

Understanding Change in the Enterprise

Liz McIntosh, Nancy Rausch, and Bryan Wolfe; SAS Institute Inc., Cary, NC

ABSTRACT

SAS® provides a wide variety of products and solutions that address analytics, data management, and reporting. It can be challenging to understand how the data and processes in a SAS deployment relate to each other, and how changes in your processes affect downstream consumers. This paper presents visualization and reporting tools for lineage and impact analysis. These tools enable you to understand where the data for any report or analysis originates or how data is consumed by data management, analysis, or reporting processes. This paper introduces new capabilities to import metadata from third-party systems to provide lineage and impact analysis across your enterprise.

INTRODUCTION

Data lineage is the management and analysis of object and metadata relationships, including dependencies and lifecycle. This management and analysis reveals where data comes from, how it is transformed, and where it is going. Data lineage also includes impact analysis. Impact analysis makes known which objects are affected when another object is changed or deleted.

Enterprises large and small use a variety of products and solutions to produce data, processes, reports, and other content. The challenge is to understand how the content produced relates to each other. With a good understanding of the relationships, individuals throughout the enterprise can answer questions pertaining to data lineage and impact analysis.

The individual data steward, for example, is able to assess the impact of a change to the data model. Before increasing the length of a column the data steward asks questions like, “Where is a column used?” and “Will the data be truncated by an ETL process using the column?” Likewise, before a job is deleted, the ETL developer can determine which reports are impacted and contact the report owners. Or the business analyst, upon seeing a surprising result in a report, can trace the origins of the data to verify that the result is indeed accurate before taking action.

Lineage capabilities, especially our ability to tie business and technical information together in a single and cohesive information store, is key toward helping you understand all the data assets in your environment.

THE IMPORTANCE OF LINEAGE IN DATA GOVERNANCE

Managing data as a key corporate asset is the primary premise behind the concept and discipline of data governance. Data governance has grown in importance as organizations are forced to comply with industry or government regulations, cut costs to improve margins, or use data-driven initiatives to increase revenue. A successful data governance initiative helps by improving visibility into a corporation’s data assets, which are used to drive better and quicker business decisions. Data lineage is a key component of any data governance initiative.

RELATIONSHIP SERVICE ARCHITECTURE

Lineage capabilities, available in the first maintenance release of SAS 9.4, rely on the SAS Relationship Service and the underlying relationship database. The relationship service collects and stores metadata about a variety of content from SAS and sources outside of SAS as well as processes that include resources used in data management, business intelligence, and data integration.

The information storage format has been generalized such that any content type that is registered with the SAS Web Infrastructure Platform can be persisted in the relationship database. This allows for all types of resources to be modeled that support multiple types of users and that provide visibility into both physical and logical object relationships.

UNDERSTANDING THE RELATIONSHIP DATABASE CONTENTS

The relationship database schema consists of *resources* and *relationships*. A resource is a basic metadata representation of a data asset or process. The resource must have an object type and ID. The object type corresponds to a registered type in the SAS Web Infrastructure Platform and identifies the type of resource

represented. The ID uniquely identifies the resource in the source system, for example a SAS metadata ID, GUID, or URL. The resource can also have a label or name.

A relationship describes how two resources are related. The relationship contains a “subject” resource, the related resource, and the relationship type. In the relationship type descriptions below, the subject resource is referred to as the first object and the related resource as the second object.

The pre-defined types of relationships are:

- Is dependent on - A relationship in which the first object is dependent on the second object, and the first object cannot function or be defined without the second object. *Example:* Report is dependent on a Table.
- Is parent of - A relationship in which the first object is the parent of the second object, and the second object cannot be found within a hierarchy without the presence of the first object. *Example:* Folder is a parent of Table.
- Contains - A relationship in which the first object contains the second object, and the second object cannot exist or is unusable without the first object. *Example:* Table contains Column.
- Is synonymous with - A relationship in which the first object and the second object are alike in meaning or significance. *Example:* Customer is synonymous with Client.
- Is associated with - A relationship in which the first object is associated with the second object through some known or unknown system or method. *Example:* Stored process is associated with Stored Process Server.
- Is equal to - A relationship in which the first object and the second object represent the same thing but are accessed through different methods or systems. *Example:* Table A is equal to Table B.

ADDING CONTENT TO THE RELATIONSHIP DATABASE

The relationship service has a variety of ways to collect and load content. For data and relationship information generated by SAS products, the information is collected and loaded by the relationship service itself. Individual SAS content types have a relationship support service that is called during the load process. The support service analyzes a resource to determine its relationships to other content and returns information to the load process. The load process then writes the information to the relationship database. This process can either be scheduled to run automatically or run on-demand using a command-line utility.

Scheduled Collection and Loading Using SAS Management Console

SAS Management Console is used to schedule the collection and loading of information about resources and their relationships.

