



THE
POWER
TO KNOW.

SAS[®] MDM 4.2

User's Guide

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2015. *SAS® MDM 4.2: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® MDM 4.2: User's Guide

Copyright © 2015, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a) and DFAR 227.7202-4 and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513-2414.

March 2015

SAS provides a complete selection of books and electronic products to help customers use SAS® software to its fullest potential. For more information about our offerings, visit support.sas.com/bookstore or call 1-800-727-3228.

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.

Contents

<i>What's New in SAS MDM 4.2</i>	<i>ix</i>
--	-----------

PART 1 Introduction to SAS MDM 1

Chapter 1 / Overview of SAS MDM	3
About SAS MDM	3
How SAS MDM Works	4
Architecture	4
Deployment Models	9
Integrating Other Applications with SAS MDM	9
Chapter 2 / Understanding SAS MDM	11
The Enterprise Environment	11
What Are Entities?	13
How Are Entities Related?	14
How Are Entities Constructed?	14
Inside the Extraction Process	15
Inside the Cleansing Process	16
Inside the Clustering Process	17
Inside the Survivorship Process	22
Inside the Harmonization Process	23
Chapter 3 / Deploying SAS MDM	25
Adding Metadata	25
Generating Jobs	32
Loading the MDM Hub	34
Loading Data	40
Migrating from a Master Data Management Foundation Hub	41
Chapter 4 / Controlling Access	43
About Access Control	43
Defining Groups	46
Interaction with DataFlux Data Management Server Permissions	47
Managing Access	47
Designing Jobs	47

PART 2 SAS MDM Interface 49

Chapter 5 / Accessing SAS MDM	51
Log On to Data SAS Management Console	51
SAS Data Management Console Home Page	52
Specifying Your Preferences	54

Overview of the SAS MDM Interface	56
---	----

PART 3 SAS MDM Components 59

Chapter 6 / Using Master Data Management	61
About Master Data Management	61
Searching Entities	62
Working with Search Results	65
Editing Entities	68
Records Tab	69
Hierarchies Tab	74
Relationship Diagram Tab	74
History Tab	78
Creating New Entities	78
Creating a Remediation Issue	80
Chapter 7 / Using Hierarchies	83
Searching Hierarchies	83
Chapter 8 / Using Source Systems	87
About Source Systems	87
Working with Source Systems	88
Chapter 9 / Using Master Data Management Data Model	93
About Master Data Management Data Model	93
Understanding Entity Type Definitions	96
Working with the Entity Type Editor	96
Understanding Relationship Type Definitions	113
Working with the Relationship Type Editor	113
Understanding Hierarchy Type Definitions	120
Working with the Hierarchy Type Editor	121
Chapter 10 / Using Master Data Management Administration	127
About Master Data Management Administration	127
Working with the Source Systems Editor	129
Working with the Language Editor	135
Chapter 11 / Working with SAS Visual Analytics	139
SAS Visual Analytics Reports	139
Creating and Viewing Reports for SAS Visual Analytics	141
Database Connections to the SAS MDM Database	143

PART 4 SAS MDM Examples 145

Chapter 12 / Working with SAS MDM	147
Example 1: Creating an Abstract Entity Type	147
Example 2: Creating a Concrete Entity Type	164
Example 3: Creating Entity Relationships	172

<i>Recommended Reading</i>	181
<i>Glossary</i>	183
<i>Index</i>	185

What's New

What's New in SAS MDM 4.2

Overview

The main enhancements and changes for SAS MDM 4.2 include the following:

- reports for SAS Visual Analytics
- source system management functionality
- process control enhancements
- relationship type attributes
- cross-field clustering
- source system harmonization
- documentation enhancements

Reports for SAS Visual Analytics

SAS MDM provides reports and associated metadata that enables data stewards and business users to gain more insight on the type and frequency of master data in the SAS MDM hub. You can see batch load statistics, record counts by entity type and source system, and many other aspects of the data in the SAS MDM database.

Note: SAS Visual Analytics, available as a separate offering, is required to view these reports.

Source System Management Functionality

New management functionality has been added to make better use of source system definitions in the SAS MDM environment. Data Stewards can control which entity types are to be contributed by a source system. They can also control how and under what circumstances master data should be harmonized

to contributing systems for consistency purposes. Source system harmonization provides the ability to execute jobs when new survivors are created, thereby enabling contributing sources to be harmonized with the hub.

Process Control Enhancements

The ability to monitor SAS MDM batch data loads through a new Source System component makes it easier to understand the flow of data into your SAS MDM system. A snapshot view clarifies the times and volumes of data load processes. When enabled, user-designed Source System tools can be invoked to initiate any number of pre-defined processes, such as pushing data to a reporting environment or synchronizing data with other systems.

Relationship Type Attributes

This feature provides the ability to model attributes on relationships.

Cross-Field Clustering

This feature provides the ability to match against multiple variant fields. For example, this is useful when multiple emails should be considered when clustering.

Source System Harmonization

This feature provides the ability to push data from the hub to data sources as specified by the Source System Harmonization setting for an entity type. Changes made in the hub that result in a new survivor record can be harmonized with the configured source systems.

Documentation Enhancements

The SAS Data Remediation and SAS Task Manager content that was previously available in this document has been moved to *SAS Data Remediation: User's Guide* and *SAS Task Manager: User's Guide*.

Part 1

Introduction to SAS MDM

<i>Chapter 1</i>	
<i>Overview of SAS MDM</i>	3
<i>Chapter 2</i>	
<i>Understanding SAS MDM</i>	11
<i>Chapter 3</i>	
<i>Deploying SAS MDM</i>	25
<i>Chapter 4</i>	
<i>Controlling Access</i>	43

1

Overview of SAS MDM

<i>About SAS MDM</i>	3
<i>How SAS MDM Works</i>	4
<i>Architecture</i>	4
Overview	4
Development	5
Run Time	5
SAS MDM Components	6
<i>Deployment Models</i>	9
<i>Integrating Other Applications with SAS MDM</i>	9

About SAS MDM

SAS MDM is a combination of software, templates, documentation, data models, and services that provide the functionality and processes necessary to build and maintain a master entity database. An entity can be a customer, product, patient, site, or any business data object that you define. You can also define attributes for each entity, as well as transformations that support data quality and identity management functionality.

The SAS MDM hub is a set of tables and table relationships that contain entity information and required entity keys and linking information. This hub provides end-users or other applications with a complete view of an entity and its relationships with other entities. In addition, the hub can be used as the single source for creating and maintaining survivor records that can be used in other applications or processes. The original source system IDs, which are important to other enterprise applications or data sources, are also maintained in the hub to facilitate linking activities with other data providers or consumers.

Note: SAS MDM is integrated with DataFlux Data Management Platform. Therefore, to use SAS MDM successfully, you must be familiar with DataFlux Data Management Platform.

SAS Data Remediation enables users to manage and correct issues triggered by business rules in SAS MDM batch jobs and real-time processes. These issues can also be triggered by external systems that communicate with data remediation through its REST-based web service application programming interface (API). Data remediation allows user- or role-based access to data exceptions, which are categorized by application and subject area. Once data remediation issues have been reviewed, they can be corrected through the same application, eliminating the need for another user to complete the

correction process. All data remediation issues can also be associated with workflow definitions that route the issues to the correct decision maker for approval or instructions for additional action.

