

Using SAS® Data Management Advanced to Ensure Data Quality for MDM

Brian W. Smith, 1-800-Flowers.com, Inc.

ABSTRACT

Data is a valuable corporate asset that when managed improperly can detract from a company's ability to achieve strategic goals. At 1-800-Flowers.com, Inc. (18F) we have embarked on a journey towards Data Governance through embracing Master Data Management (MDM). Along the path we've recognized that in order to protect and increase the value of our data we must take data quality into consideration at all aspects of data movement in the organization. This document will discuss the ways that SAS® Data Management is being leveraged by the team at 18F to create and enhance our data quality strategy to ensure data quality for MDM.

INTRODUCTION

Most software systems have collections of data which are shared and used by several applications that make up an organization's core capabilities which require a high-level of confidence that the data is verified, accurate, and trustworthy. Many companies' technology implementations grow organically over time, where new software systems are introduced just-in-time to address growing needs of the business. This growth often creates a situation where multiple software systems become systems of record for the same data; for example, multiple systems allow for the entry and retention of customer information thereby producing multiple views of a customer entity. The business value of enterprise applications is only as good as the data that can be leveraged.

Master Data Management (MDM) is a means to achieve an authoritative view of important data within an enterprise; however, maintaining an MDM system as well as ensuring the quality and trustworthiness of the data is a challenge; however, many MDM endeavors fail due to lack of planning around the initial, and ongoing data quality strategy. MDM requires a lithe methodology that looks at data quality issues not as an afterthought, but as a primary thought in the process of establishing and supporting the lifetime health and stability of an MDM platform.

This paper discusses how using SAS® Data Management in a variety of ways allows for the integration of data quality management tasks at both the transactional and batch levels.

NEED FOR DATA QUALITY WITH MDM

MDM systems reside in the center of a company's data ecosystem and typically perform as both a source of and a destination of relevant domain data. The information hosted in MDM is referenced by other systems as well as it can complete fractions of domain data for an entity which may be may reside in other systems. Organizations adopt Master Data Management systems to tackle many problems with managing their key data domains, whether it be to create process and control around data domains, or as means of de-duplicating customer or product information from multiple source systems.

The quality of data within an MDM is one of the determining factors of the success of a master data management initiative, as the solution is intended to provide a trusted source of truth about a data domain. When the data within an MDM system is considered suspect by its consumers, the MDM system fails to meet one of its key objectives. Unfortunately for the MDM system, it is not in full control of the quality of its data as the MDM is reliant on the various systems within an enterprise which provide the data.

A typical MDM system manages multiple silos of information encompassing everything from order management systems, e-commerce sites, mobile applications, customer relationship management systems, supply chain, third-party vendor, and much more. Each of these systems typically has their own method of data capture, and often the rules which govern data entry are different from system to system. Without a central data quality strategy implemented on each of the systems, the quality of the data within each system is open to scrutiny. This requires a data quality strategy be implemented during the loading

of data into an MDM system in order to prevent the MDM system from inheriting the data quality issues of its feeder systems. No one intends for this type of situation to occur; however, many companies wind up in a position of having silos of information about a master data domain, with unknown data quality levels, and spend large amounts of time and money trying to resolve them during an MDM implementation.

Often instituting a data quality program at the time of implementing a master data management strategy is an afterthought, and found necessary after profiling existing master data and discovering how much bad data exists in the current systems. Additionally, organizing redundant source data into a master data management system produces additional data quality issues in both the source and MDM target, due to mismatched definitions in the same like attributes between two different sources. As a master data management system matures the ability of keeping data quality and trust in the data is also challenging as new systems may be introduced via merger or acquisition, or the source systems themselves may change. An enterprise-level data quality program should be a key part of an MDM initiative, ideally data should be made candidate for data quality processing from its inception until it is at rest, and every now and then it won't hurt to wake it up and probe it again for quality assurance.

SAS® DATA MANAGEMENT ADVANCED AND MDM... AND DATA QUALITY

SAS® Data Management Advanced is a platform fully supportive of data management tasks, the platform offers data quality tools as well as a first class ETL engine. The core of the tenet of SAS Data Management is to support the ability to identify, profile, review, analyze and cleanse data. SAS Data Management combines data quality, data integration, and master data management into one application providing a fully fleshed out suite to assist in a master data management initiative.

How SAS Data Management fits with an MDM strategy is really a question of where does it fit, not how does it fit. SAS Data Management provides all of the tooling needed for fleshing out a robust set of rules and processes to clean, validate, enrich, validate, and load data for an MDM solution. Therefore, SAS® Data Management is able to encapsulate an MDM system where all data manipulation processes can be intercepted by the platform and processed with the data quality rules of an organization to ensure only clean, qualified data is entered into the MDM.

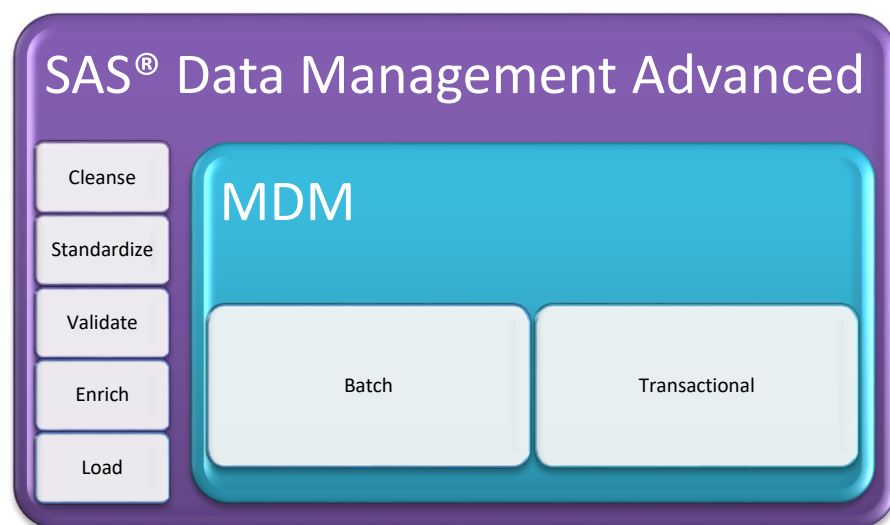


Figure 1 SAS® Data Management Advanced can encapsulate an MDM solution for data quality control on all data manipulation transactions.

With the packaging of SAS® Data Integration Studio and SAS® Data Management Studio/Server in the SAS® Data Management Advanced offering, organizations are able to build a common set of data quality process jobs which can be shared by both batch (ETL-based) and real-time transactions. This provides an opportunity for building a single set of data quality jobs which can be used for both the initial processing to load an MDM as well as the

ongoing loading of MDM and even be wired into the transactional processing of data into an MDM solution. Data Jobs are built in SAS® Data Management Studio by a developer and uploaded to a SAS® Data Management Server for use by both real-time and batch based clients.

The ability to use SAS Data Management Studio data quality jobs for both real-time and batch-driven data processing allows for the design of a data quality strategy around MDM which allows a consistent and trusted process to review, cleanse, and standardize data prior to it being loaded regardless of the method. Reducing the amount of bad data, and standardizing the attributes which will be evaluated by MDM for matching allows the matching engine of an MDM system to more accurately make decisions with the information it's given. Continuously this will produce more accurate results in the search for suspected duplicate records and help reduce the likelihood of false negative matches and false positive matches; both of which can be disruptive to the trust of master data.



DATA QUALITY JOBS WITH SAS® DATA MANAGEMENT

Data Jobs are the primary way to process data in SAS® Data Management, a data job specified a set of data-processing operations which flow from an input source to a target. Data Jobs are extremely flexible where they are able to take in data from many different relational database management systems, as well as flat-files, JMS queues, XML and others; the focus of this article looks at the External Data Provider node. The External Data Provider node enables a developer to create an input interface which can be accessed by several different external processes such as other data jobs, SAS Data Integration Studio DataFlux Data Service (and Batch Job) transformations, as well as allows a data job to be service-enabled on a SAS® Data Management Server and be called via SOAP or REST protocols.