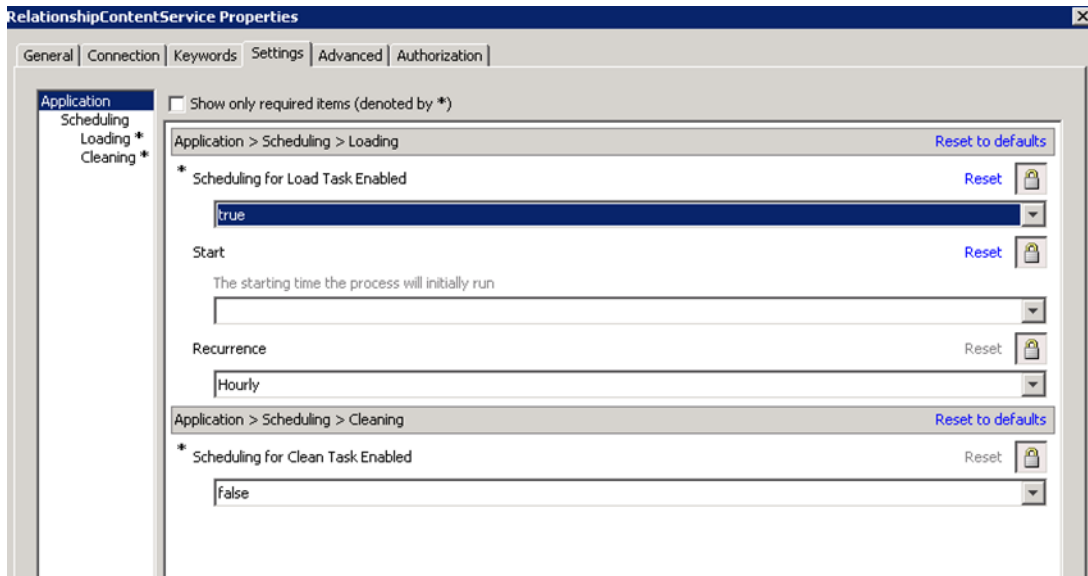
To enable the relationship service collection and loading:

- Log in to the SAS Management Console as an administrative user.
- Expand the **Configuration Manager** under the **Application Management** node.
- Expand **SAS Application Infrastructure**, locate **Web Infra Platform Services 9.4** and expand this node.
- Select **RelationshipContentService**, right-click, and select **Properties** from the pop-up menu,



Display 1. Select Properties to Configure the Relationship Content Service for Scheduled Loading

- On the **Settings** tab, change the value for **Scheduling for Load Task Enabled** from "false" to "true".
- Restart the web application server.



Display 2. Set Scheduling for Load Task Enabled to "true"

Using the options shown in Display 2, relationship metadata is loaded into the relationship database when the web application server is restarted. The default setting schedules the load operation to occur every hour. Subsequent runs of the load operation search for all content that has been changed or created since the last load and update the relationship database for the changed objects.

You can schedule the load to run hourly, daily, weekly, or you can create your own custom schedule.

The options shown in Display 2 also provide for cleaning to be scheduled. Cleaning will remove resources from the relationship database that no longer exist in the SAS system. Once cleaning is enabled, the default setting for this operation is to run once daily at the specified time.

The steps to enable cleaning are:

- On the **Settings** tab, change the value for **Scheduling for Clean Task Enabled** from "false" to "true".
- Restart the web application server.

On-Demand Collection and Loading Using a Command-Line Interface

The relationship database is loaded on-demand using the relationship loader utility. This utility, like the scheduled service, is an administrative tool that collects and loads relationships for SAS content. The utility uses standard content filtering options to select resources to be analyzed for relationship information, or you can choose all resources using the `-loadAll` option.

The relationship loader utility is an executable file launched from a command line. On a typical Windows installation of SAS 9.4, it is in the directory `<SASHOME>\SASPlatformObjectFramework\9.4\tools\admin\`.

The relationship loader is launched using

```
sas-relationship-loader [options...]
```

Refer to the documentation for more information about this utility.

Loading Relationship Information from Business Data Network and SAS Data Management Platform

The Business Data Network and SAS Data Management Platform have their own processes to load relationship information into the relationship database. Business Data Network 3.1 populates the relationship database when a term is created or modified. For existing Business Data Network systems, a one-time load is required to perform the initial population. SAS® Data Management Studio 2.6 and SAS Data Management Server 2.6 extract relationship

details to a file that is then imported using the Lineage viewer. The Lineage viewer, a new web-based interface for viewing the relationship information, is discussed in the SAS Lineage Viewer section.

UNDERSTANDING CHANGE THROUGH QUERYING AND REPORTING

Once content has been loaded into the relationship database, you can create data lineage and impact analysis reports. The reports reveal where data comes from and where it is going, and also aid in determining the impact of change.

Using the relationship reporter command-line utility, an authorized user can create reports based on the data in the relationship database. The relationship reporter utility combines filtering options available with other command-line utilities with a new set of options for filtering relationships. For SAS content, the utility also looks up the resource in the SAS system and provides additional details about the resources and applies security controls. The report is created in a LIST (a simple text) or comma-separated value (CSV) format. The CSV format can be imported to spreadsheets or used in SAS reports, and is discussed in the "Creating Reports Using the CSV File" section. Additional formats are planned in future releases.

