Exploring the PROC SQL _METHOD Option
Kirk Paul Lafler, Software Intelligence Corporation, Spring Valley, California
Charles Edwin Shipp, Consider Consulting Corporation, San Pedro, California

Abstract
The SQL Procedure contains powerful options for users to take advantage of. This presentation and poster explores the fully supported _METHOD option as an applications development and tuning tool, and learn how to use this powerful option to understand and control how a query processes.

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.

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.

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

Displaying Additional 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**

```sql
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**

```sql
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 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.

**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

```sql
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

```sql
OPTIONS MSGLEVEL=I;
PROC SQL;
    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
    sqxjsh
    sqxsrec( MOVIES )
    sqxsrec( ACTORS )
INFO: Index Rating selected for WHERE clause optimization.
QUIT;
```

Conclusion

The SQL Procedure's _METHOD option, along with the MSGLEVEL=I system option, provides users with a powerful and effective tool for gaining greater insight into the processes during specific PROC SQL operations, query evaluation, the algorithm selected and used by the optimizer in the processing of a query, testing and debugging operations, and other processes.

References


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


Acknowledgments
We want to thank Chris Aultman, MWSUG 2013 Posters Section Chair, for accepting our abstract and paper; as well as the MWSUG Executive and Conference Committees for organizing a great conference!

Trademark Citations
SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.

Author Information
Kirk Paul Lafler is consultant and founder of Software Intelligence Corporation and has been using SAS since 1979. He is a SAS Certified Professional, provider of IT consulting services, trainer to SAS users around the world, and sasCommunity.org emeritus Advisory Board member. As the author of five books including PROC SQL: Beyond the Basics Using SAS, Second Edition (SAS Press 2013); PROC SQL: Beyond the Basics Using SAS (SAS Press 2004), Kirk has written more than five hundred papers and articles, been an Invited speaker and trainer at three hundred-plus SAS International, regional, special-interest, local, and in-house user group conferences and meetings, and is the recipient of 22 “Best” contributed paper, hands-on workshop (HOW), and poster awards.

Charles Edwin Shipp is a programmer, consultant and author, and has been using the SAS and JMP software since 1980. He is credited in the original JMP manual for his roles in the early days. He has written more than one hundred papers and has been an invited speaker at more than one hundred International, regional, local, and special-interest groups. He is the recipient of 12 “Best” contributed paper and poster awards. Charlie is the co-author of three books including the ever-popular Books by Users (BBU) book, Quick Results with SAS/GRAPH Software. Currently, Charlie is involved as an eBook author, App developer for Apple iPad, sasCommunity.org Advisory Board member, consultant for 4Life, AdvoCare, Genesis Pure, Melaleuca, Trivani Foundation International, and consultant in JMP and JMP Genomics.

Comments and suggestions can be sent to:

Kirk Paul Lafler
Senior Consultant, Application Developer, Data Analyst, Trainer and Author
Software Intelligence Corporation
E-mail: KirkLafler@cs.com
LinkedIn: http://www.linkedin.com/in/KirkPaulLafler
Twitter: @sasNerd

Charles Edwin Shipp
Senior Consultant, Programmer, eBook Developer, Trainer and Author
Consider Consulting Corporation
E-mail: CharlieShipp@aol.com
Twitter: @ShippAhoy

~~~