CREATING A DATA QUALITY DATA JOB

The below example depicts creating a data quality data job in SAS® Data Management Studio.

Use-case:

Contact names are provided to data capture systems as a single string value, the data capture system stores contact names in two fields *ContactNameOne* and *ContactNameTwo*. The individual fields of a name are not parsed, e.g. "Bruce Wayne" is a single string. The MDM solution requires that the name of a person be broken up into individual fields for each part of the name. To further complicate matters, there are occasions where the customer name value provided may also include company names, numbers, and other data which are not relevant to a person's name. A data quality job is needed to review contact names, identify the values for a person name, parse the values, and return a person's name.

Figure 2 The data

	objectReferenceId	contactNameOne	contactNameTwo
1	1	John Doe	ACME Widgets, Inc
2	2	ACME Widgets, Inc	John Doe
3	3	Best Wishes	John Doe
4	4	Master John Doe	ACME Widgets, Inc
5	5	John and Jane Doe	
6	6	Mr. John J Doe	
7	7	1929342	John J. Doe

Note that the data depicts several examples of how the data capture system has received customer information

Solution Design

The data job needs to perform the following functions:

1. Identify which of the provided contact name fields contains a "NAME" value
2. Remove any special characters from the contact name
3. Identify if multiple person names exist in the contact name field (only the first name will be used by MDM)
4. Parse the contact name into person name fields
5. Create a new contact name value using only the First, Middle, and Last Names of the Person
6. Format all return values as uppercase strings
7. Return the results to the calling application, including the original values for review

Figure 3 Data job to retrieve a person's name from contact name fields

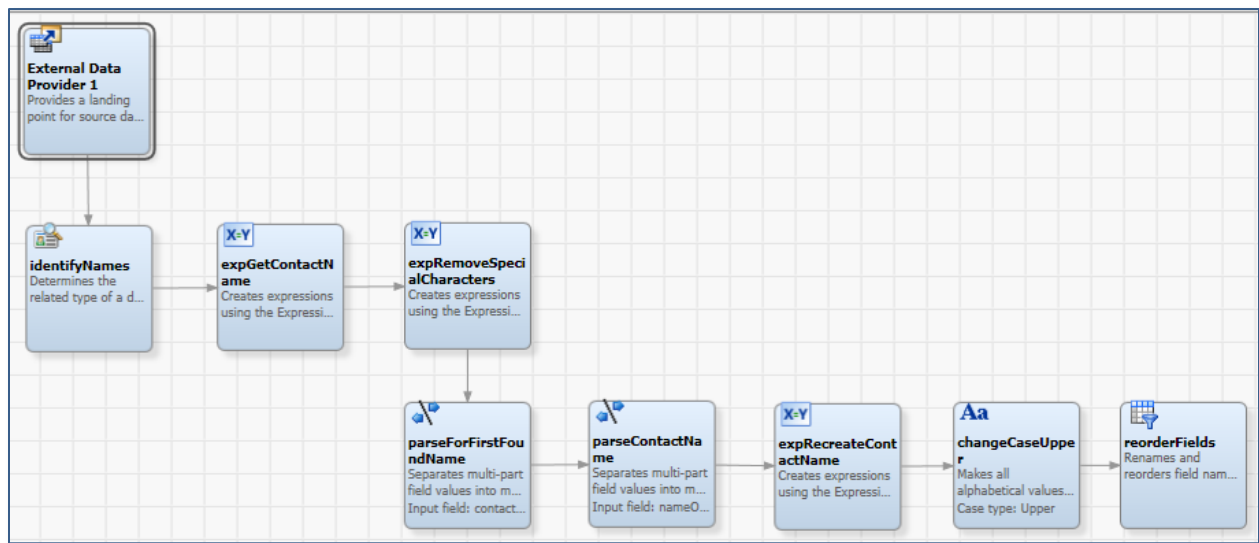
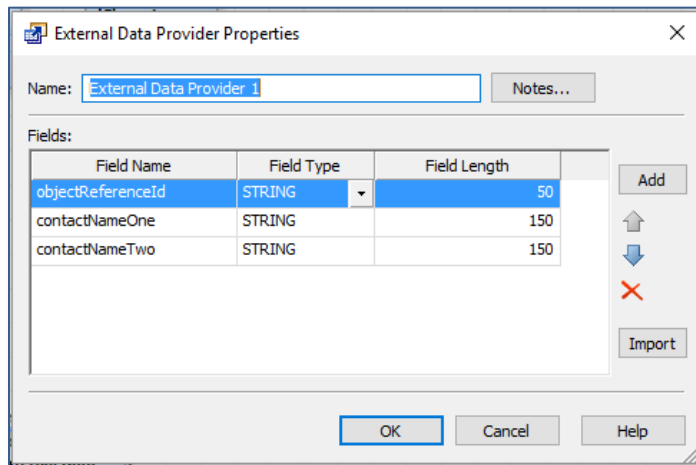


Figure 4 Establish the job inputs

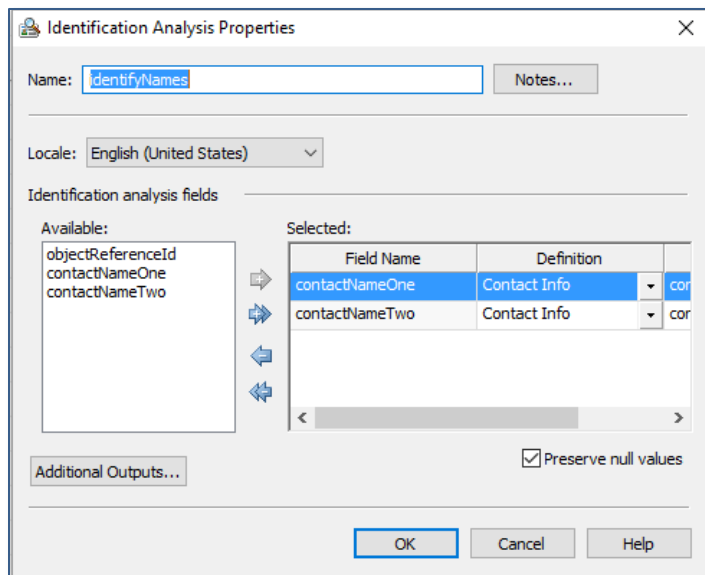


The dialog box is titled "External Data Provider Properties". It has a "Name:" field with the value "External Data Provider 1" and a "Notes..." button. Below this is a "Fields:" section containing a table with three columns: "Field Name", "Field Type", and "Field Length". The table lists three fields: "objectReferenceId" (STRING, 50), "contactNameOne" (STRING, 150), and "contactNameTwo" (STRING, 150). To the right of the table are buttons for "Add", "Import", and a red "X" icon. At the bottom are "OK", "Cancel", and "Help" buttons.

Field Name	Field Type	Field Length
objectReferenceId	STRING	50
contactNameOne	STRING	150
contactNameTwo	STRING	150

This data job will to be used as part of real-time data processing as well as in ETL batches to process many records at a time. In order to make the job accessible to external clients an External Data Provider node is used and configured with the parameters required to execute the data job

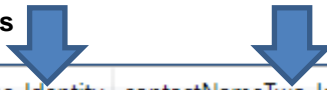
Figure 5 Identify the values in the contact name fields



The dialog box is titled "Identification Analysis Properties". It has a "Name:" field with the value "IdentifyNames" and a "Notes..." button. Below this is a "Locale:" dropdown menu set to "English (United States)". The "Identification analysis fields" section has two panes: "Available:" and "Selected:". The "Available:" pane lists "objectReferenceId", "contactNameOne", and "contactNameTwo". The "Selected:" pane contains a table with two columns: "Field Name" and "Definition". It lists "contactNameOne" and "contactNameTwo", both with the definition "Contact Info". To the left of the "Selected:" table are navigation arrows. At the bottom left is an "Additional Outputs..." button, and at the bottom right is a checked checkbox labeled "Preserve null values". At the very bottom are "OK", "Cancel", and "Help" buttons.