Reports created with the relationship reporter utility can also be automated.

There are two parts to creating a report with the relationship reporter utility. First, determine the subjects for the report. This is accomplished by either specifying the exact paths of the objects to retrieve, or by searching for the objects using common filter criteria (such as filtering by name, type, dates, and so on). Second, determine the type of relationship information that should be included.

The relationship reporter utility supports a set of pre-defined reports that hide many of the complexities involved with filtering of relationship information. The pre-defined reports are:

- lineage – returns resources related to the subject using *is dependent on* and *contains* relationships.
- impact - returns resources that have an *is dependent on* relationship from the subject.
- directDependencies - returns the objects for which the subject directly *is dependent on*.
- indirectDependencies - returns the objects for which the subject indirectly *is dependent on*.

Customized reports are created using options available for relationship content filtering and relationship filtering. Relationship content filtering options limit the types of related resources queried. Relationship filtering options limit the relationship types included in the query. For example, a query can request that only Tables (relationship content filter option) that are Equivalent (relationship filter option) with the specified subject are displayed in the report.

The relationship reporter utility is an executable file launched from a command line. On a typical Windows installation of SAS 9.4, it is in the directory <SASHOME>\SASPlatformObjectFramework\9.4\tools\.

The relationship reporter is launched using

```
sas-relationship-reporter [options...]
```

In order to execute the batch utility the following options are required on the command line:

- -host : the host name of the SAS middle-tier
- -port : the port of the SAS middle-tier
- -user : the user name
- -password : the user's password

For brevity, the examples that follow do not display these options.

SPECIFYING THE SUBJECT OR SUBJECTS OF THE REPORT

The relationship reporter utility uses common content filtering to determine the subject of the report. For a particular subject, where the path is known, it is passed as an argument. A path includes the full path, the resource name, and its object type. This example specifies that the subject of the report is the "Sample: Hello World" stored process by providing its path.

```
sas-relationship-reporter "/Products/SAS Intelligence Platform/Samples/Sample:  
Hello World(StoredProcess) "
```

If there is more than one subject of interest for this type of query, specify multiple paths:

```
sas-relationship-reporter "/path-1/resource-name-1(object-type-1) "
"/path-2/resource-name-2(object-type-2) " ...
```

No other content filter options can be used when paths are specified.

The subject or subjects might be of a particular object type or share a common location. The subjects in this example are all tables with the word "SAS" in their name, and located in the "/Shared Data" folder tree.

```
sas-relationship-reporter -types Table -name "SAS" -folderTree "/Shared Data"
```

Other filter options allow you to filter the objects by their dates, responsibilities, and keywords. Refer to the documentation for more information about these options.

USING PRE-DEFINED REPORTS

The first step to creating a report is to determine the subjects for the report. The previous section introduced the content filtering options used to accomplish this. The second step is to determine the type of relationship information that should be included. The relationship reporter utility provides pre-defined reports that hide the complexities needed to specify relationship filtering options.

Lineage Report

A *lineage* report displays resources related to a subject using the *Is dependent on* and *Contains* relationships. The first example creates a *lineage* report a SAS report named **ClassReport.srx**.

```
sas-relationship-reporter -report lineage -types Report -name "ClassReport.srx"
-format list
```

The following is the output of this report in LIST format.

```
❶ "/Shared Data/Reports/ClassReport.srx" (SAS report)
❷ --> Is dependent on: "/Shared Data/Maps/Class Relational Map" (Information map (relational))
❸ --> Is dependent on: "/Shared Data/SASHELP/CLASS" (Table)
   --> Is dependent on: "/Shared Data/SASHELP/SASHELP" (Library)
   ❹ --> Contains: "Age" (Column)
   --> Contains: "Height" (Column)
   --> Contains: "Name" (Column)
   --> Contains: "Sex" (Column)
   --> Contains: "Weight" (Column)
   --> Is dependent on: "/Shared Data/SASHELP/SASHELP" (Library)
   --> Contains: "Age" (InformationMapDataItem)
   ❺ --> Is dependent on: "Age" (Column) - in "/Shared Data/SASHELP/CLASS" (Table)
   --> Contains: "Height" (InformationMapDataItem)
   --> Is dependent on: "Height" (Column) - in "/Shared Data/SASHELP/CLASS" (Table)
   --> Contains: "Name" (InformationMapDataItem)
   --> Is dependent on: "Name" (Column) - in "/Shared Data/SASHELP/CLASS" (Table)
   --> Contains: "Sex" (InformationMapDataItem)
   --> Is dependent on: "Sex" (Column) - in "/Shared Data/SASHELP/CLASS" (Table)
   --> Contains: "Weight" (InformationMapDataItem)
   --> Is dependent on: "Weight" (Column) - in "/Shared Data/SASHELP/CLASS" (Table)
```

Output 1. Default Lineage Report

The first line (❶) in Output 1 lists the path, name, and object type of the subject being analyzed. The arrows (❷) indicate the direction of the relationship of the item on that line with the item on the previous line. For example, **ClassReport.srx** is dependent upon the **Class Relational Map** SAS Information Map (❷), and **Class Relational Map** is dependent on the **CLASS** table (❸) that contains 5 columns (❹). The **Class Relational Map** SAS Information Map contains the **Age** data item (❺) that is dependent upon the **Age** column in the **CLASS** table.

