Introduction to SAS® Data Loader: The Power of Data Transformation in Hadoop

Keith Renison, SAS Institute Inc.

ABSTRACT

Organizations are loading data onto Hadoop platforms at an extraordinary rate. However, in order to extract value from these platforms, the data must be prepared for analytic exploitation. As the volume of data grows, it becomes increasingly more important to reduce data movement, as well as to leverage the computing power of these distributed systems. This paper provides a cursory overview of SAS® Data Loader, a product specifically aimed at these challenges. We cover the underlying mechanisms of how SAS Data Loader works, as well as how it's used to profile, cleanse, transform, and ultimately prepare data for analytics in Hadoop.

INTRODUCTION

Hadoop platforms are designed for storing and processing massive amounts of data on distributed, commodity hardware. This means that much of the data that needs to be processed cannot feasibly be moved or copied to other systems. In addition, the computational layers provided by the open-source community, such as MapReduce and Hive, require knowledge of advanced programming languages, or at a minimum, a working knowledge of structured query languages (SQL). Given the widespread and rampant adoption of Hadoop technologies, it is critical that better access to data management tools and techniques be given to non-programming users in the business, especially to those that prepare data for analytics. With these challenges in mind, SAS created SAS Data Loader, an easy-to-use application designed to execute powerful transformation directives ***inside*** the Hadoop platform.

It is important to understand the broad range of functional capabilities within the application. However, it is equally important to understand how SAS Data Loader is implemented and how it interacts with Hadoop. This is because Hadoop platforms can appear as if they are a relational database management system (RDBMS). In reality, they operate quite differently. Therefore, this paper will first review the unique way in which SAS Data Loader is implemented in order to provide a background for what is happening when you execute various directives. Then, we will explore at a high level the variety of ways that the application can be used. This is not designed to take the place of the user's guide, but rather provide a practical introduction to concepts and capabilities of the tool. This paper also assumes that you have a basic understanding of the core Hadoop projects, namely Hadoop Distributed File System (HDFS), MapReduce, and Hive. If you need to brush up on these topics, consult the *Recommended Reading* section at the end.

HOW DOES SAS DATA LOADER WORK?

The SAS Data Loader architecture is designed to deploy easily, yet extend the data management capabilities beyond that which is built into Hadoop. The simplest way to think of the architecture is in two parts: the SAS Data Loader client application and the platform components in Hadoop.

THE SAS DATA LOADER CLIENT

The SAS Data Loader client is a virtual application, also known as a ***vApp***. A vApp consists of one or more virtual machines with pre-installed software, and runs inside a virtual operating system, layered on top of your existing operating system. The benefit of a vApp is that there is no installation required, and it can be run on a variety of host operating systems and hardware profiles. Think of it as a pristine copy of SAS® installed by the very smart people in SAS R&D onto a lightweight operating system, and handed over to you all ready to go. You will want to check the latest system requirements on the SAS Technical Support site for supported host operating systems and virtual application players. The SAS Data Loader vApp contains a Linux operating system, SAS® Foundation, the SAS/ACCESS® Interface to Hadoop, a complete set of compatible Hadoop JAR files for the supported Hadoop distributions, as well as a web tier used to host the SAS Data Loader interface. Once the server components are deployed (described in the

next section) and the client connection is configured, using SAS Data Loader is achieved by simply turning on the SAS vApp, and using the web browser from your local machine to operate the client.

THE COMPONENTS IN HADOOP

One of the most important parts of using SAS inside Hadoop, is the deployment of a lightweight execution engine, lovingly referred to as the SAS® Embedded Process (EP). The SAS Embedded Process is deployed on each node of the Hadoop cluster, and serves three very important purposes when it comes to SAS Data Loader:

1) Accepts SAS Data Step 2 program code (DS2) as part of the SAS® In-Database Code Accelerator for Hadoop
2) Initiates data quality routines as part of the SAS® Data Quality Accelerator for Hadoop
3) Lifts data in parallel into the distributed SAS in-memory analytic engine, called the SAS® LASR™ Analytic Server

DS2 works across multiple platforms, and does not require code updates even as the underlying technology platform evolves. So no matter what platform language is popular at the time, you can continue to leverage your existing knowledge and tools for uninterrupted business continuity.