SAS MDM includes the following features:

- batch and real-time modes
- capability to register user-defined tools that operate on the data in the hub
- entity matching, de-duplication, and survivor record creation
- entity linking through relationships
- data quality functionality such as address verification, standardization, and parsing
- the ability to search based on fields within the records
- capability of adding, modifying, and retiring SAS MDM entities
- workflow-based data remediation functionality to view and correct SAS MDM data errors
- metadata management for entities, attributes, and transformations
- historical tracking, including changes saved for contributor records and survivor records
- data access, data profiling, data monitoring, data quality, and data enrichment

How SAS MDM Works

SAS MDM performs the following functions:

- extracts business information from your data sources
- validates and standardizes the data
- captures data errors through user-defined business rules and sequesters the data for review and correction
- consolidates the information into a single view of the information available from all the data sources

The organizational data that you provide can be customer data, product data, services data, or data for other entity types. SAS MDM applies a rigorous methodology to the problem of integrating disparate enterprise data.

Architecture

Overview

SAS MDM is built on a sophisticated data management platform that provides both development and run-time support for applications.

Development

During development, DataFlux Data Management Studio provides the ability to design and test processes (batch jobs) and real-time services (data jobs), which can then be deployed and run using the following:

- DataFlux Data Management Server
- DataFlux Data Management Studio
- scripting with a command line interface

The DataFlux Data Management Platform relies on the Quality Knowledge Base (QKB) for data quality processing. The QKB provides capabilities such as standardization of data and computation of match codes. The QKB is used both during development and at run time.

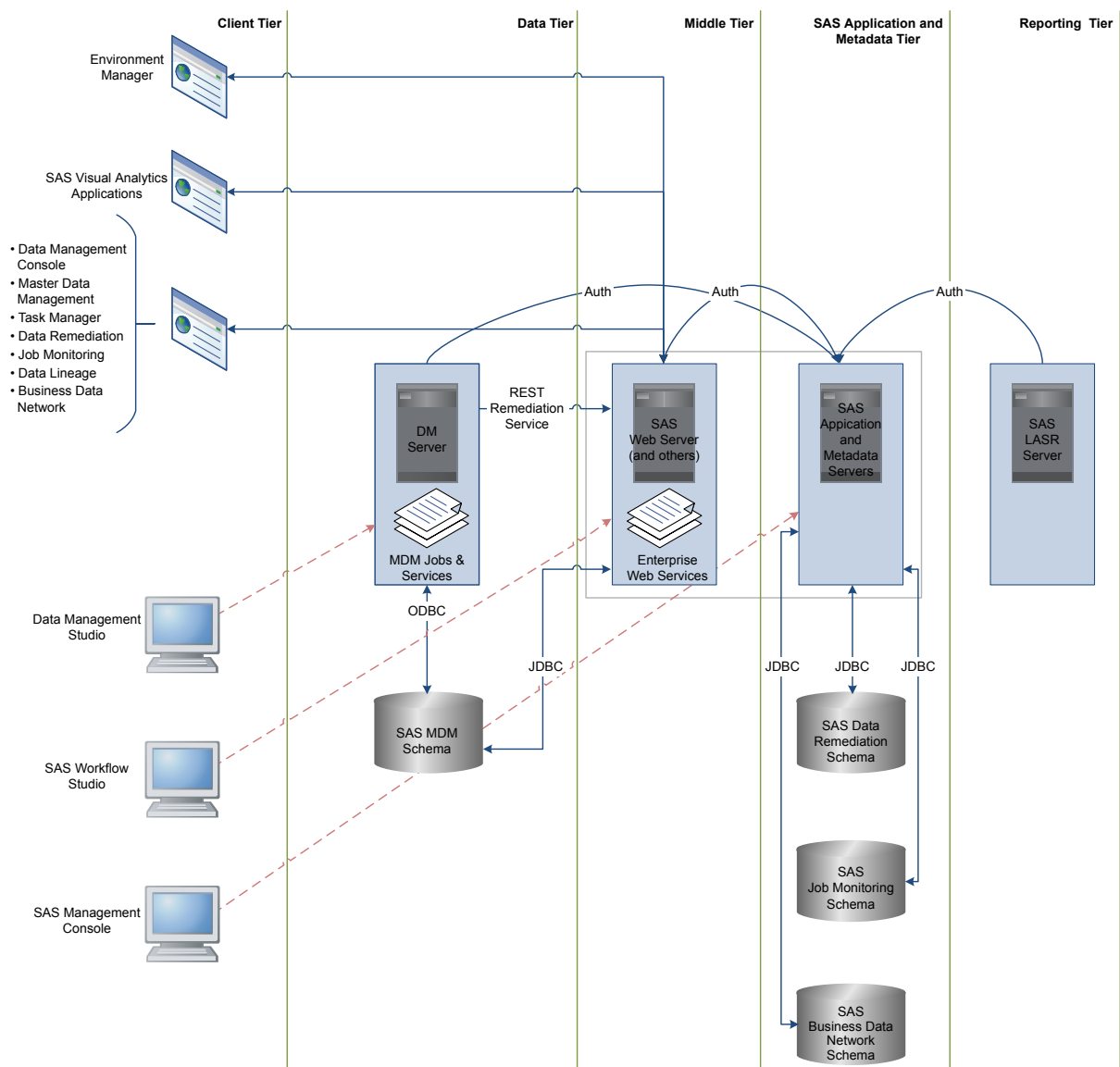
Run Time

At run time, a multi-tier architecture that consists of presentation, business, and data tiers is used.

- The presentation tier hosts the web interface in a Servlet 2.5-compliant container and includes a Java application server and a browser. Batch interfaces might be available for some operations such as bulk loads and updates.
- The business tier runs on a DataFlux Data Management Server and is implemented using DataFlux Data Management Studio processes (batch jobs) and real-time services (data jobs). It implements a Service Oriented Architecture (SOA) that enables any system to use its services through SOAP requests. Common web service standards are used to maintain simplicity when integrating with other systems. Authentication within the business tier runs on a SAS Metadata Server. This server can authorize user permissions for applications, assign roles and security restrictions, and define authentication parameters for accessing other systems, such as the database, for each user or group. The real-time or transactional component of SAS MDM interfaces with other applications through an SOA environment. Specifically, the ability to call business services hosted on DataFlux Data Management Server is accomplished through web services. Process jobs can also be used as business services that accept data on a record-by-record basis. This enables users to use SAS MDM data and services within an enterprise architecture.
- The data tier hosts the supporting data and metadata for the system, which consists of a relational database such as Oracle or Microsoft SQL Server. The data model is documented and accessible to third-party tools and can also be accessed through DataFlux Data Management Studio jobs.

The following display shows how the components fit together:

Figure 1.1 Architecture Diagram



SAS MDM Components

DataFlux Data Management Studio

DataFlux Data Management Studio is a graphical user interface (GUI)-based application that provides access to other tools and services. You can use DataFlux Data Management Studio for job flow management and customization, to manage data source connections, to manage user accounts through SAS Metadata Server, to deploy and execute jobs on the DataFlux Data Management Server, and to run batch jobs and data services within a Windows environment.

