Abstract
Users have at their disposal a unique and powerful feature for retaining historical copies of SAS data sets. This collection of copied generation data sets, representing versions of the same data set consists of a root member name and a unique version number, is aptly referred to as a generation group. This presentation explores the power associated with generation data sets by showing users their purpose, how they are created and maintained, the approach used for accessing a specific version of a generation group, as well as other useful techniques. Simple coding examples will be illustrated to reinforce concept and application.
**Generation Data Sets**

- Are archived versions of SAS data sets
- Provide historical versions of a data set
- Can only be specified for a data set (not view)
- Stored as part of a Generation Group
- SAS processes the base version by default

**Purpose of a Generation Data Set**

- Provides versions for a data set
- Access a specific data set version in a DATA or PROC step
- Backup and recover a specific data set
- Compare two or more data sets
  - Audit purposes
  - Observations – Adds, deletes, modifications

**Terminology**

- Base version – most recent data set version
- Generation group – consists of the base version and one or more historical versions
- Historical versions – older copies of the base data set
- Newest version – data set version closest to the base version in a generation group
- Oldest version – data set version farthest from the base version in a generation group
Terminology (continued)

- Generation number – incremental number that identifies a historical version in a generation group
- GENMAX= – specifies the maximum number of versions to be created in a generation group (use with DATA statement) (max is 999)
- GENNUM= – specifies a version from a generation group to use as input to a DATA and/or PROC step process

Exploring Generation Data Sets

Generation Groups & Versions

- SAS appends a 4-character version number to its data set (member) name. When GENMAX=4:

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOVIES</td>
<td>Base version</td>
</tr>
<tr>
<td>MOVIES#003</td>
<td>Newest historical version</td>
</tr>
<tr>
<td>MOVIES#002</td>
<td>2nd newest historical version</td>
</tr>
<tr>
<td>MOVIES#001</td>
<td>Oldest historical version</td>
</tr>
</tbody>
</table>
Generation Group Creation

- Using the GENMAX=4 data set option in a DATA step, we'll create four versions as follows:

```plaintext
data movies (GENMAX=4); /* Base version */
    set class.movies;
data movies;   /* Oldest version */
    set movies;
data movies;   /* 2nd oldest version */
    set movies;
data movies;   /* Newest version */
    set movies;
run;
```

Processing the Base Version

- Using PROC PRINT with our MOVIES example, we'll specify the base version of our generation group as follows:

```plaintext
proc print data=movies; /* Base version */
run;
```

< or >

```plaintext
proc print data=movies (gennum=0);
run;
```

Processing the Newest Version

- Using PROC PRINT with our MOVIES example, we'll specify the newest version in our generation group as follows:

```plaintext
proc print data = movies (gennum = -1);
run;
```

< or >

```plaintext
proc print data = movies (gennum = 3);
run;
```
Processing 2 Generations Back

- Using PROC PRINT with our MOVIES example, we'll specify the version two generations back from the base version as follows:

```plaintext
proc print data = movies (gennum = -2);
run;
```

Processing 3 Generations Back

- Using PROC PRINT with our MOVIES example, we'll specify the version three generations back from the base version as follows:

```plaintext
proc print data = movies (gennum = -3);
run;
```

Managing Generations Groups

- The DATASETS procedure provides the tools to manage generation groups.
- Using the DELETE statement in PROC DATASETS, the following options can be specified:
  - ALL – deletes all data sets including historical versions
  - HIST – deletes only historical versions
- Only generation groups can be copied with the COPY statement (not individual versions)
- A specific version or the entire generation group can be renamed with the CHANGE statement
In Conclusion

- Generation data sets are archived versions of SAS data sets
- Provide historical versions of data sets
- Can only be specified for a data set
- Provide access to a specific data set version for processing
- SAS processes the base version by default
- Generation groups can be managed using PROC DATASETS

Acknowledgments
The author would like to thank Debbie Buck and Pete Lund, SUGI Tutorials Section Co-Chairs, for accepting this paper and presentation, as well as the SUGI Leadership for their support of a great Conference.

Contact Information
Kirk Paul Lafler, a SAS Certified Professional® and SAS Alliance Partner® (1996 - 2002), has been using SAS software since 1979. As a consultant and trainer with Software Intelligence Corporation, he provides consulting services and hands-on SAS training to users around the world. Kirk has written four books including PROC SQL: Beyond the Basics Using SAS by SAS Institute, Power SAS and Power AOL by Apress, and more than one hundred peer-reviewed articles and papers for professional journals and SAS User Group proceedings. His popular SAS Tips column appears regularly in the BASAS, HASUG, SANDS, SAS, and SESUG Newsletters and websites. Kirk can be reached at:

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

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.