Functionally speaking, SAS Data Loader uses the existing Hadoop infrastructure, such as executing explicit HiveQL, or engages the SAS Embedded Process to do things it's better suited to do. It's also important to note that the SAS Embedded Process makes its requests for resources through MapReduce, and is therefore compliant with the load balancing and resource negotiation layers of Hadoop (YARN).

Even though these technologies are at play, as a user, you don't need to make decisions about the execution method. Instead, simply use the intuitive interface to prepare data for analytics, and SAS will make the call for you. Figure 1 shows how these pieces fit together:

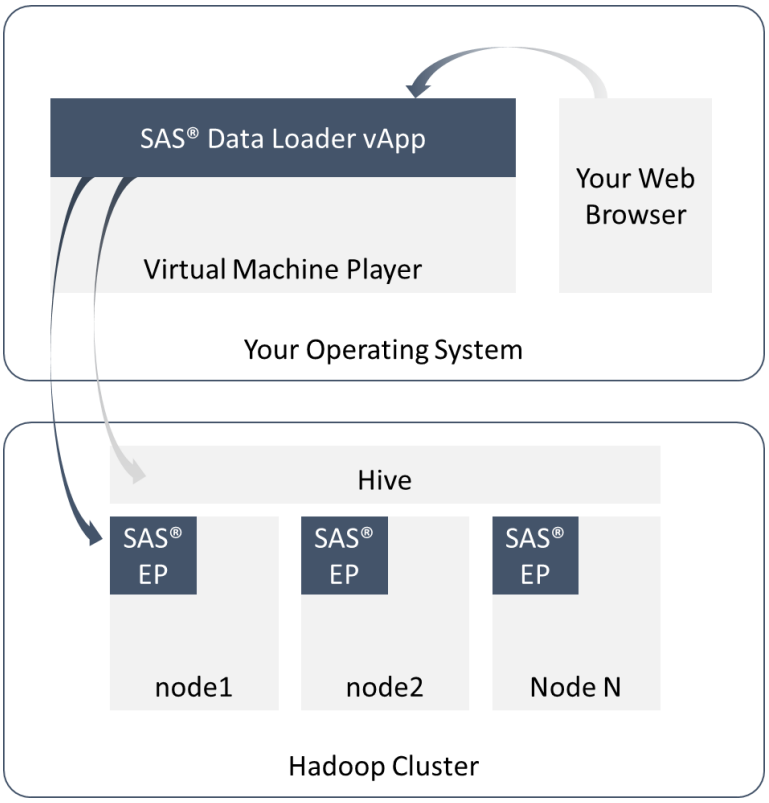

Figure 1: High-Level View of the Data Loader Architecture

WHAT DOES SAS DATA LOADER DO?

There are a wide variety of data management capabilities built into SAS Data Loader. The screen capture below (Figure 2) shows the directives currently available. Although these directives are set out as individual tasks, think of them in the context of four main capabilities: data movement, data profiling, data quality, and data transformation.