DataFlux Process Jobs and Real-time Business Services

A DataFlux process job is a multi-step data management process that can be run from DataFlux Data Management Studio, from DataFlux Data Management

Server, or from a command line interface or script. Real-time business services, also called data jobs, are deployed through DataFlux Data Management Server and are used by the SAS MDM web application to provide data to the application through SOAP calls.

Quality Knowledge Base

The Quality Knowledge Base (QKB) contains the files, file relationships, and metadata needed to correctly parse, match, standardize, and otherwise process data as required for SAS MDM. The QKB can be customized to meet an organization's needs. The QKB can support different locales to support processing for many different languages.

DataFlux Data Management Server

The DataFlux Data Management Server (labeled as DM Server in [Figure 1.1](#)) enables data management and integration processes to be deployed across an entire IT environment. SAS MDM uses batch jobs and real-time services deployed through DataFlux Data Management Server to manage the master data hub and to provide data through the web interface and reports. DataFlux Data Management Server handles all aspects of database connectivity and service availability through SOAP calls.

Note: In many instances, you should have at least two DataFlux Data Management Servers working together through network load balancing or failover hardware or software, to achieve the desired results for performance and meet server uptime goals, particularly for real-time service processing.

Databases

SAS MDM relies on access to a Database Management System (DBMS, labeled as SAS MDM Schema in [Figure 1.1](#)), such as Oracle or SQL Server, to act as a repository for metadata and data. The Master Data Management component of SAS MDM requires two database schemas for its operation.

SAS Data Management Console

The SAS Data Management Console is a central, web-based location for all master data management activities. This environment provides a launch location for all associated SAS MDM applications and SAS Visual Analytics reports, as well as other SAS applications.

The SAS MDM applications Master Data Manager, Data Remediation, and Task Manager enable you to perform the following functions:

- Master Data Manager
 - ☐ create and manage a hub
 - ☐ define new entity types and their attributes
 - ☐ search for data within the hub
 - ☐ create and update data within the hub
- Data Remediation
 - ☐ monitor and correct errors in data loaded into SAS MDM
- Task Manager

- ☐ start, stop, view, and interact with active workflow instances
- Data Lineage (not available in all SAS MDM product bundles)
 - ☐ view relationships and perform impact analysis for selected objects stores in SAS metadata
- Business Data Network (not available in all SAS MDM product bundles)
 - ☐ define and manage metadata for business terms

Other applications might appear in your SAS Data Management Console depending on the SAS MDM offering.

Several of these web applications use SAS data server repositories to store application data. These are represented in [Figure 1.1](#) as schemas in the SAS Application and Metadata Tier.

SAS Workflow Studio

This application is a drag-and-drop workflow designer. Users can modify existing MDM workflow templates or create new ones for use with other SAS MDM processes. Workflows can also be uploaded and activated in the SAS server environment and can be downloaded for additional editing.

SAS Visual Analytics Applications

If your SAS MDM product bundle includes reporting capabilities, business users and data stewards have several reporting applications available to them to design, view, and share reports for SAS MDM and SAS Data Remediation. See the documentation for SAS Visual Analytics for more information about application capabilities.

Environment Manager

This application enables an administrator to see in one location key operational metrics of all SAS servers and other technologies deployed with SAS MDM. This application also includes server configuration and management capabilities.

SAS Web Server

This server handles web traffic and communication for all SAS web applications running in the mid-tier environment.

SAS Application Server

An application server hosts the SAS Data Management Console and other associated applications. This is installed and configured during the deployment process. No other third-party application servers are supported.

SAS Metadata Server

This server makes important system wide metadata available to SAS clients and other SAS servers. Interaction with this server is accomplished through SAS Management Console.

SAS LASR Analytic Server

If your SAS MDM product bundle includes reporting capabilities, then SAS LASR Analytic Server is present in your SAS MDM environment. You can load SAS MDM instance data to SAS LASR Analytic Server using SAS/ACCESS engines. This loads the data into memory making it available to SAS Visual Analytics applications.

Normal MDM functionality does not require an access engine. However, if your SAS MDM product bundle includes reporting capabilities delivered through SAS Visual Analytics, you must have the following installed on the computer that hosts SAS LASR Analytic Server:

- for the SAS MDM reports, either SAS/ACCESS to Oracle if your SAS MDM target database is Oracle or SAS/ACCESS to ODBC if your SAS MDM target database is Microsoft SQL Server
- for the SAS MDM reports, SAS/ACCESS to Postgres

SAS Management Console

This application enables users to manage groups, users, and roles on the SAS Metadata Server for all SAS MDM web applications. It also is the location where various web application configuration parameters are set.

Deployment Models

The most common type of deployment for SAS MDM is the physical deployment model. This model assumes that all the information necessary to construct a complete survivor record for an entity is stored in the hub. Source system keys are also stored in the hub so that information used to identify and describe entities can be recorded. This supports the ability to link or reference source system data, provides traceability for the contributor records, and describes key relationships with the data as it exists in the hub.

With this approach, uniquely identifying information for entities is stored in the hub, along with other dependent attributes. You can access the survivor record by querying the hub by itself, without the need to refer to the systems of record. This approach simplifies the task of constructing the survivor record, but results in higher storage requirements.

The physical deployment model is the primary implementation, but SAS MDM also works in conjunction with other technologies to provide functionality that supports deployment styles such as hybrid MDM, registry-based MDM, reference-style MDM, and other integration scenarios.

Integrating Other Applications with SAS MDM

You can integrate other end-user or enterprise applications, as well as other Extraction, Transformation, and Loading (ETL) software with SAS MDM.

Two methods for integration are as follows:

- direct-database or flat-file access, running batch load, or batch update processes. In this case, the DataFlux Data Management Server can connect directly to data sources, transform the data, and load it into the hub.
- communicating with the DataFlux Data Management Server using web services. Any application or process that can make a web service call can send data to the hub through DataFlux Data Management Server and receive data from the hub in the same manner.

Both methods are typically used to interact with SAS MDM. In some cases it might make sense to communicate directly with DataFlux Data Management Server from an application integration layer of an enterprise application such as Siebel or SAP. In other cases, using an enterprise messaging mechanism such as TIBCO or webMethods might be more practical if that infrastructure has already been deployed within your organization.

You also have the option of making web service calls directly from a .NET or Java component or application that can use web services. This might be common in circumstances where internally designed and built operational or reporting processes need to interact with the DataFlux Data Management Server to access customer or service data inside the hub.

2

Understanding SAS MDM

<i>The Enterprise Environment</i>	11
Which Data Is Most Important to Your Business?	11
Data Profiling	12
<i>What Are Entities?</i>	13
<i>How Are Entities Related?</i>	14
Relationships	14
Hierarchies	14
<i>How Are Entities Constructed?</i>	14
<i>Inside the Extraction Process</i>	15
Overview	15
The Extract Parameter	15
Handling Invalid Data	16
<i>Inside the Cleansing Process</i>	16
Validation Rules	16
Enabling Data Quality	16
<i>Inside the Clustering Process</i>	17
Unique and Related Entities	17
Improving Clustering Results	20
Using Match Codes	20
<i>Inside the Survivorship Process</i>	22
<i>Inside the Harmonization Process</i>	23

The Enterprise Environment

Which Data Is Most Important to Your Business?