Field Name	Definition
contactNameOne	Contact Info
contactNameTwo	Contact Info

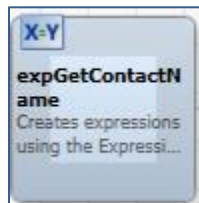
Figure 6 Results of Identification Analysis of the contact name fields



	objectReferenceId	contactNameOne	contactNameTwo	contactNameOne_Identity	contactNameTwo_Identity
1	1	John Doe	ACME Widgets, Inc	NAME	ORGANIZATION
2	2	ACME Widgets, Inc	John Doe	ORGANIZATION	NAME
3	3	Best Wishes	John Doe	ORGANIZATION	NAME
4	4	Master John Doe	ACME Widgets, Inc	NAME	ORGANIZATION
5	5	John and Jane Doe		NAME	BLANK
6	6	Mr. John J Doe		NAME	BLANK
7	7	1929342	John J. Doe	UNKNOWN	NAME

The Identification Analysis node uses the Quality Knowledge Base in DataFlux® to analyze the provided values and evaluate what kind of data is within a field.

Figure 7 Use an expression to find the proper contact name for a person




Using an Expression Node and some Expression Engine Language (EEL) scripting, testing of the results of the Identification Analysis of the contact name fields allows for the proper determination of which field has a person name value in it

```
string contactName

integer result
if (`contactNameOne_Identity` == "NAME")
    begin
        contactName = `contactNameOne`
        result = 0
    end
else if (`contactNameOne_Identity` != "NAME" and `contactNameTwo_Identity` == "NAME")
    begin
        contactName = `contactNameTwo`
        result = 1
    end
else
    if ((`contactNameTwo_Identity` == "UNKNOWN" OR `contactNameTwo_Identity` == "ORGANIZATION")
        AND ( `contactNameOne_Identity` != "NAME" AND `contactNameOne_Identity` != "BLANK"))
        begin
            contactName = `contactNameTwo`
            result = 2
        end
    else
        begin
            contactName = `contactNameOne`
            result = 3
        end
    end
end
```

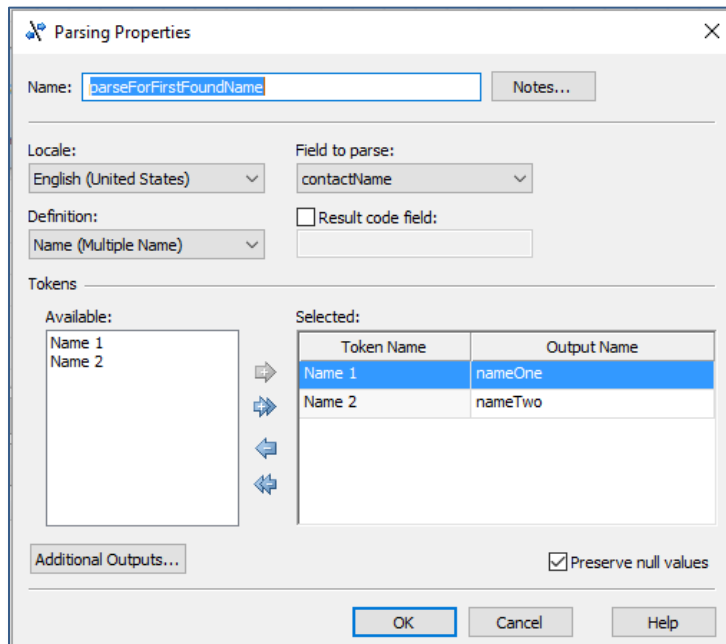
The above code performs evaluations of the results from the Identification Analysis of the contact name fields and determines which holds a person name value returning it in a new field *contactName*.

Figure 8 Results of the Expression to find the person name value



	objectReferenceId	contactNameOne	contactNameTwo	contactNameOne_Identity	contactNameTwo_Identity	contactName	result
1	1	John Doe	ACME Widgets, Inc	NAME	ORGANIZATION	John Doe	0
2	2	ACME Widgets, Inc	John Doe	ORGANIZATION	NAME	John Doe	1
3	3	Best Wishes	John Doe	ORGANIZATION	NAME	John Doe	1
4	4	Master John Doe	ACME Widgets, Inc	NAME	ORGANIZATION	Master John Doe	0
5	5	John and Jane Doe		NAME	BLANK	John and Jane Doe	0
6	6	Mr. John J Doe		NAME	BLANK	Mr. John J Doe	0
7	7	1929342	John J. Doe	UNKNOWN	NAME	John J. Doe	1

Figure 9 Check the contactName field for multiple names



Parsing Properties

Name: Notes...

Locale: Field to parse:

Definition: ☐ Result code field:

Tokens

Available:

- Name 1
- Name 2

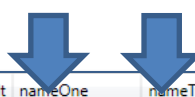
Selected:

Token Name	Output Name
Name 1	nameOne
Name 2	nameTwo

☒ Preserve null values

A requirement of this solution is to identify if multiple people's names exist in a contact name field and to only return the first name found. The Parsing node, with the Name (Multiple Name) definition, will evaluate the *contactName* field and split it into multiple fields when two names are found.

Figure 10 Results of multiple name parsing



	objectReferenceId	contactNameOne	contactNameTwo	contactNameOne_Identity	contactNameTwo_Identity	contactName	result	nameOne	nameTwo
1	1	John Doe	ACME Widgets, Inc	NAME	ORGANIZATION	John Doe	0	John Doe	
2	2	ACME Widgets, Inc	John Doe	ORGANIZATION	NAME	John Doe	1	John Doe	
3	3	Best Wishes	John Doe	ORGANIZATION	NAME	John Doe	1	John Doe	
4	4	Master John Doe	ACME Widgets, Inc	NAME	ORGANIZATION	Master John Doe	0	Master John Doe	
5	5	John and Jane Doe		NAME	BLANK	John and Jane Doe	0	John Doe	Jane Doe
6	6	Mr. John J Doe		NAME	BLANK	Mr John J Doe	0	Mr John J Doe	
7	7	1929342	John J. Doe	UNKNOWN	NAME	John J Doe	1	John J Doe	

Figure 11 Parse the first name found into Name fields

Parsing Properties

Name: Notes...

Locale: Field to parse:

Definition: ☐ Result code field:

Tokens

Available:

- Prefix
- Given Name
- Middle Name
- Family Name
- Suffix
- Title/Additional Info

Selected:

Token Name	Output Name
Given Name	givenNameOne
Middle Name	givenNameTwo
Family Name	lastName
Prefix	Prefix
Suffix	Suffix
Title/Additional Info	Title/Additional Info

Additional Outputs...

☒ Preserve null values

OK Cancel Help

The Parsing node with the Name (Global) definition will extract the name tokens from a provided value.

