Exploring the MSGLEVEL= SAS System Option and PROC SQL _METHOD Option

Kirk Paul Lafler, Software Intelligence Corporation, Spring Valley, California

Abstract

The SQL Procedure contains powerful options for users to take advantage of. This paper explores the fully supported _METHOD option as an applications development and tuning tool to provide an understanding of the processing hierarchy (or execution plan) of any SQL query.

Introduction

PROC SQL supports a powerful option called **_METHOD**. Since its implementation, many SAS® SQL users have expressed very favorable comments for the value-added information it provides on the SAS Log. In fact, the _METHOD option is worth exploring simply due to the benefits associated with gaining a better understanding of the processes during specific PROC SQL operations, query evaluation, algorithm selected by the optimizer and used in the processing of a query, or testing and debugging operations.

Tables Used in Examples

The data used in all the examples in this paper uses the movies and actors data sets (tables). The Movies table, below, consists of twenty-two observations (rows) and six variables (columns): Title, Length, Category, Year, Studio, and Rating. Title, Category, Studio, and Rating are defined as character columns with Length and Year being defined as numeric columns.

	Title	Length	Category	Year	Studio	Rating
1	Brave Heart	177	Action Adventure	1995	Paramount Pictures	R
2	Casablanca	103	Drama	1942	MGM / UA	PG
3	Christmas Vacation	97	Comedy	1989	Warner Brothers	PG-13
4	Coming to America	116	Comedy	1988	Paramount Pictures	R
5	Dracula	130	Horror	1993	Columbia TriStar	R
6	Dressed to Kill	105	Drama Mysteries	1980	Filmways Pictures	R
7	Forrest Gump	142	Drama	1994	Paramount Pictures	PG-13
8	Ghost	127	Drama Romance	1990	Paramount Pictures	PG-13
9	Jaws	125	Action Adventure	1975	Universal Studios	PG
10	Jurassic Park	127	Action	1993	Universal Pictures	PG-13
11	Lethal Weapon	110	Action Cops & Robber	1987	Warner Brothers	R
12	Michael	106	Drama	1997	Warner Brothers	PG-13
13	National Lampoon's Vacation	98	Comedy	1983	Warner Brothers	PG-13
14	Poltergeist	115	Horror	1982	MGM / UA	PG
15	Rocky	120	Action Adventure	1976	MGM / UA	PG
16	Scarface	170	Action Cops & Robber	1983	Universal Studios	R
17	Silence of the Lambs	118	Drama Suspense	1991	Orion	R
18	Star Wars	124	Action Sci-Fi	1977	Lucas Film Ltd	PG
19	The Hunt for Red October	135	Action Adventure	1989	Paramount Pictures	PG
20	The Terminator	108	Action Sci-Fi	1984	Live Entertainment	R
21	The Wizard of Oz	101	Adventure	1939	MGM / UA	G
22	Titanic	194	Drama Romance	1997	Paramount Pictures	PG-13

The ACTORS data set (table) consists of thirteen observations (rows) and three variables (columns): Title, Actor_Leading, and Actor_Supporting which are all character columns, and is illustrated below.

	Title	Actor_Leading	Actor_Supporting
1	Brave Heart	Mel Gibson	Sophie Marceau
2	Christmas Vacation	Chevy Chase	Beverly D'Angelo
3	Coming to America	Eddie Murphy	Arsenio Hall
4	Forrest Gump	Tom Hanks	Sally Field
5	Ghost	Patrick Swayze	Demi Moore
6	Lethal Weapon	Mel Gibson	Danny Glover
7	Michael	John Travolta	Andie MacDowell
8	National Lampoon's Vacation	Chevy Chase	Beverly D'Angelo
9	Rocky	Sylvester Stallone	Talia Shire
10	Silence of the Lambs	Anthony Hopkins	Jodie Foster
11	The Hunt for Red October	Sean Connery	Alec Baldwin
12	The Terminator	Arnold Schwarzenegge	Michael Biehn
13	Titanic	Leonardo DiCaprio	Kate Winslet

Displaying Informational SAS Log Messages with MSGLEVEL=

SAS users can control how much information the SAS System writes to the SAS log by specifying the MSGLEVEL= SAS System option in an Options statement. The MSGLEVEL= option supports two possible values: **N** (which is the default) to print standard notes, warnings, and error messages; and **I** to print standard notes, warnings, error messages, plus additional information about sort, merge, and index processing. When specifying **MSGLEVEL=I** in an options statement, SAS displays the sort product that was used in a sort operation, a warning when variables are overwritten during merge processing; and the name of the available index that was used in index processing (or helpful suggestions on what can be done to influence SAS to use an available index); along with the usual assortment of notes, warnings, and error messages.