It is important to understand the environment in which SAS MDM is to be deployed. The source systems that provide the data for SAS MDM have a large impact on the implementation choices that you make.

The following types of data should be considered for incorporation into SAS MDM:

- data that represents the nouns of your business. Customers, Partners, Products, and Services are examples of entities that many organizations can have in common.

- data that exists in multiple silos. It does not matter if the silos are a result of acquisition or in-house development. Any redundant data can pose a risk to your organization.

When starting a master data management initiative, your business needs must be considered first and foremost. All of your data does not need to be managed as master data.

The following types of data should not be considered for incorporation into SAS MDM:

- data that rarely changes. Reference data is easier to synchronize across the enterprise because it rarely changes.
- data that is not shared. If a data set is used by only one part of your business, the merge and reconciliation aspects of a master data management initiative are less valuable.

Data Profiling

It might not be obvious where the entities in your organization are defined or are being used. Data profiling can help you discover which enterprise systems contain entity information and the attributes, including identifiers, associated with it. With this information, you can understand more about your master data management implementation.

Consider the following questions related to data profiling:

Table 2.1 Data Profiling

Question	Discussion
Which source systems are involved and which entities do they manage?	In SAS MDM, entities are constructed from records in source systems where the data originally resides. Some examples of source systems are Enterprise Resource Planning (ERP) systems, accounts receivable systems, and payroll systems. Data profiling can help discover similarities among the source systems in an enterprise, thereby suggesting entities for incorporation into the hub.
How are the entities identified?	By identifying the keys used within each source system, you can identify strategies for matching the same entity across different source systems.
What attributes do the entities manage?	As you reconcile the views of an entity that have been adopted by each source system, you can begin to formulate the data needs of the entity.
What steps are needed to address quality issues in each of the systems?	Standardization of data elements within the hub provides better entity results and a consistent view of the data.

Question	Discussion
Do the different systems provide different levels of accuracy and reliability?	A customer-facing system might be more accurate and reliable than third-party enrichment data that is only occasionally refreshed. In this case, survivorship rules can be implemented to capture the best data from the most trustworthy system.
How big will the final data set be?	Addressing operational issues, such as database size and processing time, are critical for success.
How frequently does the data change in each system?	Especially in the case of Extraction, Transformation, and Loading (ETL) scenarios, where the hub is updated at regular intervals, it is important to know the latency associated with these updates. Is it acceptable for the hub data to lag behind the source system by a week? Or must the hub data always be up-to-date, thereby requiring a real-time interaction between the systems? Knowing the answers to these types of questions ensures that you have realistic expectations from the SAS MDM system.

What Are Entities?

A SAS MDM entity is a survivor record that is constructed from contributing data from the source systems in the enterprise. This section explains more about entities within the hub.

An entity has a type, where the types represent the nouns of your business. For example, an entity can be an individual, organization, product, or service.

SAS MDM does not pre-define which data your entities require. Instead, SAS MDM accepts metadata about your entities and modifies the hub accordingly.

This metadata includes the following:

attributes

data elements used by entities. Attributes are analogous to columns in a database table. Examples include addresses, e-mail addresses, and dimensions. Attributes have metadata that describes them, such as data type, length, and whether the attribute is a required value for a particular entity.

cluster conditions

rules used to determine which records contribute to the survivor record.

The metadata is collectively known as an entity type, which describes the information that is used by all the instance data for that type. Parent entity type references enable child entities to inherit definitions from their ancestors. For example, the top-level SAS MDM entity type (named ENTITY) defines a set of attributes that are used to maintain instance data in the hub. Because every other entity type derives (either directly, or through other entity types) from the ENTITY entity type, modeling these attributes in the derived types is

unnecessary. Attributes, cluster conditions, and relationship types are all inherited by derived entity types.

The metadata is used in several ways in the operation of the hub:

- Entity type information (including the attributes) is used to construct tables in the hub, where each attribute is expressed as a column. These tables are regular database tables and can be used by reporting and other line-of-business applications.
- Cluster conditions are used to generate batch jobs and data services that drive the operation of the hub. You can use these services to add, query, and retire information in the hub.
- Metadata is used at run time to determine how to present the contents of the hub, including the layout of forms and tables.

How Are Entities Related?

Relationships

Entities often maintain relationships among one another. For example, Companies employ People, or Cities are part of a Geographic region. The hub supports relationships between entities by modeling binary relationships between two entity types. A relationship type defines one entity type as the "source" of the relationship, and the other entity type as the "target" of the relationship. Each relationship type can describe conditions that enable SAS MDM to determine whether two entities are related.

Attributes that are unique to the relationship can be modeled on the relationship type and are available for editing within the SAS MDM user interface.

Hierarchies

Relationships can also be used to construct hierarchies of entities. Hierarchies provide a tree-structured view of the relationships between entities, based on the types of relationship types that you specify when designing the hierarchy.

How Are Entities Constructed?

Each entity in the hub is represented by a set of records, loaded from source systems, that have undergone the following processes:

extraction

Extraction works with the system of record to query data from the source and format it so that it can be loaded into SAS MDM.

cleansing

Data quality, standardization, and other transformations are applied to the incoming records to ensure that the data meets business requirements. The cleansing process removes irregularities, standardizes values so that records can be compared, and computes match codes, which are used in the

clustering process. Records that do not meet (or cannot be made to meet) the standards are redirected for further processing.

clustering

Records that represent the same entity are clustered based on criteria specified by the entity type. The criteria include sets of attributes to identify records that contribute to an entity. For example, if two customers have the same name, e-mail, and address, you can create a cluster condition that specifies that the two customers should be considered as part of the same entity.

survival

Records in the cluster are used to construct a single survivor record (also known as the best record or golden record) based on rules that are specified by the entity type.

harmonization

The construction of a new survivor is an event that, optionally, can, notify or update the source systems in your environment. The event can be used to harmonize contributing sources with the contents of the SAS MDM hub.

The following sections provide additional details about each of these processes.

Inside the Extraction Process

Overview

The extraction process is responsible for obtaining data from source systems and preparing it for processing by the hub. By default, SAS MDM generates an extraction data service that can be configured to connect to your source system. The extraction data service (`mdm_extract_<entity type>.ddf`) is invoked as part of the processing of the `add_update_<entity type>.djf` batch job.

The Extract Parameter

As part of the operation of the extraction process, the extraction data service is provided with a parameter that can be used to control its operation. After the service runs, it can provide a new value for the parameter that is available to the service on its next invocation. The extract service uses this parameter like a bookmark, allowing it to load some data during one run, and then resuming with subsequent data on its next execution.

SAS MDM does not interpret the value of the extract parameter. This is done by the extract service itself. For example, the value of the extract parameter might be:

- the last key value that was processed
- the timestamp of the latest record that was read
- some other data capture identifier

Handling Invalid Data

Most organizations have business rules that define the suitability of data for various business purposes. For example, an e-mail address might be a required data element in a customer record on which other business processes, such as marketing or customer billing, rely. If the e-mail address is not present in a customer record, the omission might have ripple effects across the business.