Figure 12 Results of parsing the first person name found

	objectReferenceId	contactName	origContactNameOne	origContactNameTwo	contactNameOne_Identity	contactNameTwo_Identity	givenNameOne	givenNameTwo	lastName	Prefix	Suffix	Title/Additional Info
1	1	John Doe	John Doe	ACME Widgets, Inc	NAME	ORGANIZATION	John		Doe			
2	2	John Doe	ACME Widgets, Inc	John Doe	ORGANIZATION	NAME	John		Doe			
3	3	John Doe	Best Wishes	John Doe	ORGANIZATION	NAME	John		Doe			
4	4	Master John Doe	Master John Doe	ACME Widgets, Inc	NAME	ORGANIZATION	John		Doe	Master		
5	5	John and Jane Doe	John and Jane Doe		NAME	BLANK	John		Doe			
6	6	Mr John J Doe	Mr. John J Doe		NAME	BLANK	John	J	Doe	Mr		
7	7	John J Doe	1929342	John J. Doe	UNKNOWN	NAME	John	J	Doe			

Figure 13 Use an Expression Node to rebuild the contact name using only selected fields



```
// This expression recreates the contact name field and ensures it only includes the Given  
Names and Last Name, Contact Name should not include Title or Prefix
```

```
hidden string gnOne  
hidden string gnTwo  
hidden string ln
```



```

hidden string cn

hidden boolean gnOneNull
hidden boolean gnTwoNull
hidden boolean lnNull

string recContNameResult

cn = ''
gnOne = `givenNameOne`
gnTwo = `givenNameTwo`
ln = `lastName`

if !isNull(gnOne)
    cn = trim(gnOne)

if len(trim(gnTwo)) > 0
    begin
        cn = cn & ' ' & trim(gnTwo) & ' '
        recContNameResult = 1
    end
else
    begin
        cn = cn & ' '
        recContNameResult = 0
    end

cn = cn & trim(ln)

`contactName` = trim(replace(cn, " ", " "))

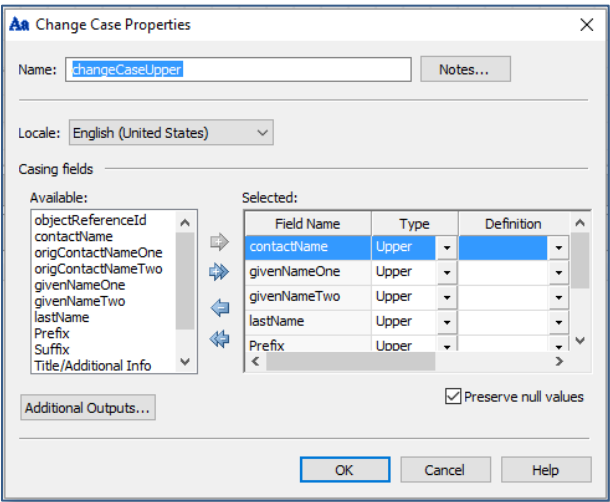
```

The above code creates a contact name value for the person name by evaluating the individual fields of a person's name and adding spacing where appropriate (note similar functionality can be achieved from the Concatenate node; however, the Concatenate node does not allow for trimming of blank spaces)

Figure 14 Results for expression to create new contact name field value

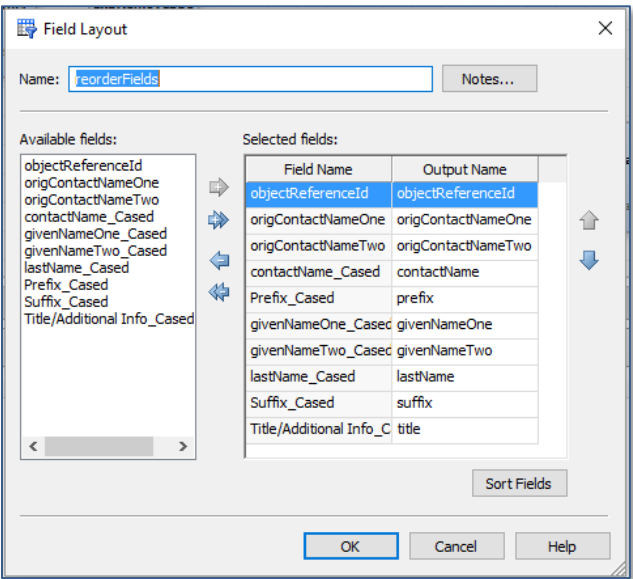
	objectReferenceId	contactName	origContactNameOne	origContactNameTwo	contactNameOne_Identity	contactNameTwo_Identity	givenNameOne	givenNameTwo	lastName	Prefix	Suffix	Title/Additional Info
1	1	John Doe	John Doe	ACME Widgets, Inc	NAME	ORGANIZATION	John		Doe			
2	2	John Doe	ACME Widgets, Inc	John Doe	ORGANIZATION	NAME	John		Doe			
3	3	John Doe	Best Wishes	John Doe	ORGANIZATION	NAME	John		Doe			
4	4	John Doe	Master John Doe	ACME Widgets, Inc	NAME	ORGANIZATION	John		Doe	Master		
5	5	John Doe	John and Jane Doe		NAME	BLANK	John		Doe			
6	6	John J Doe	Mr. John J Doe		NAME	BLANK	John	J	Doe	Mr		
7	7	John J Doe	1929342	John J. Doe	UNKNOWN	NAME	John	J	Doe			

Figure 15 A Change Case node is used to convert all of the fields to uppercase



The Change Case node allows for changing the case of alphanumeric fields in the data job to Proper, Lower and Uppercase formats.

Figure 16 A Field Layout node is used to logically reorder the fields for output



A Field Layout node allows for the reordering and renaming of the fields in the data job. This node is important when communicating with external systems as it allows for the customization of the output into a machine friendly layout.

Figure 17 Final results of the data job to identify a person's name from contact name fields

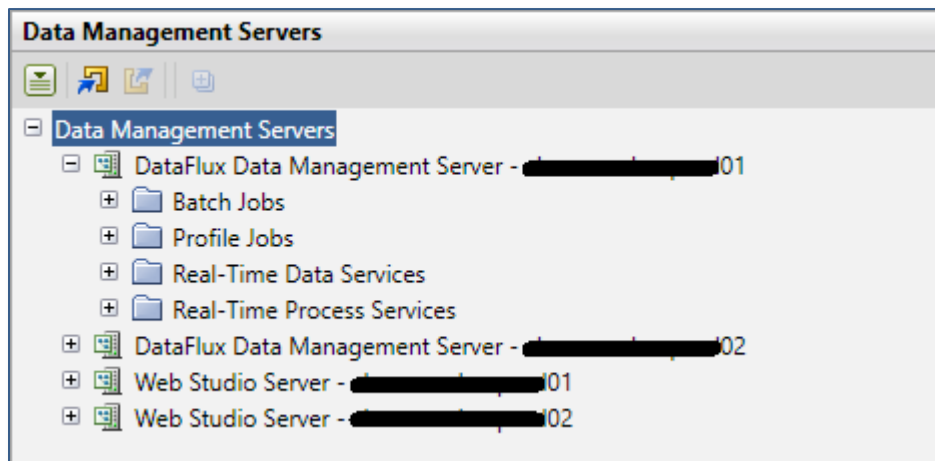
	objectReferenceId	origContactNameOne	origContactNameTwo	contactName	prefix	givenNameOne	givenNameTwo	lastName	suffix	title
1	1	John Doe	ACME Widgets, Inc	JOHN DOE		JOHN		DOE		
2	2	ACME Widgets, Inc	John Doe	JOHN DOE		JOHN		DOE		
3	3	Best Wishes	John Doe	JOHN DOE		JOHN		DOE		
4	4	Master John Doe	ACME Widgets, Inc	JOHN DOE	MASTER	JOHN		DOE		
5	5	John and Jane Doe		JOHN DOE		JOHN		DOE		
6	6	Mr. John J Doe		JOHN J DOE	MR	JOHN	J	DOE		
7	7	1929342	John J. Doe	JOHN J DOE		JOHN	J	DOE		

The results of this data job show the originally inputted values, as well as the resolved Contact Name and Person Name fields.