The lineage report allows us to trace the origin of data used in **ClassReport.srx** though the information map back to

the table and columns.

Impact Report

An impact report is essentially the opposite of the *lineage* report. It retrieves the resources related using the *Is dependent on*, *Contains*, or *Is associated with* relationships from the subject. This example creates an *impact* report for all Table objects named **CARS**.

```
sas-relationship-reporter -report impact -types Table -name CARS -format list
```

Here is the output of this report in LIST format:

```
❶ "/Shared Data/DataSources/CARS" (Table)
❷ <-- Is dependent on: "/Shared Data/Cubes/BID_MDDBCARS" (Cube)
❸ <-- Is dependent on: "/Shared Data/Maps/OLAPMaps/MDDBCARS" (Information map (OLAP))

❹ "/User Folders/sasdemo/My Folder/CARS" (Table)
<-- Is dependent on: "Parameters" (Prompt group)
<-- Contains: "/User Folders/sasdemo/My Folder/cars-pak" (Stored process)
❺ <-- Is dependent on: "/User Folders/sasdemo/My Folder/Cars Report.srx" (SAS report)
<-- Is dependent on: "Parameters" (Prompt group)
<-- Contains: "/User Folders/sasdemo/My Folder/cars-stream" (Stored process)
```

Output 2. Default Impact Report

In Output 2 there are two subjects (❶ ❹). The content filtering requested all tables named **CARS** regardless of the location. The arrow indicates that the **BID_MDDBCARS** cube (❷) *Is dependent on* the **CARS** table (❶). The **MDDBCARS** information map *Is dependent on* the **BID_MDDBCARS** cube (❷).

The impact report allows us to determine the resources that are impacted if the subject of the report changes. For example, if the **CARS** table (❹) is changed the **Cars Report.srx** (❺) is impacted.

CUSTOMIZING THE PRE-DEFINED REPORTS

The previous examples use the pre-defined reports to filter the relationship types returned for the subject. It is also possible to specify the related resources that should be included in the report. This is accomplished using the same common content filters that are used for specifying the subject. These options are indicated on the command line using `-relContent {options...}`.

The following example is similar to the command-line that produced Output 1, but includes the additional `-relContent` option. This option limits the related resources to those that contain the word "CLASS" in the name.

```
sas-relationship-reporter -report lineage -types Report -name "ClassReport.srx"
-format list -relContent {-name "CLASS"}
```

This time the output contains only related objects that match the `-relContent` option.

```
"/Shared Data/Reports/ClassReport.srx" (SAS report)
--> Is dependent on: "/Shared Data/Maps/Class Relational Map" (Information map (relational))
--> Is dependent on: "/Shared Data/SASHELP/CLASS" (Table)
```

Output 3. Report Filtering for Related Resources That Contain “Class” in the Name

When the same report is run using “AGE” instead of “CLASS”, no related resources are displayed because there are no resources directly related that contain “AGE” in the name. However, an inspection of Output 1 shows that there are indirect dependencies on the “AGE” columns. An indirect dependency exists when the resource is part of the lineage but not directly related to the subject. To obtain related resources, regardless of where they are in the lineage, `-report indirectDependencies` is used.

```
sas-relationship-reporter -report indirectDependencies -types Report -name
"ClassReport.srx" -relContent {-name "AGE"} -format list
```

Here is the output for an `indirectDependencies` report with the `-relContent` option.

```
"/Shared Data/Reports/ClassReport.srx" (SAS report)
❶ --> Is dependent on: "/Shared Data/Maps/Class Relational Map" (Information map (relational))
❷ --> Contains: "Age" (InformationMapDataItem)
❸ --> Is dependent on: "Age" (Column) - in "/Shared Data/SASHELP/CLASS" (Table)
```