Using data remediation, invalid data can be captured before it goes into SAS MDM or after it has been loaded into SAS MDM and combined with other data. Business rules can be built into SAS MDM to identify incorrect or invalid data and isolate it from other data until data administrators can review and correct the data issues and return the data to the load process. If additional processes are needed for correction and approval, workflows can be used to route the data through different groups before it is finally committed to the SAS MDM database.

Inside the Cleansing Process

Validation Rules

The cleansing step is a key component of SAS MDM. The hub keeps a clean, standardized version of each core attribute that matches the semantics of the attribute type. The possibilities for validation rules are unlimited. A rule can accomplish simple tasks such as determining whether a field is null, or more complex tasks such as cross-field calculations.

The cleansing process is implemented in a data service named `mdm_std_<entity type>.ddf`, and is referenced from the `add_update_<entity type>.ddf`, `mdm_query_survivor_<entity type>.ddf`, and `mdm_query_relationship_<entity type>.ddf` jobs.

The following are examples of logical validation rules for SAS MDM:

full name

modifies the name field to use proper case.

address

performs address verification, including augmenting the record with missing fields that can be provided by third-party services or databases.

telephone

removes records with no area code.

Enabling Data Quality

SAS MDM can perform data cleansing and standardization by using an extensive suite of data profiling and data quality nodes, including the Quality Knowledge Base (QKB). Standardized values and other transformed attributes can be stored with the original data in the hub. For more information about the available standardization schemes, see the DataFlux Quality Knowledge Base online Help. For the QKB to function, the record must be associated with a locale that guides interpretation of the data. This is provided by the

mdm_language_id attribute, which is part of every entity type. The ID is used to look up the QKB locale based on its value.

In addition to standardizing values, you can use the QKB to compute match codes that assist in clustering similar records into a survivor record. Match codes must be modeled as attributes and become part of the entity record as part of data ingestion processing. For more information about match codes, see [“Inside the Clustering Process” on page 17](#).

Note: Remember that if you are creating standardized values or match codes, they must be modeled as attributes in the metadata as well.

Standardized attributes and match codes are collectively referred to as flux attributes. A flux attribute is always associated with another attribute.

To make it easy to reuse data cleansing and data quality processing, the standardized and match code computations must occur in the mdm_stnd_<entity_type>.ddf job. This job is referenced by the parts of SAS MDM that need to use a consistent view of standardized attributes. For example, both data loading and search jobs can use standardized values and match codes.

To process data in different languages or from different countries at the same time, you must load all the required locales into the QKB both in your DataFlux Data Management Studio design environment and on the DataFlux Data Management Server. Then, you must modify the standardization jobs that clean data and generate match codes so that the appropriate algorithms are used.

To support additional locales:

- You must have a valid and supported QKB license for each locale that you want to use.
- You must update the embedded mdm_stnd_<entity_type>.ddf job with the processing for the additional locales.
- You must update your language definitions to include the new locale, which is used to verify the data when adding information to the hub.

Note: You need individual license codes in your DataFlux Data Management Studio and DataFlux Data Management Server license files to enable you to use more than one locale in your SAS MDM system. By default, you are licensed to use one locale. In most cases, the licensing is for US English, but this can vary according to your needs.

Inside the Clustering Process

Unique and Related Entities

Clustering helps identify unique entities. First, you must determine which fields can be used to identify related records. These are usually combinations of attributes, which can include standardized values or match codes that were computed in the cleansing step. For example, to determine unique entities in your database, you might set up the following rules.

Table 2.2 Clustering Rules

Fields	Result
If all of the fields within a rule match: FULL NAME (Match Code) ADDRESS LINE 1 (Match Code) CITY NAME (Match Code) POSTAL CODE (Match Code) or FULL NAME (Match Code) CUSTOMER_ID (Match Code)	A cluster that contains several records representing the same individual has been identified. You can use similar logic to find match clusters for organizations, households, and others.

You might also choose to use cross-field clustering rules in your matching logic. Rules of this kind look across columns for potential matching values, unlike standard clustering rules that only compare values in the column specified in the rule. Cross-field clustering rules can be used in combination with standard rules. For example, if you want to create a clustering rule that looks for similar names and similar addresses, but where the addresses could be in an Address1 field or an Address2 field, the rule would look like this:

Table 2.3 Clustering Rule Example

Fields	Result
Match or cross match: FULL NAME (Match Code)+ADDRESS LINE 1 (Match Code), FULL NAME (Match Code)+ADDRESS LINE 2 (Match Code)	This rule instructs the clustering process to look for similar address values in two columns instead of just one.

Note: Cross-field clustering can impact performance and should be used only when no other technique achieves the same result.

Clustering processing occurs in a data service named `mdm_cluster_<entity type>.ddf`, and is referenced from the `add_update_<entity type>.djf`, `mrm_add_<entity type>.ddf`, and `mrm_revert_move_<entity type>.ddf` jobs.

Successful identity management uses as many fields as necessary to limit the potential for false matches. For example, if you use only full name fields for match criteria, the result can be hundreds of instances of John Smith, most of which are not duplicates because they live at different addresses. The inclusion of address information in the matching criteria significantly improves the matching and linking results in this case. The entity type should specify as many cluster conditions as needed to define the identity criteria.

Note: Be aware of the following:

- We recommend building indexes for attributes that are used in the clustering process. For large hubs, the use of an index can speed up the processing significantly.
- Different database types can handle empty strings differently. For example, Oracle converts empty strings into NULLs, but SQL Server preserves the empty strings. This can cause anomalous results, especially when these values are used in cluster conditions. We recommend that you test your data

carefully, and ensure that null fields and empty strings are being handled as you expect in your jobs.

- The more match conditions that you add, the larger the impact on overall processing time. Use functionality available to you in DataFlux Data Management Studio to evaluate the suitability of your clustering strategy relative to the cost-benefit ratio of performance and final matching results. This caution applies especially to cross-field matching. The logic to find potential matches can become quite complex when more than a few cross-field clustering rules are used or when many alternate columns are used in cross-field clustering rules.

When new records are brought into the hub, any of the following conditions can occur:

Table 2.4 Condition Results

Condition	Result
The record is unique and does not match any records in the hub.	A new cluster is created; a survivor record is created for the cluster.
The record matches records in exactly one other cluster.	The record is added to the existing cluster, the cluster's previous survivor record is retired, and a new survivor record is constructed from all the active contributor records.
The record matches records in more than one cluster.	The hub looks at all the clusters that are matched to determine whether any of the matches present are records that have been manually moved. The presence of moved records indicates that the natural clustering process either overmatched or undermatched the data, and should not be used.

This processing can be changed by modifying the jobs that assign records to clusters (mrm_add_<entity_type>.ddf and add_update<entity_type>.djf). By default, SAS MDM proceeds as follows:

- If "natural" clusters do not contain moved records, the new record is clustered with these records. If more than one natural cluster matches, the clusters are collapsed into the cluster that has the lowest value of mdm_entity_cluster_id, usually the oldest one.
- If all the clusters contain moved records, the new record is placed in the cluster that has the lowest value of mdm_entity_cluster_id, usually the oldest one.
- In either case, existing survivor records are retired, and a new survivor record is generated.