PUBLISHING TO A SAS® DATA MANAGEMENT SERVER

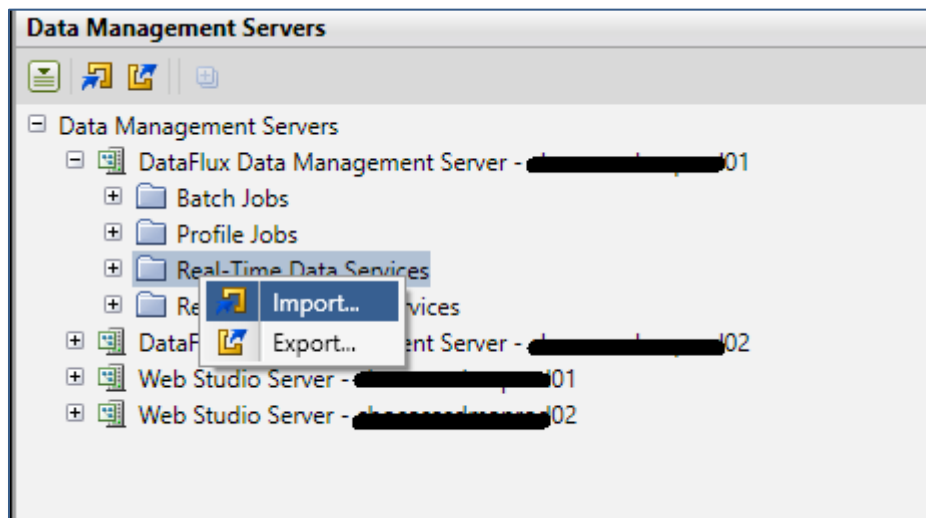
After the data job is created it is only accessible to the local instance of SAS® Data Management Studio, in order for the data job to be accessible to external systems, such as SAS® Data Integration Studio, the data job must be published to a SAS® Data Management Server.

Figure 18 SAS® Data Management Servers connected to a SAS® Data Management Studio



Selecting the Data Management Servers riser from within SAS® Data Management Studio and expand the server where the data job should reside.

Figure 19 Use the Import option to publish the data job



Right-clicking on any folder under the SAS Data Management Server will bring up the Import Wizard which allows you to publish a data job to the server

Figure 20 Select the data job from a repository

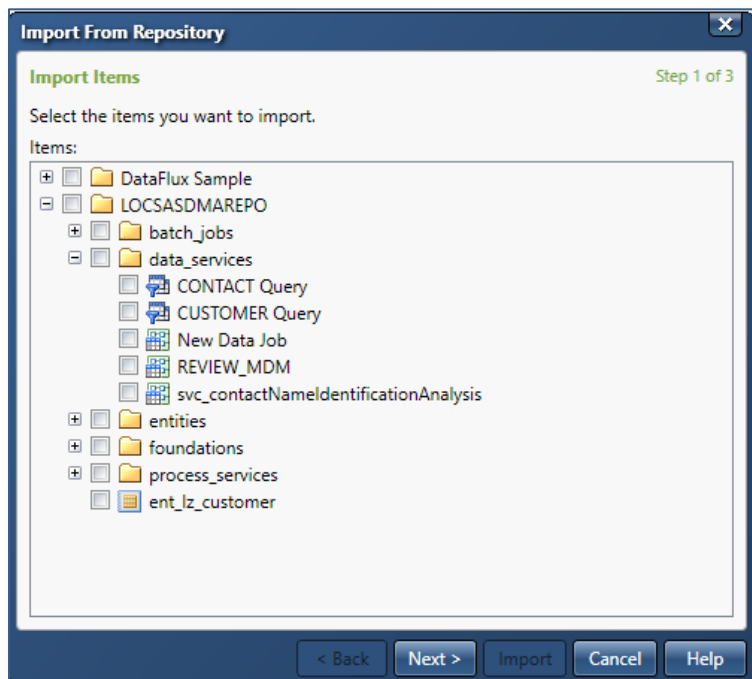
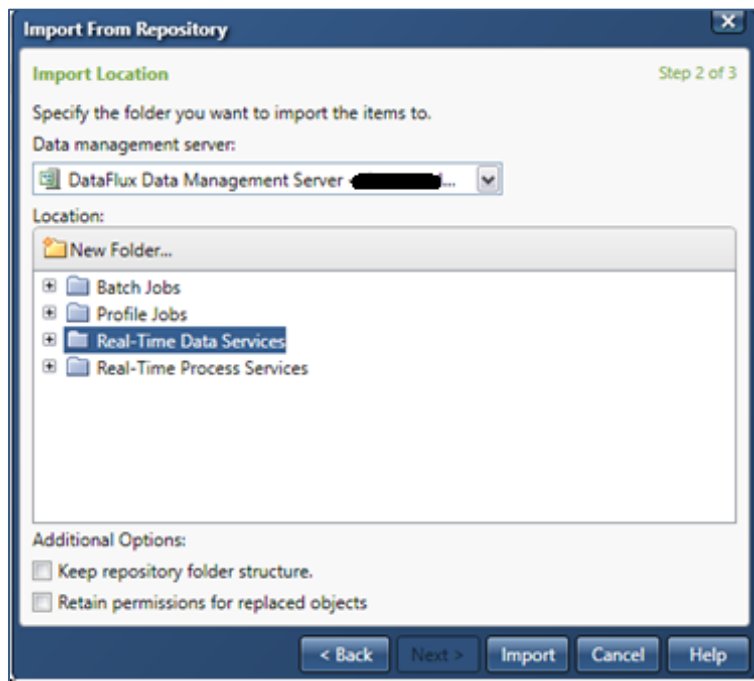
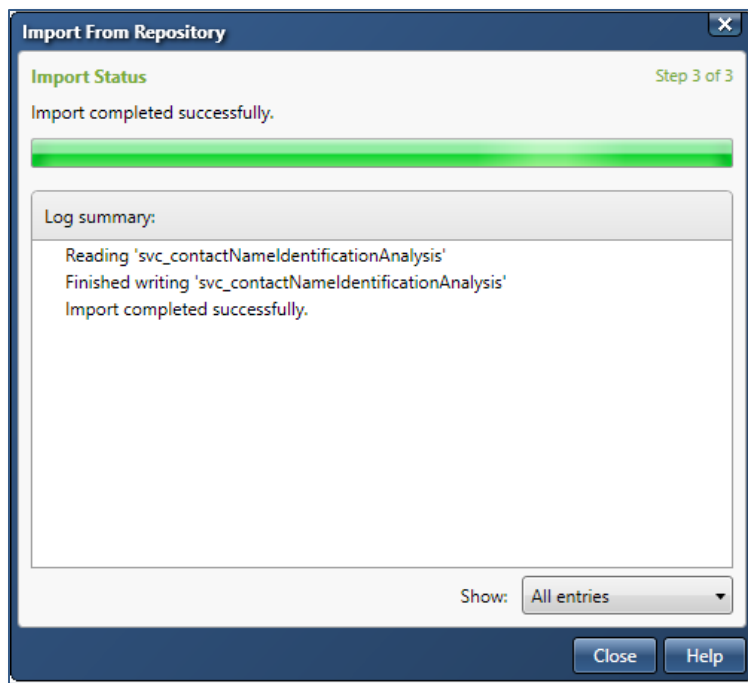


Figure 21 Choose a Data Management Server and Folder to Publish to



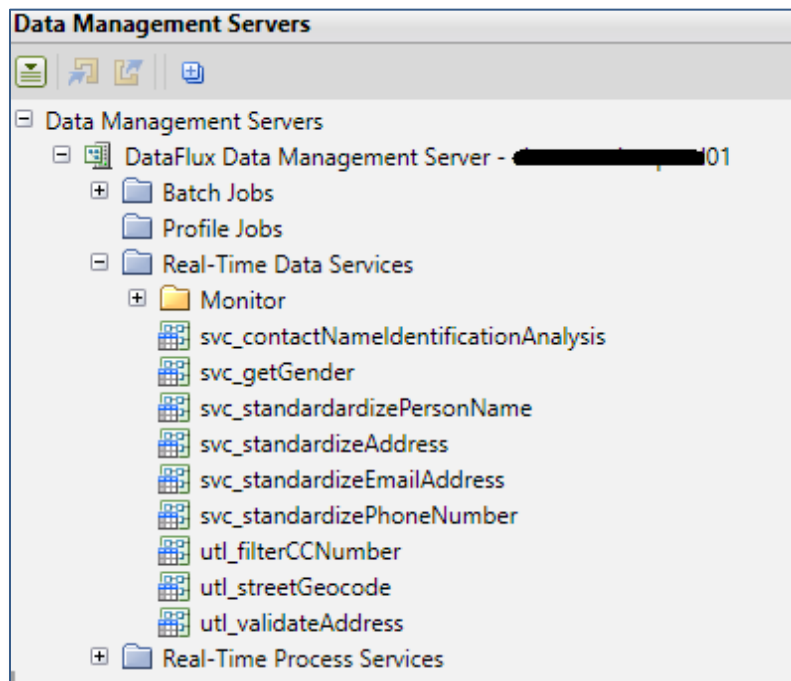
Data Jobs which are intended to be called with input parameters and in a real-time manner must be imported to the Real-Time Data Services folder. Posting data jobs to Batch Jobs folder does not allow for configuring external data inputs and outputs for the job.