To demonstrate the effect of a **MSGLEVEL=I** option statement the following example illustrates a simple SQL join query on two tables, MOVIES and ACTORS. As shown in the resulting SAS Log, an informative message was generated explaining that the SAS system chose to use an available index, Rating, to optimize WHERE clause processing. This use of the MSGLEVEL=I system option provides users with a better understanding of what the SAS system did to improve processing, as well as the specific name of the index that was selected during processing of the query.

SQL Code

```
OPTIONS MSGLEVEL=I ;
PROC SQL ;
  SELECT MOVIES.TITLE, RATING, LENGTH, ACTOR_LEADING
    FROM MOVIES,
         ACTORS
      WHERE MOVIES.TITLE = ACTORS.TITLE
        AND RATING = 'PG' ;
QUIT ;
```

Log Results

```
OPTIONS MSGLEVEL=I ;
PROC SQL ;
  SELECT MOVIES.TITLE, RATING, LENGTH, ACTOR_LEADING
    FROM MOVIES,
         ACTORS
      WHERE MOVIES.TITLE = ACTORS.TITLE
        AND RATING = 'PG' ;

INFO: Index Rating selected for WHERE clause optimization.

QUIT ;
```

PROC SQL Join Algorithms

When it comes to performing PROC SQL joins, users supply the names of the tables for joining along with the join conditions, and the PROC SQL optimizer determines which of the four available join algorithms to use for performing the join query operation. The four join algorithms available to the optimizer include:

- ✓ **Nested Loop** – A nested loop join algorithm may be selected by the SQL optimizer when processing small tables of data where one table is considerably smaller than the other table, the join condition does not contain an equality condition, first row matching is optimized, or using a sort-merge or hash join has been eliminated.
- ✓ **Sort-Merge** – A sort-merge join algorithm may be selected by the SQL optimizer when the tables are small to medium size and an index or hash join algorithm have been eliminated from consideration.
- ✓ **Index** – An index join algorithm may be selected by the SQL optimizer when indexes created on each of the columns participating in the join relationship will improve performance.
- ✓ **Hash** – A hash join algorithm may be selected by the SQL optimizer when sufficient memory is available to the system, and the BUFFERSIZE option is large enough to store the smaller of the tables into memory.

The _Method Option and Code Descriptions

The PROC SQL _METHOD option can be specified as an effective way to analyze a query process or for debugging purposes. Processing information from the _METHOD option is automatically displayed on the Log using a variety of codes. The complete list of codes available with the _METHOD option along with their corresponding descriptions is displayed in the following table.

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.

Application of the _METHOD Option

In the following example a _METHOD option is specified to show the processing hierarchy in a two-way equi-join. As illustrated in the SAS Log, the PROC SQL optimizer utilized a hash join algorithm in the performance of the join query.

SQL Code

```
OPTIONS MSGLEVEL=I ;
PROC SQL _METHOD ;
   SELECT MOVIES.TITLE, RATING, ACTOR_LEADING
      FROM MOVIES,
           ACTORS
         WHERE MOVIES.TITLE = ACTORS.TITLE
           AND RATING = 'PG' ;
QUIT ;
```

Log Results

```
OPTIONS MSGLEVEL=I ;
PROC SQL _METHOD ;
  SELECT MOVIES.TITLE, RATING, LENGTH, ACTOR_LEADING
    FROM MOVIES,
         ACTORS
      WHERE MOVIES.TITLE = ACTORS.TITLE
        AND RATING = 'PG' ;
NOTE: SQL execution methods chosen are:
      sqxslct
          sqxjhsh
              sqxsrc( MOVIES )
              sqxsrc( ACTORS )

INFO: Index Rating selected for WHERE clause optimization.

QUIT ;
```

Conclusion

SAS users are encouraged to use the MSGLEVEL=I SAS System option and the _METHOD PROC SQL option. Used together, SAS provides an effective way to display useful information and achieve greater insight into the processes performed during specific PROC SQL operations, including query evaluation, the algorithm that is selected and used by the optimizer in the processing of a query's index, testing and debugging, and other processes.