In all cases, when an incoming record is already represented in the hub (based on its `mdm_src_sys_id` and `mdm_src_sys_rec_id` fields), the previous contributor is retired. This prevents data that has been replaced in the source system from being used in clustering and survivorship decisions.

If a bulk update includes two or more records with the same values for `mdm_src_sys_id` and `mdm_src_sys_rec_id`, the hub is unable to determine which record is the most recent. In this case, all the records are applied to the hub. If your source data enables you to prefer one record over another one, consider filtering out the other records before providing the data to the bulk jobs.

In rare cases, incoming data that matches both other incoming data as well as both natural and forced clusters in the hub can cause previously separated records to cluster together again. If this happens, moving the records again restores the previous configuration.

Improving Clustering Results

Entities are constructed from contributing records that are selected based on cluster conditions for the entity type. The cluster conditions encode matching rules, which are used to bring records with the same sets of values together. You can use any attribute within the cluster condition. However, if you choose the raw values from the incoming records, you might miss some matches. For example, if two records use different cases for a customer name but are otherwise identical, the matching rules places these records in different clusters.

By leveraging the [cleansing process](#), you can improve the match results by the following methods:

- standardizing values with the QKB, enabling case and common derivations, such as 'Robert,' 'Bob,' and 'Rob,' to be normalized
- using match codes to allow fuzziness to be built into the process

Using Match Codes

Overview of Match Codes

Match code generation is probably the most critical aspect of using SAS MDM. Match codes ensure that data that shares fundamental similarities come together into match clusters when match codes are compared. When used in groups with conditional logic, match codes provide a powerful way to match and link records that do not contain obvious similarities. Match codes are used where a degree of fuzziness is required to find all potential matches or relationships.

For example, to identify all the records in a data set where Robert Brauer exists, you might simply scan the data set for exact matches. However, that approach can miss Bob Brauer and Mr. Robert Brauer, who might be the same person. Using match codes, SAS MDM can identify these entities as potential duplicates, and process business rules to determine the nature of their relationship.

The following three name strings are considered a match:

Table 2.5 Match Codes

Input Data	Match Code
Bob Brauer	MYLY\$\$M@M\$\$\$\$\$
Mr. Robert Brauer	MYLY\$\$M@M\$\$\$\$\$
Robert Brauer	MYLY\$\$M@M\$\$\$\$\$

Typically, match processing is applied to several fields simultaneously. Records are considered to match if data strings from each field produce the same match codes. For example, a cluster condition that parses the **Name** and **Organization** fields from a contact table might return the following results:

Table 2.6 Multiple Field Match Codes

Input Data Name	Organization	Match Code (Name)	Match Code (Organization)
Bob Brauer	DataFlux	MYLY\$\$M@M\$\$\$\$\$ \$	8~GWX\$\$\$\$\$\$\$\$\$
Mr. Robert Brauer	DataFlux Corp	MYLY\$\$M@M\$\$\$\$\$ \$	8~GWX\$\$\$\$\$\$\$\$\$
Robert Brauer	General Motors	MYLY\$\$M@M\$\$\$\$\$ \$	B3Y4G~\$\$\$\$\$\$\$\$\$

The first and second records match, because both have the same match codes for Name and Organization. The third record does not match either of the other records, because the match code generated for the **Organization** field differs from those of the other records.

Note: The length of the match code data fields for most QKB locales is set at 40 characters. Some QKB locales (for example, French) require a longer match code data field. The match code data field lengths might need to be modified based on the locale used to process the data.

In DataFlux Data Management Studio, cleanse the data before matching for standardization or display purposes, because, in addition to generating the match codes from the data, you also want to store the data in a clean and valid format.

Match Definitions

The DataFlux Data Management Studio **Match Codes** node provides many match definitions for various data types and performs specific matching techniques and algorithms on each of those types. For example, the logic used to match name fields is very different from that used to match address fields. If a field is to be included in a match job for a given table, you must associate one of the available field types with that field. The accuracy of a match can be adversely affected by not choosing the field type that most closely resembles the data in the field. For example, the phrases William Street and Bill Street are

considered duplicates using the Name match definition, but they are not considered duplicates using the Address match definition.

Match Sensitivity

Match sensitivity defines how closely the DataFlux Data Management Studio **Match Codes** node compares data to determine whether duplicate records exist. Adjusting the sensitivity provides the flexibility needed to accurately identify a wide array of duplicate and near-duplicate records. You must apply one of the available match sensitivities to each field that is used to identify duplicate records. The sensitivity and match definition combine to make up the match job for the selected field.

Match sensitivity uses DataFlux proprietary data quality algorithms to create match codes that are clustered together to identify the duplicate and near-duplicate records contained in a database. You can apply a sensitivity level from 50 to 95. Adjusting the sensitivity level increases or decreases the accuracy and precision of the generated match codes. The default sensitivity level is 85.

If the Exact option is specified in the **Match Definition** field for a given record, a character-by-character match is performed on the specified field. You can select the number of characters on which to match. These numbers are available in the **Sensitivity** field. Exact sensitivity does not use any of the DataFlux data quality algorithms. It clusters data in the same manner as in an SQL ORDER BY statement.

Inside the Survivorship Process

After the clustering process is complete, SAS MDM will have identified a set of records that are logically part of the same entity. You can use SAS MDM to add survivorship rules that use these records to create a survivor record.

For example, when establishing the identity of unique entities, you might find two similar addresses listed for a particular individual. The business logic has identified these two records as matches, but SAS MDM must determine which address should be used as the standard.

Your rules must process the related clusters of information and pull together the correct pieces to create an accurate whole. As in the earlier example of how to determine unique entities in the database, a rule might state that when you have near-duplicate address information, you should use the address that comes from the Customer Relationship Management (CRM) system because the addresses are updated more regularly than in the other source systems. SAS MDM can translate this rule into usable business logic to correctly identify that individual. All of the logic used for identity management is completely customizable within SAS MDM.

Survivorship rules are implemented in a data service named `mdm_survive_<entity type>.ddf`, and are referenced from the `add_update_<entity type>.djf`, `mrm_add_<entity type>.ddf`, `mrm_move_<entity type>.ddf`, and `mrm_revert_move_<entity type>.ddf` jobs.

Inside the Harmonization Process

During the entity type job generation, a push job template is also generated. Data Stewards can modify the push job template to push data from the hub to data sources as specified by the Source System Harmonization setting for an entity type. When the push job is in place, changes made in the hub that result in a new survivor record can be harmonized with the configured source systems. SAS MDM provides a sample scenario to demonstrate source system harmonization for the INDIVIDUAL sample entity type. The source system harmonization sample requires you to set up the SAS MDM sample data as described in the *SAS MDM: Administrator's Guide*.

Assume that Call Center US needs to collect all the survivor records of individuals from all data sources. The Call Center US data source in the SAS MDM sample is represented by a table named CC_US_INDIVIDUAL in the SAS MDM database, and is created during the INDIVIDUAL entity type record harmonization process. In a real business scenario, this can be a table in a remote database, a text file, or any other form of data output. Every time an INDIVIDUAL survivor record is generated, the new survivor is sent to the Call Center US source system.