Figure 22 Review the details of the deployment



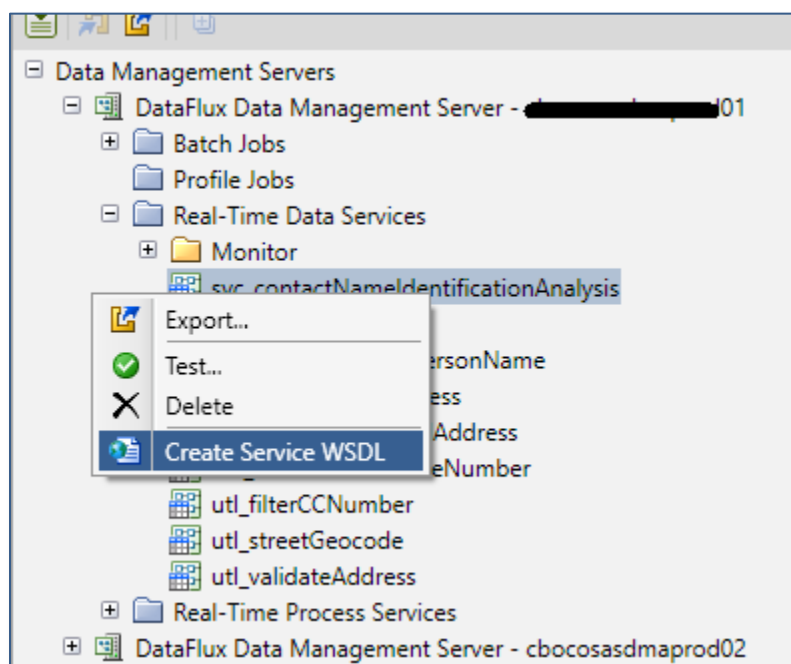
Following the deployment if all goes well a successful message will be posted in the wizard.

Figure 23 The data job has been deployed to the server



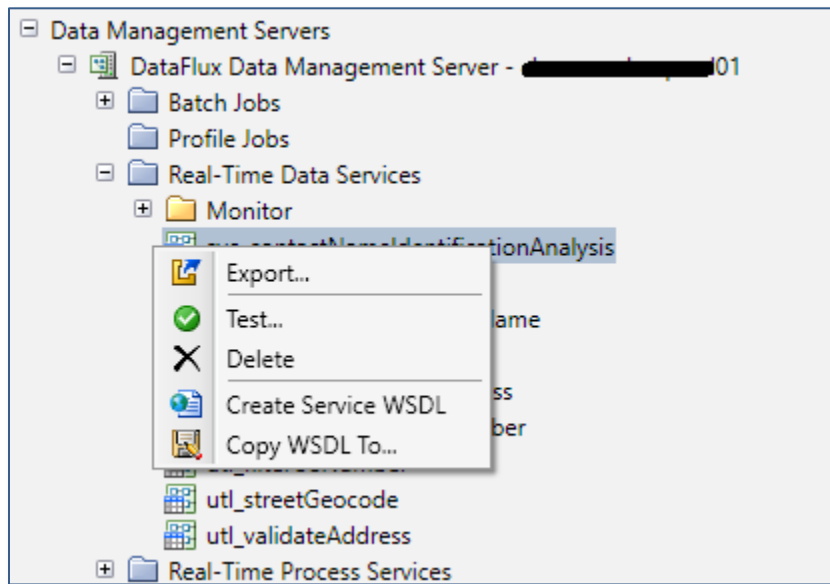
At this point the data job is now available from the SAS® Data Management Server and can be used by authorized users via command prompt execution or through SAS® Data Integration Studio transformation tasks; however, one more step is required to service enable the data job for use by SOAP and REST clients.

Figure 24 Creating a WSDL for a Real-Time Data Service will service-enable the job



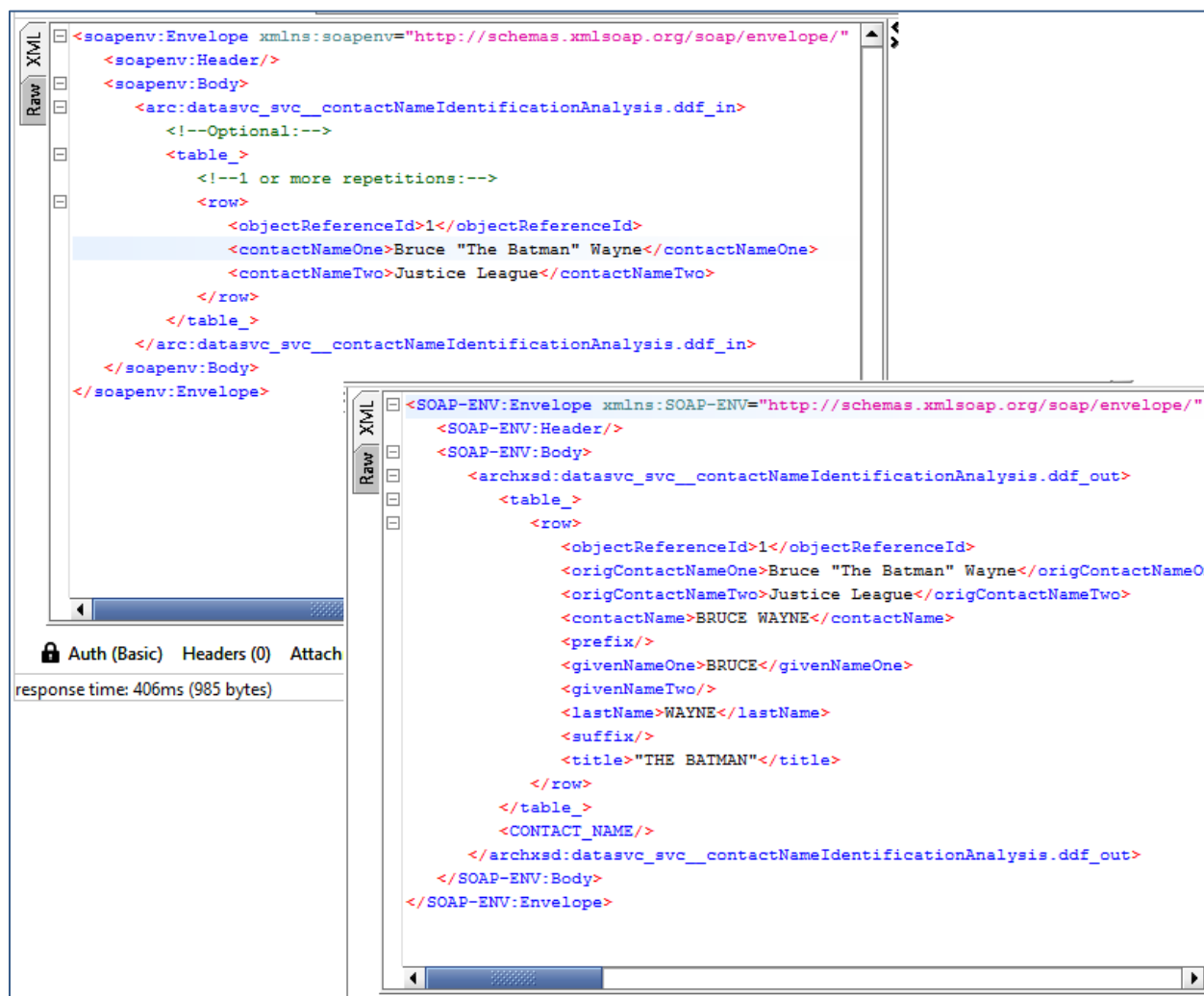
By default, the Real-Time Data Services are not enabled for SOAP and REST-based communications; by creating a WSDL for the data service the data job can now be called externally with SOAP and REST clients.