References

Lafler, Kirk Paul (2016), *"Triggering the SAS SQL Execution Plan,"* Gateway Area of Users of SAS Software (GAUSS) 2016 Meeting, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2013). *PROC SQL: Beyond the Basics Using SAS, Second Edition*, SAS Institute Inc., Cary, NC, USA.

Lafler, Kirk Paul and Charlie Shipp (2013), *"Exploring the PROC SQL _METHOD Option,"* Proceedings of the 2013 Western Users of SAS Software (WUSS) Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul and Charlie Shipp (2013), *"Exploring the PROC SQL _METHOD Option,"* Proceedings of the 2013 MidWest SAS Users Group (MWSUG) Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul and Charlie Shipp (2013), *"Exploring the PROC SQL _METHOD Option,"* Proceedings of the 2013 NorthEast SAS Users Group (NESUG) Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2013), *"Exploring the PROC SQL _METHOD Option,"* Proceedings of the 2013 SAS Global Forum (SGF) Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2012), *"Exploring the PROC SQL _METHOD Option,"* Proceedings of the 2012 MidWest SAS Users Group (MWSUG) Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2010), *"DATA Step and PROC SQL Programming Techniques,"* Ohio SAS Users Group (OSUG) 2010 One-Day Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2009), *"Exploring DICTIONARY Tables and SASHELP Views,"* Proceedings of the 2009 South Central SAS Users Group (SCSUG) Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2009), *"Exploring DICTIONARY Tables and SASHELP Views,"* Proceedings of the 2009 Western Users of SAS Software (WUSS) Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2009), *"Exploring DICTIONARY Tables and SASHELP Views,"* Proceedings of the 2009 PharmaSUG SAS Users Group Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2008), *"Kirk's Top Ten Best PROC SQL Tips and Techniques,"* Wisconsin Illinois SAS Users Conference (June 26th, 2008), Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2008), *"Exploring the Undocumented PROC SQL _METHOD Option,"* Proceedings of the 2008 Western Users of SAS Software (WUSS) Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2007), *"Undocumented and Hard-to-Find PROC SQL Features,"* Proceedings of the 2007 NorthEast SAS Users Group (NESUG) Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2007), *"Undocumented and Hard-to-Find PROC SQL Features,"* Proceedings of the 2007 PharmaSUG Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2006), *"A Hands-on Tour Inside the World of PROC SQL,"* Proceedings of the 31st Annual SAS Users Group International Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2004). *PROC SQL: Beyond the Basics Using SAS*, SAS Institute Inc., Cary, NC, USA.

Lafler, Kirk Paul (2003), "*Undocumented and Hard-to-find PROC SQL Features*," Proceedings of the 2007 Western Users of SAS Software (WUSS) Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2002). *PROC SQL Programming Tips*; Software Intelligence Corporation, Spring Valley, CA, USA.

Shipp, Charles Edwin and Kirk Paul Lafler (2013), *"Exploring the PROC SQL _METHOD Option,"* Proceedings of the 2013 SAS Global Forum (SGF) Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Trademark Citations

Author Information

Kirk Paul Lafler is an entrepreneur, founder and consultant at Software Intelligence Corporation, and has been using SAS software since 1979. As a SAS application developer, programmer, certified professional, provider of SAS consulting services, mentor, lecturer and adjunct professor at San Diego State University, advisor and adjunct professor at University of California San Diego Extension, and emeritus sasCommunity.org Advisory Board member, Kirk has taught SAS courses, seminars, workshops and webinars to thousands of users around the world.. As the author of seven books including PROC SQL: Beyond the Basics Using SAS, Third Edition (SAS Press. 2019) and Google® Search Complete (Odyssey Press. 2014); and hundreds of papers and articles; Kirk has been selected as an Invited speaker, trainer, keynote and section leader at SAS International, regional, special-interest, local, and in-house user group conferences and meetings; and is the recipient of 25 “Best” contributed paper, hands-on workshop (HOW), and poster awards.

Comments and suggestions can be sent to:

Kirk Paul Lafler
SAS® Consultant, Application Developer, Programmer, Data Analyst, Educator and Author
Software Intelligence Corporation
E-mail: KirkLafler@cs.com
LinkedIn: https://www.linkedin.com/in/KirkPaulLafler/
LinkedIn: https://www.linkedin.com/in/Order-of-Magnitude-Analytics/
Twitter: @sasNerd