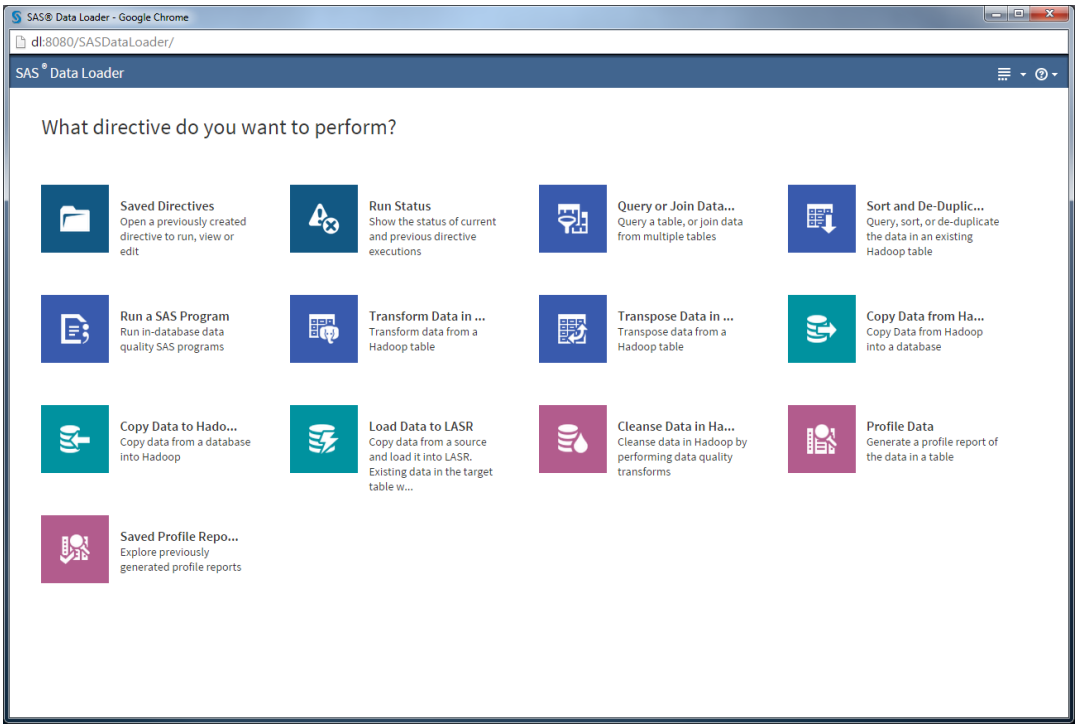

Figure 2: Main Data Loader Directive Page

DATA MOVEMENT

With all this talk about transforming data in place, we often forget about the need to move data around. SAS Data Loader has the ability to pull data from outside sources such as relational DBMS systems that use the open-source project from Apache called *Sqoop*, as well as to move SAS data sets to and from Hadoop. This can be particularly useful when merging large data sets inside Hadoop with outside data sets from, for example, an Oracle database. This is done using the ***Copy Data to Hadoop*** directive.

This same mechanism can push data to relational database systems as well, using the ***Copy Data from Hadoop*** directive. This is commonly used at the end of a data transformation routine, where aggregated results are pushed back to other operational systems for downstream reporting.

Also included is the ability to load data to the SAS LASR Analytic Server. With this capability you can lift a table from HDFS using parallel steams into the memory space of the SAS LASR Analytic Server. From a business perspective, this is one of the most important features, enabling you to combine the power of data management inside Hadoop, with the high-performance of in-memory analytics. A very common example is to pull data into Hadoop, clean it up a bit, and then push it to LASR to be explored with SAS® Visual Analytics or SAS® Visual Statistics. Efficiently stitching together data preparation tasks with analytic tasks can result in significant time and resource savings.