In the Source System Harmonization setting for the INDIVIDUAL entity type, you can see that Call Center US is specified as the selected source system to which to harmonize data when the hub sends an update. The sample push job for this is `mdm_push_individual.ddf`. When you load INDIVIDUAL entity type data, update existing Individual records, or create new INDIVIDUAL records from the hub, the survivors records are harmonized to the sample output table `CC_US_INDIVIDUAL`.

3

Deploying SAS MDM

<i>Adding Metadata</i>	25
Overview of Metadata	25
Entity Type Definitions	26
Attribute Definitions	27
Cluster Conditions	30
Relationship Types	31
Sample Metadata	31
<i>Generating Jobs</i>	32
<i>Loading the MDM Hub</i>	34
Overview of Loading the Hub	34
Managing Data Issues through Data Remediation	35
Managing Updates	39
Recovering from Load Problems	39
Trimming Historical Data	39
<i>Loading Data</i>	40
<i>Migrating from a Master Data Management Foundation Hub</i>	41
Overview of Migration	41
Migrating Metadata	41
Importing Source Systems and Languages	41
Migrating Instance Data	41

Adding Metadata

Overview of Metadata

To support different types of entities, SAS MDM uses metadata extensively to describe the information that it manages. Every entity and attribute in the system has associated metadata that tells SAS MDM what it is and how it should be handled. One of the first tasks in a SAS MDM deployment is determining the entities and their attributes that are to be managed by SAS MDM. Metadata can be entered into the system either from the interface or by using a batch job to load metadata definitions that are located in text files. In either case, metadata must be loaded before the introduction of instance data. After metadata has been added to the hub, data administrators can generate jobs through the Master Data Management user interface. We recommend that you start with the generated jobs and adjust them according to your needs.

Entity Type Definitions

Overview

Entity types are managed in the SAS MDM hub. Each entity type has attributes that must be added to the system to manage instances of that type. Entity type metadata is stored in the MDM_ENTITY_TYPES table. The required attributes are described in [“Entity Properties Tab” on page 101](#).

Inherited Metadata

An entity type can inherit metadata from another entity type by specifying the parent_id of the other entity type. When metadata is inherited, it is available for use within the entity type.

The ENTITY entity type defines the following required attributes. These attributes are inherited by every other entity type in the hub.

Table 3.1 Required Attributes

Attribute	Description
mdm_entity_type_id	Foreign key reference to the MDM_ENTITY_TYPES table, which identifies the type of the entity.
mdm_language_id	Foreign key reference to the MDM_LANGUAGES table, which identifies the locale to be used when processing the record with the QKB.
mdm_entity_id	The primary key for the MDM_ENTITY table. Each record in the hub has a unique value.
mdm_entity_cluster_id	The cluster that includes this record. Each record in the hub is part of exactly one cluster at a time.
mdm_src_sys_id	Foreign key reference to the MDM_SRC_SYS table, which identifies the source system for the record. SAS MDM defines two source systems: one for the Master Data Management user interface (for changes from the user interface), and another for the hub itself, which always consists solely of survivor records.
mdm_src_sys_rec_id	An opaque identifier for the originating source system. By using the source system ID along with the source system record ID, the hub can trace records back to the original source data.
mdm_valid_from_dttm	The earliest date and time at which the data in the record should be considered valid.

Attribute	Description
mdm_valid_to_dttm	The latest date and time at which the data in the record should be considered valid. Either leave mdm_valid_to_dttm blank (in database terms, set this value to null) or set it to a date where the data is no longer used. Clustering and nomination of survivor records must use information that has not expired. For example, clustering uses only records where the following condition exists: mdm_valid_from_dttm <= current time and (mdm_valid_to_dttm is null or current time < mdm_valid_to_dttm). Entities are retired by changing the mdm_valid_to_dttm attribute to the current time.

Abstract Entity Types

An entity type can be designated as abstract, which prevents instances of that entity type from being created. Abstract entity types can be useful for modeling purposes. For example, an attribute can be modeled once and inherited by other concrete entity types.

Attribute Definitions

Overview

Attributes define the data elements that collectively constitute the information represented by an entity type. As part of the data profiling phase, the necessary attributes for each entity type must be identified, along with standardization rules and any other necessary data quality rules. Attribute metadata is stored in the MDM_ATTRS table.

These attributes are then merged with a small set of attributes required for hub operation. These attributes are defined by the ENTITY entity type and are inherited by any derived entity types. By building on this base, you can extend SAS MDM to support any type of entity.

Attributes have a number of properties, such as a name (used for uniqueness constraints within the hub), presentation parameters, such as labels and display controls, and an associated data type, which identifies how the attribute is stored within the hub. These properties are described in [“Attribute Properties Tab” on page 103](#)

The information in the attribute definition is used to construct a column of the same name, with the data type, length (if any), and constraints as indicated by the metadata. In order for the entity type to be published successfully, the attribute name must be compatible with any restrictions imposed by the underlying database. Reserved words in the underlying database cannot be used as attribute names although they can be used as attribute labels.

Note: The total number of attributes that can be modeled on an entity type is subject to database limits. SAS MDM uses dynamic SQL to support flexible models, so the number of attributes that can be supported is a function of the length of the attribute names expressed as a comma-delimited list. For example, with Oracle the maximum size of this list is 4000 bytes, which allows for approximately 129 attributes if each attribute uses the full 30-byte name.

Attribute Data Types

Here are the supported data types for SAS MDM attributes:

Table 3.2 *Supported Data Types*

Data Type	Description
String	Presented in the Master Data Management user interface as a single-line text entry field, with a maximum length of 4000 bytes.
Text	Presented as a multi-line text entry field, with a maximum length of 4000 bytes.
List	Presented as a drop-down selection list. Values are provided by a semicolon-delimited string in the attribute's constraint field. The constraint field is limited to 200 bytes, including the semicolon delimiters. If the attribute is not marked as a required field, a blank choice indicates that there is no value for this attribute.
Date	Presented as a text field with an attached calendar widget. Clicking in the text field enables you to choose the date from the widget. Dates do not include a timestamp component.
Timestamp	Presented as a date, but with a time component. Note that changes to the default time must be entered into the text field directly.
Boolean	Presented as a drop-down selection list with the values Yes and No. If the attribute is not marked as a required field, a blank choice indicates that there is no value for this attribute.
Integer	Presented as a text field, which accepts up to 10 digits.
Real	Presented as a field limited to 31 digits in total length, with up to 5 digits behind the decimal point. In database terms, these are numbers with a precision of 31 and a scale of 5. Note: Some rounding can occur.
URL	Presented as a text field with a link widget to the right. Clicking the link widget opens a new browser window or tab for the URL specified in the text field. URL fields can be up to 4000 bytes in length. Note that no validation of the URL is performed.