Output 4. IndirectDependencies Report

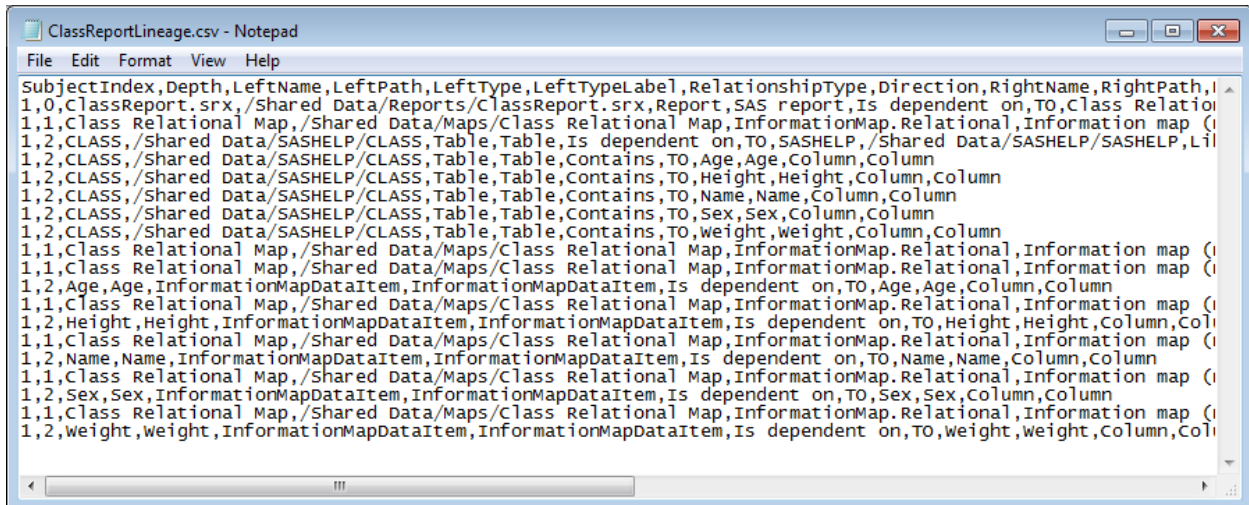
The report in Output 4 shows two resources that are indirectly related to the **ClassReport.srx** report and that contain “AGE” in the name (❷ ❸). The relationship from the report to the **Class Relational Map** (❶) is included in the report to provide a complete lineage from the report to the indirect dependencies.

CREATING REPORTS USING THE CSV FILE

In addition to creating plain-text reports, the Relationship Reporter batch utility can be used to create CSV files. The relationship data used to create the lineage report in Output 1 is retrieved and stored in the `ClassReportLineage.csv` file using this command:

```
sas-relationship-reporter -report lineage -types Report -name "ClassReport.srx"
-format csv -file ClassReportLineage.csv
```

CSV is a plain-text format consisting of data fields separated by commas. Column headings are included in the first row of data in Display 3.



Display 3. Partial View of the ClassReportLineage.csv File Created by the SAS Relationship Reporter

You can import the CSV data into a SAS data set, and then create a report using the PRINT, REPORT, or any other SAS procedure or DATA step code. Display 4 shows the result of using PROC REPORT to create an HTML format lineage report similar to the plain-text report of Output 1.

The complete code to create the output of Display 4 is in the Appendix.

Subject Path and Name	Subject Type	Relationship	Related Item Type	Related Item Path and Name
/Shared Data/Reports/ClassReport.srx	SAS report	Is dependent on	Information map (relational)	/Shared Data/Maps/Class Relational Map
/Shared Data/Maps/Class Relational Map	Information map (relational)	Contains	InformationMapDataItem	Age
				Height
				Name
				Sex
				Weight
		Is dependent on	Library	/Shared Data/SASHELP/SASHELP
			Table	/Shared Data/SASHELP/CLASS
/Shared Data/SASHELP/CLASS	Table	Contains	Column	Age
				Height
				Name
				Sex
				Weight
		Is dependent on	Library	/Shared Data/SASHELP/SASHELP
Age	InformationMapDataItem	Is dependent on	Column	Age
Height	InformationMapDataItem	Is dependent on	Column	Height
Name	InformationMapDataItem	Is dependent on	Column	Name
Sex	InformationMapDataItem	Is dependent on	Column	Sex
Weight	InformationMapDataItem	Is dependent on	Column	Weight

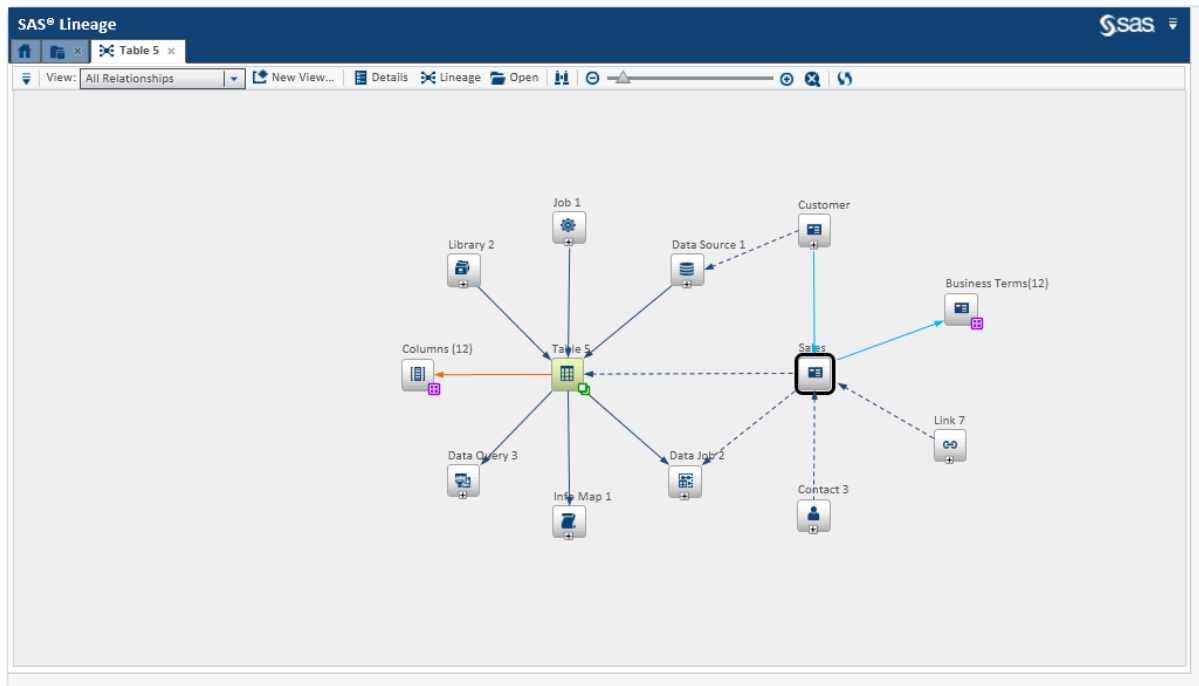
Display 4. Lineage Report for the ClassReport.srx Object in HTML Format