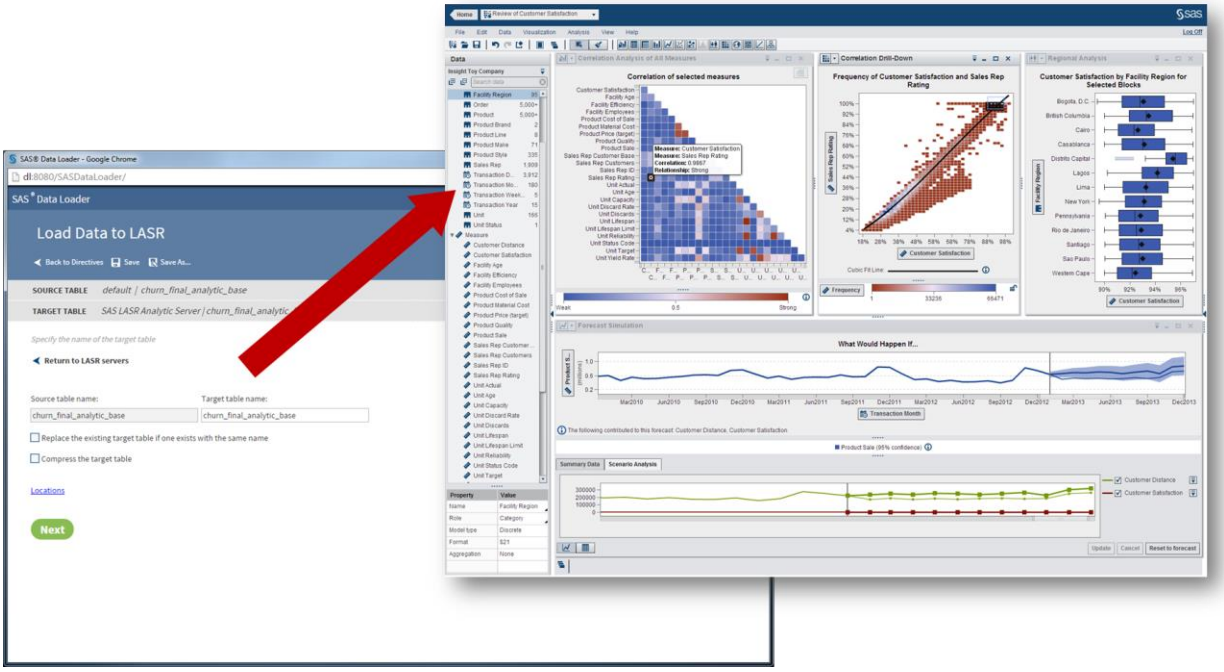

Figure 3: Loading Data to the SAS LASR Analytic Server

DATA PROFILING

The inputs to the data profiling directive are fairly simple. However, the results are powerful. Because Hadoop was designed to be schema on read, metadata has always been an afterthought. This can present quite a challenge when dealing with extremely large or new data sets. In this case, a data profiling routine can be an invaluable tool that lets you learn more about a data set before rolling up your sleeves to work with it. The data profiling directive helps answers critical questions, such as these:

- What is the cardinality of my variables?
- How many blank or null values will I be joining on?
- How many distinct values and patterns exist for this variable?
- What are the summary statistics for my numeric variable, and do they make sense?
- What are the formats for these character variables?

Without answers to these questions at the beginning, downstream data transformation and analytics might have to be reworked, costing valuable time and resources. Figure 4 provides an example of the types of insights available in the Profile Report.

