As a SAS Certified Professional and SAS Institute Alliance Member (1996 – 2002), Kirk provides IT consulting services and training to SAS users around the world. He is the author of four SAS books and more than two hundred peer-reviewed papers and articles that have appeared in professional journals and SAS User Group proceedings. He has also been an Invited speaker at more than two hundred SAS International, regional, local, and special-interest user group conferences and meetings throughout North America. His popular SAS Tips column, "Kirk's Korner of Quick and Simple Tips", appears regularly in several SAS User Group newsletters and Web sites, and his fun-filled SASword Puzzles is featured in SAScommunity.org.

Kirk's Korner: Quick and Simple Tips

PROC SQL Join Algorithms and the _METHOD Option

When it comes to performing PROC SQL joins, users supply the list of tables for joining along with the join conditions, and the PROC SQL optimizer has the task of determining which of the available join algorithms to use for performing the join operation. There are three basic algorithms used in joining:

Nested Loop Join – When an equality condition is not specified, a read of the complete contents of the right table is processed for each row in the left table.

Merge Join – When the tables specified are already in the desired sort order, resources will not need to be extended to rearranging the tables.

Hash Join – When an equality relationship exists, the smaller of the tables is able to fit in memory, no sort operations are required, and each table is read only once.

Under Version 8.2 or later, PROC SQL supports a powerful "undocumented" option called **_METHOD**. Although undocumented features like the _METHOD option should be used with caution, SAS users may find this option to provide far greater value than risk. In fact, the _METHOD option is worth exploring because the benefits associated with gaining a better understanding associated with what happens during specific PROC SQL processes, including joins.

The various codes and their corresponding descriptions associated with the _METHOD option appear in the table below.

_METHOD Option Codes and Descriptions

Code	Description
SQXCRTA	Create table as Select.
SQXSLCT	Select statement or clause.
SQXJSL	Step loop join (Cartesian).
SQXJM	Merge join operation.
SQXJNDX	Index join operation.
SQXJHSH	Hash join operation.
SQXSORT	Sort operation.
SQXSRC	Source rows from table.
SQXFIL	Rows filtration.
SQXSUMG	Summary stats (aggregates) with GROUP BY clause.
SQXSUMN	Summary stats with no GROUP BY clause.

The following PROC SQL code example, and corresponding SAS Log, illustrates the results from running the **_METHOD** option with a simple two-way equi-join. The _METHOD option displays information that can help users better understand, as well as tune and debug their join queries. The SAS Log illustrates that the two-source tables are MOVIES and ACTORS, and the join algorithm used by the PROC SQL optimizer is a hash join. Because a hash join utilizes available real memory to perform the join, it is often faster than a merge or index-join operation because of the speed of real memory as well as no sort operation on the source tables being required.

PROC SQL Code:

```
PROC SQL  _METHOD;
  SELECT  MOVIES.TITLE, RATING, ACTOR_LEADING
    FROM MOVIES,
        ACTORS
      WHERE MOVIES.TITLE = ACTORS.TITLE;
QUIT;
```

SAS Log Results:

```
NOTE: SQL execution methods chosen are:
      sqxslct
         sqxjhsh
            sqxsrc( MOVIES )
            sqxsrc( ACTORS )
```

Contact Information:
If you would like more information or have any questions about this tip, please contact:

Kirk Paul Lafler
Software Intelligence Corporation
P.O. Box 1390
Spring Valley, California 91979-1390
E-mail: KirkLafler@cs.com