Figure 25 The WSDL is now created and can be downloaded for use in external service clients



By selecting “Copy WSDL To...” the WSDL file can be copied to a local computer and integrated with external applications.

Figure 26 Example request and response when calling a data job as a SOAP service



The data job now being service enabled can be called using any SOAP client. Note that the request model allows for execution of one or more records by creating additional `<row>` elements.

DATA QUALITY JOBS AS SERVICES

Data Quality is the foundation on which a Master Data Management system is founded. Information hosted by an MDM solution is only as valuable as the quality of the data which underlies it. With the introduction of a new MDM system, an initial challenge is in preparing the data for the first load into the system, a later challenge remains in ensuring the quality and standards of subsequent data loads (batch or transactional). MDM systems focus on master data which tends to have a longer life and is valuable to both transactional as well as analytical systems, therefore data quality must be considered at all times of the lifespan of an MDM system.

Creation of data quality data jobs, such as the example in the previous section allows an organization to enhance its ability to maintain high-quality data standards throughout the lifespan of the MDM as service-enabled SAS® Data Management data jobs can easily integrate with transactional systems through existing SOA technologies. The ability to have data quality jobs exposed as services from SAS® allows organizations to directly tie them into their service-oriented architecture through either an Enterprise Service Bus (ESB) or directly through code-based orchestration.

Service-enabled SAS® data jobs are not limited to the scope of an MDM solution

only. Using SAS® data jobs in conjunction with an ESB allows for sharing of core data quality services between applications, both internal and external, creating a central repository for Data Quality as a Service (DQaaS) within an organization's technology stack. Data quality strategy can be expanded to all real-time data capture systems to ensure the same set of rules is implemented in each with consistency.

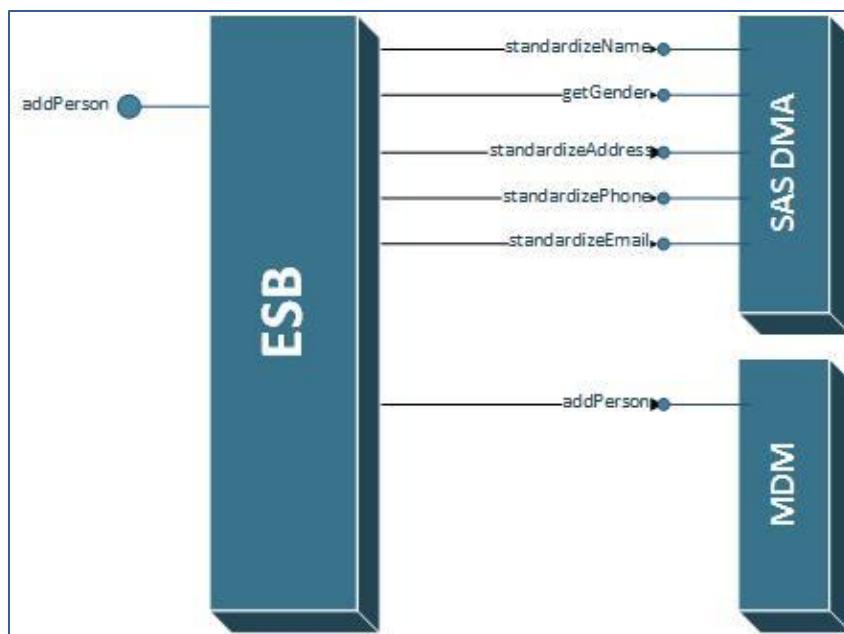


Figure 27 SAS® Data Services can easily be included in an organization's Enterprise Service Bus and tied into business services to provide Data Quality as a Service (DQaaS) for any service oriented transaction.

DATA QUALITY JOBS WITH DATA INTEGRATION STUDIO

The collection of data jobs built in SAS® and deployed to the Data Management Server offer tremendous value when exposed as services and utilized in an organization's SOA; however, they offer additional high value when used in conjunction with a company's data integration strategy as well. As stated earlier, SAS® data jobs, when deployed to a Data Management Server, are also available as transformations in SAS® Data Integration Studio. Implementing the data quality jobs as part of data integration strategy at the ETL level allows an organization to further maintain their data quality strategy at any point where data is moving between systems.

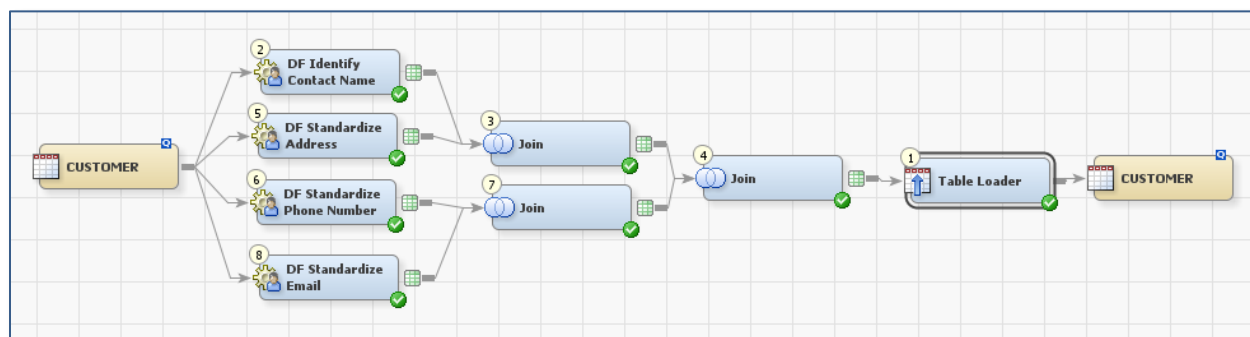
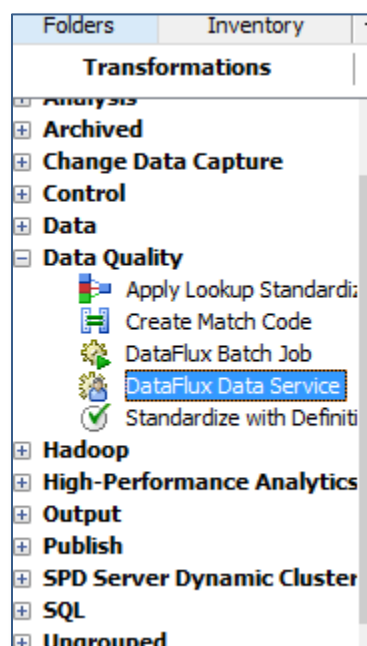


Figure 28 A data integration strategy can be further enhanced with SAS Data Integration Studio when it uses the same DataFlux® data jobs as are implemented at the transactional-level.

Data quality jobs when included in a Data Integration Studio job allow an organization to ensure a consistent set of data quality rules that are shared by both real-time processes as well as ETL processing.