UNDERSTANDING CHANGE THROUGH VISUALIZATION

SAS provides applications that give a graphical view of the data that is stored in the relationship service database. Visualizing the relationships helps in understanding where data comes from and where it is going, and also aids in determining the impact of change. The Lineage viewer and Business Data Network 3.1 are two of these applications.

Lineage Viewer

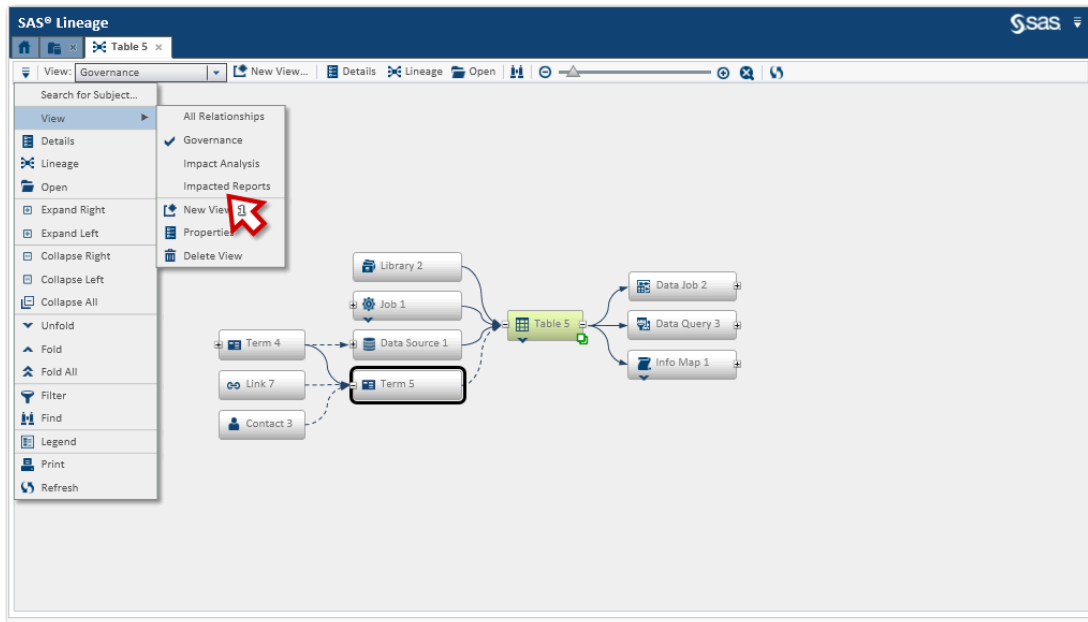
The Lineage viewer is used to visualize information stored in the relationship database. The lineage viewer supports many different types of relationships in its view, including data dependencies, associated objects, synonyms, and other resource relationships. The viewer includes relationships to third-party objects such as Microsoft Excel spreadsheets or documentation. Display 5 is an example of the lineage viewer.



Display 5. Lineage Viewer

The Lineage viewer has a number of different available views and filtering capabilities to help in understanding the relationships between your data. Display 5 shows the All Relationships view in Lineage for the subject **Table 5**. A quick look at these relationships reveals useful information. For example, we see that the table is dependent upon a job and is used in a data query, an information map, and a data job. We also see that the table gains some context from the Sales business terms. **Table 5** contains twelve columns that have been collapsed in this view that can be expanded when more information about a particular column in the table is needed.