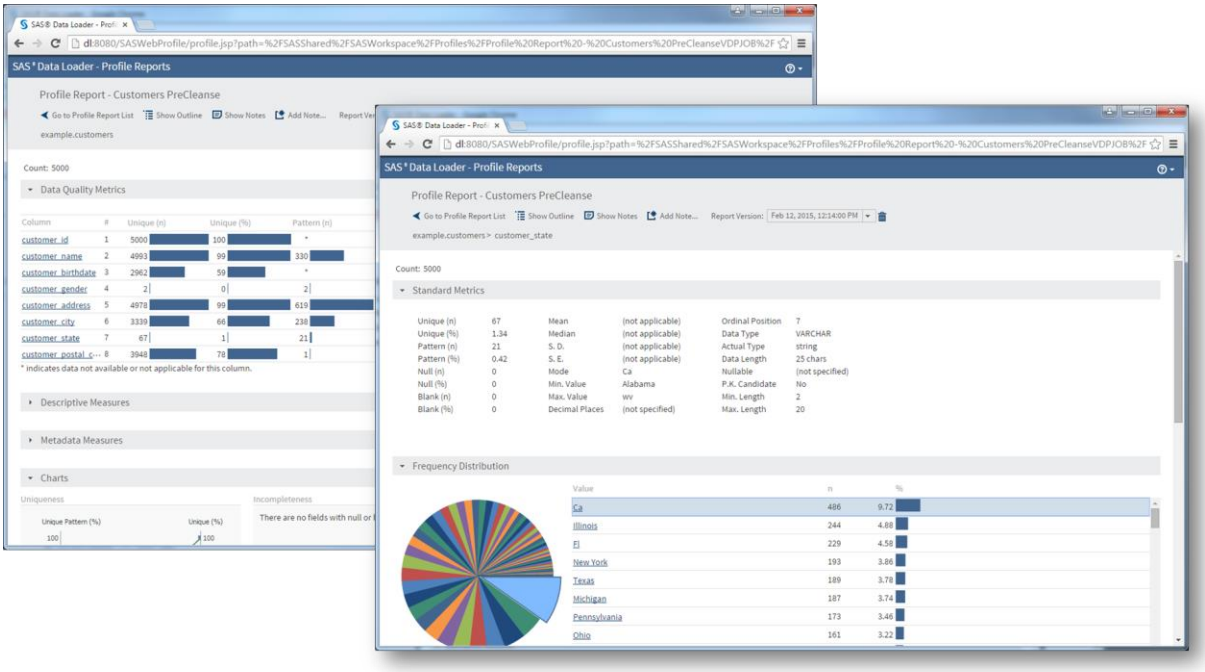

Figure 4: Data Profile Report

DATA QUALITY

As mentioned earlier, Hadoop is designed for storing and processing raw data, not enforcing transactional schemas and business rules. This presents a significant data quality challenge, because vast amounts of messy data is pushed into Hadoop that cannot practically be moved to other systems for cleansing. It can also be extremely tedious, complex, and time-consuming to write data quality algorithms from scratch using the programming languages available to the Hadoop platform. This is why embedding SAS data quality logic inside Hadoop, and providing easy access for non-programmers is critical to the successful preparation of data for analytics. Among the extensive library of SAS data quality routines, there are specializations for common data types, such as dates, geography, phone numbers, account numbers, business entities, and so on.

Match Coding

Match coding is the process of applying fuzzy logic to columns in a data set. For example, let's suppose that you have a clean data set that contains standard addresses of your customers, and you would like to see which of these customers appear in last month's clickstream data (containing user-entered addresses). Using SAS Data Loader, you can generate identical matching keys for values that indeed represent the same household, but likely have small variations that would prevent an identical match in a normal join.

Parsing

Have you ever needed to separate a field into multiple parts? The parse feature of SAS Data Loader enables you to extract certain entities, like area codes, or someone's family name, and to place them in their own column.

Standardization

The standardization transform is quite powerful when you are dealing with messy data. Lots of spaces? Misplaced numeric values, nonstandard country names, the list goes on. There are a wide variety of standardization algorithms that can be applied.

Identification

Use the identification transform to determine if a column contains certain values, such as contact information, dates, email, field names, offensive content, and phone numbers.

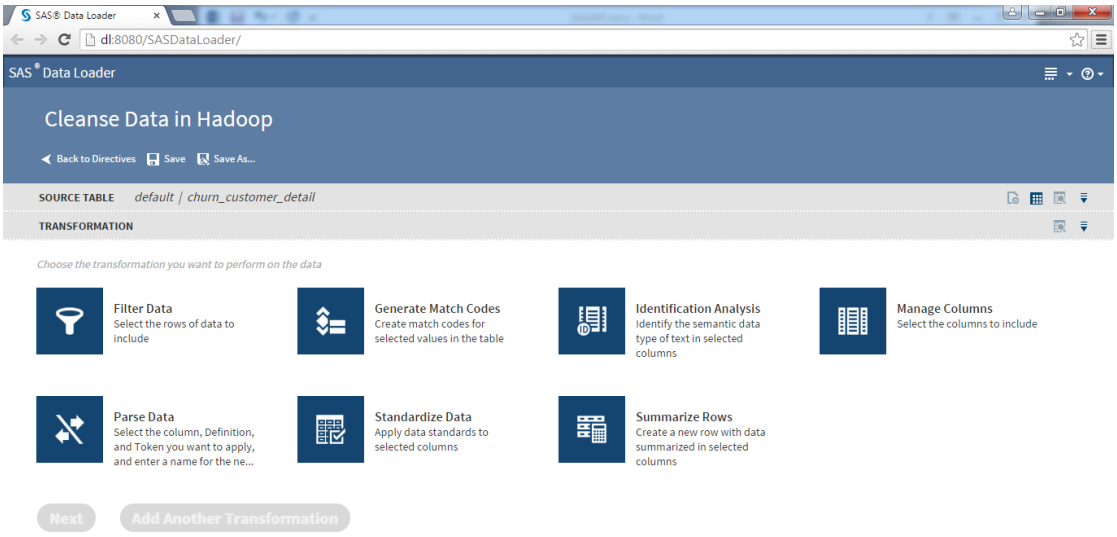

Figure 5: Data Quality Directives

DATA TRANSFORMATION

SAS Data Loader contains a number of other directives used to sort, copy, filter, join, transpose, aggregate, or otherwise prepare data for analysis. The directives are arranged by common business tasks and therefore contain some cross over in functionality. There is no need to review each directive in extraordinary detail here since the purpose of some are self-explanatory. However, here is some background on a few important transforms:

Transpose Data in Hadoop

Database administrators love a small number of columns in a detailed fact table, with multiple dimension tables tied into it. Analytic work, however, requires one very wide table, one record per subject, with lots of variables. For this, the transpose capability is extremely powerful in creating analytic base tables. Because this powerful directive has a capacity to create a large number of columns, a nice tutorial is included in the interface to ensure that you know what to expect before executing the directive.