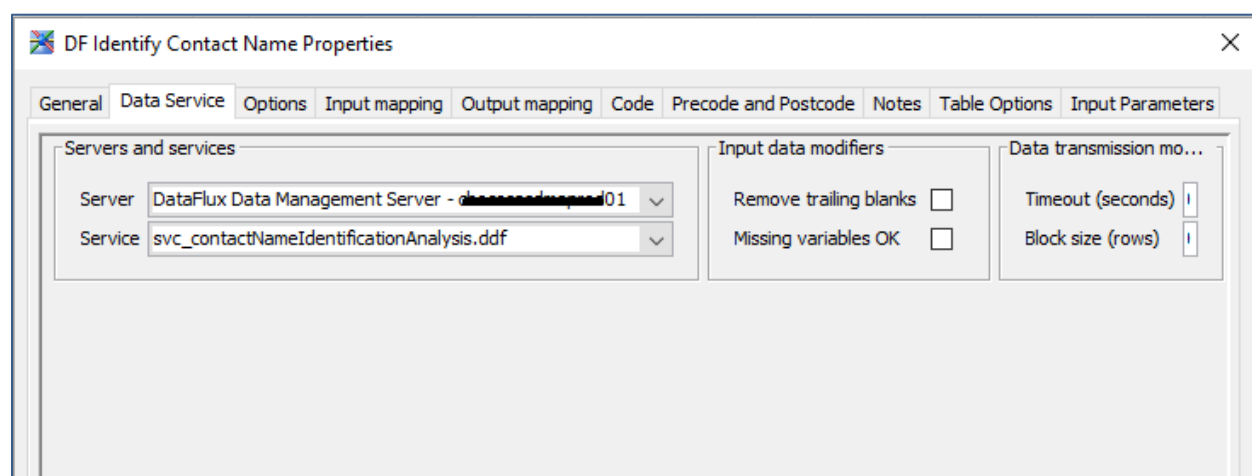
USING A SAS® DATA MANAGEMENT DATA JOB IN SAS DATA INTEGRATION STUDIO

Figure 29 Select a DataFlux Data Service transformation from the Transformations riser



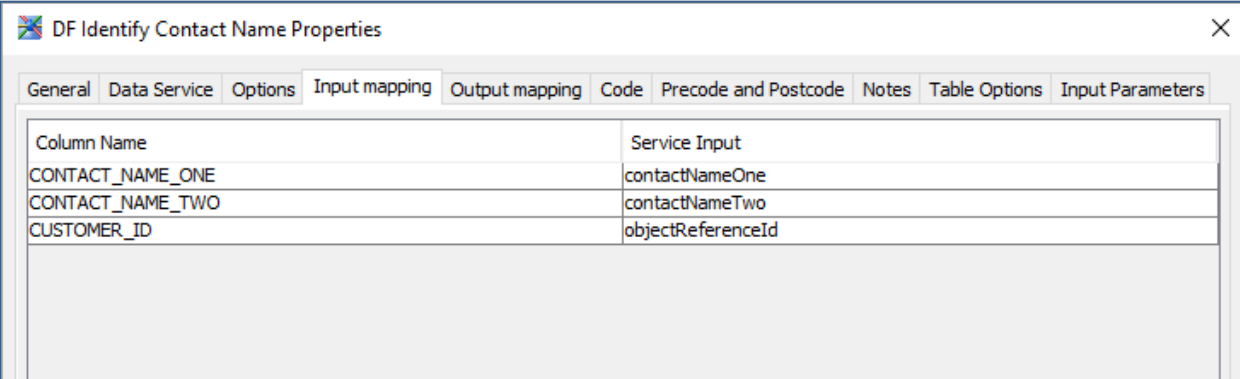
Data Integration Studio supports both DataFlux Data Services as well as DataFlux Batch Jobs, dragging the transformation onto the design panel will add the transformation to the job.

Figure 30 Configure the DataFlux Data Service to reference the service on the Data Management Server



Select the appropriate server where the job is deployed, and select the data job.

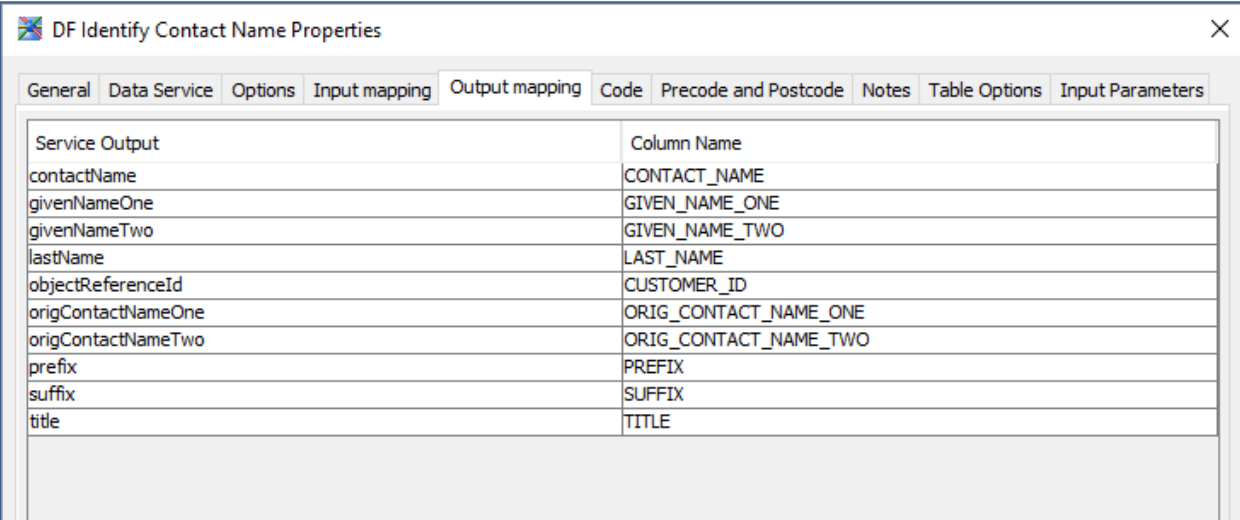
Figure 31 Configure the input mappings



Column Name	Service Input
CONTACT_NAME_ONE	contactNameOne
CONTACT_NAME_TWO	contactNameTwo
CUSTOMER_ID	objectReferenceId

Input mappings are based on the interface specified at the creation of the data job, and the values in the input data source of the DIS job.

Figure 32 Configure the output mappings



Service Output	Column Name
contactName	CONTACT_NAME
givenNameOne	GIVEN_NAME_ONE
givenNameTwo	GIVEN_NAME_TWO
lastName	LAST_NAME
objectReferenceId	CUSTOMER_ID
origContactNameOne	ORIG_CONTACT_NAME_ONE
origContactNameTwo	ORIG_CONTACT_NAME_TWO
prefix	PREFIX
suffix	SUFFIX
title	TITLE

Map the outputs of the data service to the work table for the transformation.

CONCLUSION

Data is a valuable corporate asset that when managed improperly can detract from a company's ability to achieve strategic goals, creating a data quality strategy is essential to the protection of this asset. Using SAS Data Management Advanced with its SAS® Data Management Studio tools allow an organization to design a robust data quality strategy which will ensure data quality at all levels of data integration.

REFERENCES

Dreibelbis, Allen, et al. 2008. Enterprise Master Data Management: An SOA Approach to Managing Core Information. Boston, MA: Pearson plc (Publishing as IBM Press)

Agresta, Ron. 2015. "Master Data and Command Results: Combine Master Data Management with SAS® Analytics for Improved Insights." *Proceedings of the SAS Global Forum 2015 Conference*. Cary, NC: SAS Institute Inc. Available at

CONTACT INFORMATION

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

Brian W. Smith
1-800-Flowers.com, Inc.
bsmith@1800flowers.com
1-800-Flowers.com