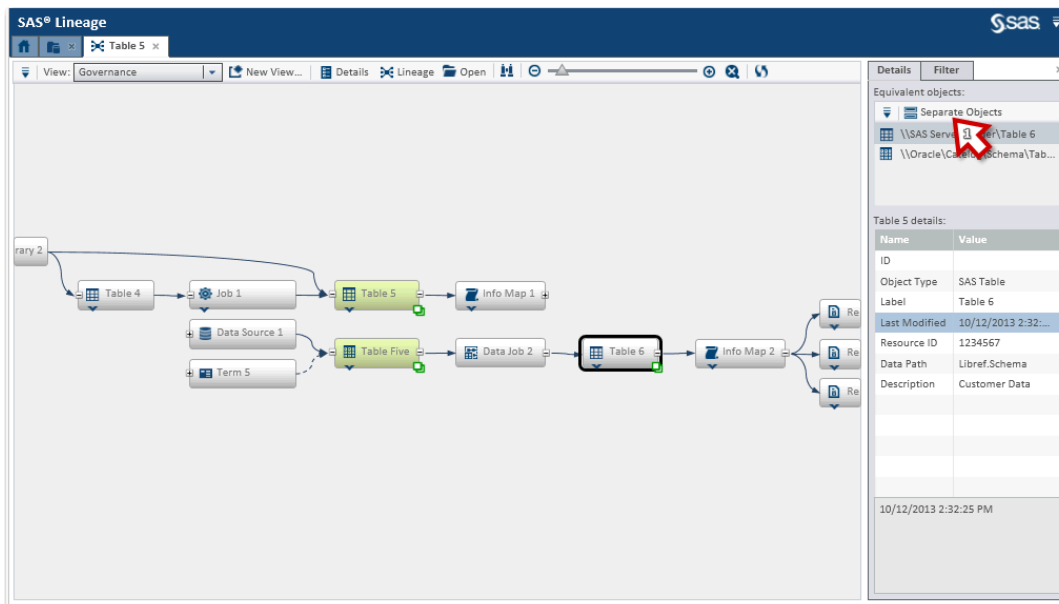
The Governance view (Display 6) supports a more traditional data flow. This view shows how physical data flows through your system from source to targets such as reports, information maps, and resources in other applications or systems.



Display 6. An Example of the Lineage Governance View

In Display 6, arrow 1 highlights the Governance view selection. **Table 5** is again the subject. We see the dependency that **Table 5** has on **Job 1**. We also see the job, the data query, and the information map that is dependent on the table. The relationships shown here are the similar as in Display 5, the difference being the layout that makes the data flow apparent.

Any information stored in the relationship database can be displayed, and there are a number of features such as filtering, collapsing and sub-setting, that help you visualize the data. You can also mark objects from different systems as equivalent, tying different systems together when exploring the relationships. Display 7 illustrates some of these features.

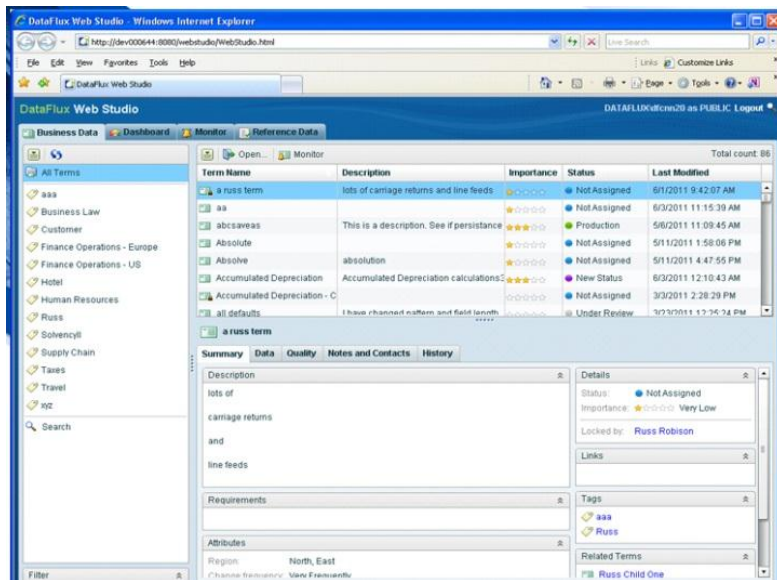


Display 7. Additional Features in the Lineage Viewer

In Display 7, arrow 1 highlights the ability to define equivalent resources, and this panel shows additional details for the selected resource.