Transform Data in Hadoop

This directive is used to daisy chain a series of sort, filter, and summarization steps, because these are commonly used together when preparing a single table for analysis.

Query and Join Data in Hadoop

Last, but certainly not least, is one of the most commonly used and powerful transforms: the ability to run queries and combine data sets. This directive contains much of what you would expect, but also exposes the ability to embed custom Hive expressions, as well as edit the HiveQL directly.

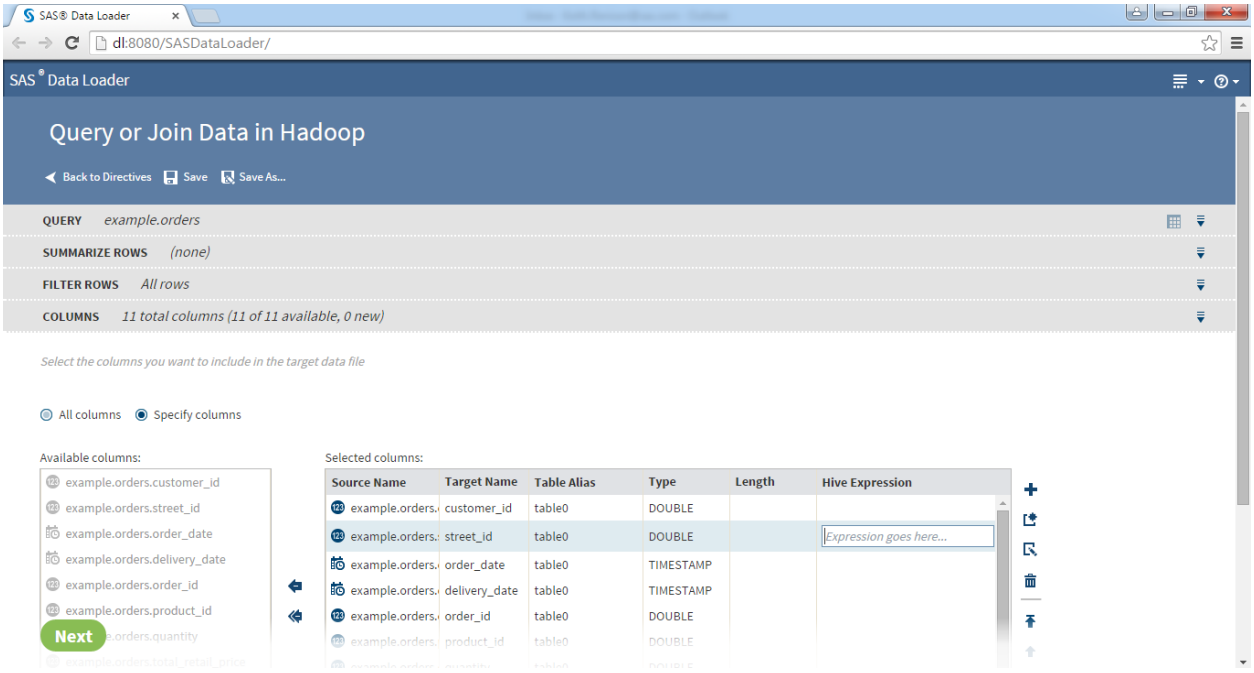

Figure 6: Custom Hive Expression

CONCLUSION

As we enter the New Analytic Culture, big data platforms are here not only to stay, but to fundamentally change the way we exploit data. Like many new technologies, early use becomes the exclusive domain of programmers. This, however, is a temporary condition. As distributed platforms mature, tools like SAS Data Loader tear down the barriers to entry, exposing the extraordinary power of SAS to the masses on a platform designed to handle them.

I hope this paper has provided you with an introduction to the power of SAS Data Loader for Hadoop as well as an understanding of how it works.

RECOMMENDED READING

- *Hadoop: The Definitive Guide*, by Tom White
- *SAS® Data Loader for Hadoop: User's Guide*
- *SAS® Data Loader for Hadoop: Installation and Configuration Guide*
- *SAS/Hortonworks Blog: Data Management and the New Analytics Culture*

CONTACT INFORMATION

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

Keith Renison
SAS Institute, Inc.
+1 (919) 531-9612
keith.renison@sas.com
www.sas.com