

  index create buyer_code; /*
Simple index */

  modify parts;

  index create partinfo =
(part_num plant_code); /*
Composite index */

run;

quit;
```

### Perform the Lookup Operation

Once the indexes are created, the lookup operation can be performed. No sorting is necessary because of the indexing. The following example shows the code for performing the lookup operations.

```
data mylib.after;
```

```
 set mylib.master; /* Master
dataset */
 if buyer_code ^= ' ' then /* Do
lookup only if key in master */
   set tables.buyer(keep=
      BUYER_CODE /* key variable
*/
      BUYER_NAME
      BUYER_DEPT
      BUYER_PLANT)
    key = buyer_code/unique; /*
Start at the top of the table */
 select (_iorc_); /* Look at
return code to see if successful
*/
  when(%sysrc(_dsenom)) do; /* If
key not found */
      BUYER_NAME = ' ';
      BUYER_DEPT = ' ';
      BUYER_PLANT = ' ';
  end;
  otherwise;
  end;
 if part_num ^= ' ' and
    plant_code ^= ' ' then
   set tables.parts(keep=
      PART_NUM
      PLANT_CODE
      PART_DESCRIPTION
      VENDOR
      COST)
    key = partinfo/unique; /*
Composite index used */
select (_iorc_); /* Look at return
code to see if successful */
  when(%sysrc(_dsenom)) do; /* If
key not found */
      PART_DESCRIPTION = ' ';
      VENDOR = ' ';
      COST = ' ';
  end;
  otherwise;
  end;
run;
```

## Conclusion

Using the KEY= of the SET statements to perform lookups can be a major performance enhancement. This option requires SAS to only access the observations it needs. You do have to be careful when using return codes and dealing with key variables. However, the benefits far outweigh the risks.

## Contact Information

Your comments and questions are valued and encouraged. Contact the author at:


Zeph Stemle

Qualex Consulting Services Inc.
231 East Church Street
5th Floor
Martinsville, Virginia, USA  24112

E-mail: zeph.stemple@qlx.com
Phone: 877-887-4SAS (4727)