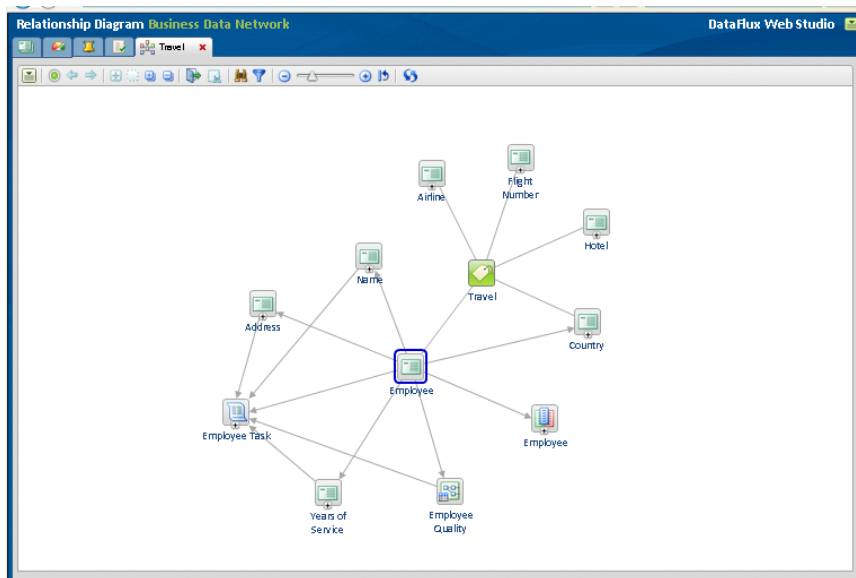
Business Data Network

The Business Data Network supports collaboration of domain knowledge between business, technical, and data steward users. Business Data Network is used as a single entry point for all data consumers to better understand their data. It contains a web-based user interface that documents business terms and their associated rules, jobs, applications, data, documentation, and other information. Technical users collaborate on the rules used to validate data, and share knowledge about data transformations using the data network. Data stewards view the data from a business standpoint and visualize problem areas by domain so that they can identify and fix issues more effectively. Display 8 shows the Business Data Network main page with documented business terms and associated rules that have been defined.



Display 8. Business Data Network

Typically, a user that understands their business terminology provides the initial information in the Business Data Network. This user also attaches documents or rules that describe each term. The technical user adds additional information related to the term, for example, jobs that are used to modify the term and data that is related to the term. The relationship service is used to store these relationships in the relationship database, and the Lineage viewer is used to display the relationships between the business and technical information. Display 9 shows a typical example of the business and technical information relationships that are stored in the relationship database and visualized in the Lineage viewer.



Display 9. Business Data Network Relationship Diagram

WHAT'S NEXT

SAS is continuing research and development in this area. You can expect to see support for more SAS content types, import from third-party products, and a public REST interface for loading and reporting. The Lineage viewer will also be integrated in other SAS products.

As we continue to build out the framework and see how the information stored in the relationship database is used more features will be available. As always, your feedback is welcome and encouraged.

CONCLUSION

The first maintenance release of SAS 9.4 introduces the relationship service and database and reporting features that enable you to understand the resources in the enterprise and how they are related. Business Data Network 3.1 and the Lineage viewer use the relationship service, and provide additional ways to interact and explore both physical and logical relationships. These features aid in data lineage and impact analysis, and ultimately provide the ability to understand change in the enterprise.

APPENDIX

The SAS code to create the output of Display 4 follows.

```
* Import the CSV data into a SAS data set;

proc import file='ClassReportLineage.csv' out=ClassReportLineage
  dbms=csv replace;
run; quit;

* Create an HTML-format report using the REPORT procedure;

ods _all_ close;

ods html file='ClassReportLineage.htm' style=HTMLBlue;

title; footnote;

proc report data=ClassReportLineage nowd missing;
  column SubjectIndex  Depth                LeftPath                LeftName
```

```

LeftTypeLabel RelationshipType RightTypeLabel RightPath;

define SubjectIndex / order noprint;
define Depth / order noprint;
define LeftPath / order left 'Subject Path and Name';
define LeftName / order noprint;
define LeftTypeLabel / order center 'Subject Type';
define RelationshipType / order center 'Relationship';
define RightTypeLabel / order center 'Related Item Type';
define RightPath / order left 'Related Item Path and Name';

ods html close;

```

REFERENCES

SAS Institute Inc. 2013. *SAS® 9.4 Intelligence Platform: System Administration Guide, Second Edition*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/documentation/cdl/en/bisag/66813/HTML/default/viewer.htm#titlepage.htm>

RECOMMENDED READING

SAS Institute Inc. 2014. "Data Management" SAS Support Community. Available at https://communities.sas.com/community/support-communities/sas_enterprise_data_management_integration

Rausch, Nancy, Mike Frost, and Mike Ames. 2014. "What's New in SAS® Data Management." *Proceedings of the SAS Global Forum 2014 Conference*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/resources/papers/proceedings14/index.html>.

Rausch, Nancy, and Malcolm Alexander. 2013. "Best Practices in SAS Data Management for Big Data." *Proceedings of the SAS Global Forum 2013 Conference*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/resources/papers/proceedings13/077-2013.pdf>.

ACKNOWLEDGMENTS

The author would like to thank Vince DelGobbo of SAS Institute Inc. for his valuable contributions to this paper

CONTACT INFORMATION

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

Liz McIntosh
100 Campus Drive
Cary, NC 27513
SAS Institute Inc.
(919) 677-8000
Liz.McIntosh@sas.com
support.sas.com

Nancy Rausch
940 NW Cary Parkway, Suite 201
Cary, NC 27513
SAS Institute Inc.
(919) 677-8000
Nancy.Rausch@sas.com
support.sas.com

Bryan Wolfe
100 Campus Drive

Cary, NC 27513
SAS Institute Inc.
(919) 677-8000
Bryan.Wolfe@sas.com
support.sas.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.