Building Indices for Staging Tables

Custom indices can be created on user-defined attribute columns in staging and transpose tables either at the time of loading metadata or afterward. The list of indices is maintained in MDM_ENTITY_TYPE_INDICES table in the hub. At the time of loading metadata, the indices are created by add_sample_meta.djf job. The job add_indices.djf can be used to add custom indices anytime after the initial metadata creation. This job drops any existing custom indices for entity types in the indices input file and creates new ones as specified in the input file.

Presentation Metadata

You can control the order in which attributes appear in various elements of the Master Data Management user interface by specifying ordinal values in the attributes shown in the following table:

Table 3.3 *Attribute Order*

Attribute	UI Element
mdm_search_display_order	Entity search form.
mdm_table_display_order	Entity search results tables.
mdm_cluster_display_order	Cluster table on entity editor or Clustering tab.
mdm_form_display_order	Entity editor Properties tab.
mdm_preview_display_order	Preview panels throughout the interface.

The ordinal values are used to designate where in the Master Data Management user interface the element is placed, with lower-numbered items that appear vertically above those with higher numbers. The table views (for example, entity search results tables and entity editor cluster tables) display items from lowest to highest and from left to right.

All display order attributes follow the same rules. Enter either zero (0) or a negative value to hide the attribute in the Master Data Management user interface, or enter a value greater than 0 to show the attribute in the corresponding user interface element. Fields appear in the order specified by the display order values of their attributes.

An attribute does not appear in the search form or search results if access control has been applied to the attribute in either the current entity type or any of its derived types and the credentials of the user do not provide access to the attribute.

Including sufficient padding between the attribute sort order values is recommended. This simplifies the addition or reordering of attributes.

Attribute Modifiers

The following additional modifiers can be used to alter the processing of the data within your hub.

Table 3.4 *Additional Modifiers*

Modifier	Description
MDM_READONLY	This modifier causes Master Data Management to display values, but does not allow changes to the values. This can be useful to prevent users from attempting to modify the computed standardized values.

Modifier	Description
MDM_REQUIRED	When you model an attribute as a required attribute, SAS MDM uses that information to add a NOT NULL constraint in the database. Attempts to load a record that does not have a value for a required attribute is disallowed by the database. Note: If you retire a required attribute, the NOT NULL constraint is still present and can lead to difficulties when adding more data. Because the attribute is retired, it is not presented in the Master Data Management user interface; and because it is required, if it does not have a value, it cannot be added to the hub. SAS MDM cannot drop the NOT NULL constraint automatically, because the attribute might be involved in indices that need to be rebuilt. Contact your database administrator if you need to retire a required attribute, and ensure that the NOT NULL constraint is dropped, and any indices that reference the attribute have been rebuilt.
MDM_ENCRYPT_FLAG	SAS MDM provides the ability to encrypt sensitive data in the hub. The data is encrypted while it is stored in the database, but is transparent to the SAS MDM application. To enable encryption for an attribute, select the Encrypt attribute field in its attribute definition. After an attribute is published, this flag is set to read-only and cannot be modified. For this reason, it is important to decide what attributes to encrypt before an entity type is published and its jobs are generated. The following limitations apply: ■ Database limitations for encryption apply. ■ Encryption can increase the size of the data to be stored. When modeling encrypted attributes, set the field length approximately five times higher than the normally anticipated field length. ■ Only string fields can be encrypted. To encrypt integer values (for example, Social Security numbers or credit card numbers), they should be defined as string fields in the input data. ■ Entity Type label attributes cannot be encrypted. ■ Use of encrypted fields in cluster conditions and relationship type definitions is not supported. ■ Encrypting flux attributes (attributes holding standardization and match code values) is not supported.

Cluster Conditions

Cluster conditions are sets of attributes that are used to provide an identity to an entity. If records have the same identity, they become part of the same cluster, and are used when constructing the survivor record for the cluster. This enables data from disparate systems to be matched as a way of consolidating information in the hub.

Relationship Types

Relationship types illustrate a linkage between two entity types. For example, you might want to maintain a linkage between a manager and his or her employees. If the data provided to the hub includes information that enables the manager-employee relationship to be determined automatically (for example, by including the manager's employee ID in the Employee entity), then these relationships can be determined automatically by the hub and used to explore the resulting graphs of objects within the Master Data Management user interface.

Relationship attributes can be added to relationship types. Relationship attributes let you attach information to the relationships that link two entities together. This is similar to how entity attributes work, except that the attribute information describes the relationship instead of the entity itself. For example, a relationship type named *Employs*, which describes how companies and their employees are related to each other, might have two relationship attributes to describe in more detail the relationship between a company and one of its employees. A *Status* relationship attribute might describe the type of employment, such as full-time or part-time. A *Start Date* relationship attribute might describe the date on which the employee began to work for the related employer.

Once relationship attributes have been defined, values can be added from the record editor or from the relationship diagram. Values can also be added programmatically using data management jobs.

Sample Metadata

SAS MDM provides the following sample entity types, which demonstrate the concepts that are used to construct a hub. You can use these samples for installation verification, or as study guides to understand how the hub works.

Table 3.5 *Sample Entity Types*

Entity Type	Description	Notes
Part	Represents a product with dimensions.	Part includes a supplier name attribute that supports a one-to-many relationship with the Company entity type.

Entity Type	Description	Notes
Party	An abstract entity type used to define common attributes, cluster conditions, and relationship types for Individual and Company.	■ Party defines the name attribute shared by both Individual and Company, which enables search capabilities for entity types derived from it. You can search for Parties as a way of retrieving both Individuals and Companies. ■ Party provides a standardization job (mdm_stnd_party.ddf) that must be either referenced or reimplemented by derived entity types. ■ Tool_party_us_addver requires a license for the USPS address verification module of DataFlux Data Management Platform.
Individual	Represents a person as a type of Party. Individual builds on Party to introduce additional attributes for a shipping address, do-not-call processing, and demographic information.	■ Individual provides a standardization job (mdm_stnd_individual.ddf) that references the Party standardization job. ■ Individual extends the cluster conditions defined by Party. ■ The tool_individual_multi_update.ddf job requires credentials to function. Add the user ID and password to the real-time service node named Realtime Service - Call Add/Update Service.
Company	Represents an organization as a type of Party. Company builds on Party to introduce fax and website attributes, and adds a relationship type that can be used to derive relationships to Parts.	■ Company provides a standardization job (mdm_stnd_company.ddf) that references the Party standardization job. ■ Company extends the cluster conditions defined by Party.

Generating Jobs

Many of the jobs that you use in SAS MDM have a similar structure. The Master Data Management user interface makes it easy to generate jobs that embody these default behaviors. After creating the metadata for an entity type, you can use the **Generate Jobs** option from the entity type editor toolbar to create and publish jobs to the DataFlux Data Management Server. Selecting this option generates the jobs listed in the following table.

Note: Not all of these jobs are required in all cases. For example, abstract entity types generally support search operations, but do not support any of the other jobs. The unnecessary jobs are not shown in the **Generate Jobs** dialog box.

Table 3.6 Jobs

Name	Description
mdm_std_<entity_type>.ddf	Computes standardized and match codes for use in searches and creating new records.
mdm_extract_<entity_type>.ddf	Extracts data from a source system.
mdm_push_<entity_type>.ddf	Runs when changes to the cluster are made.
mdm_cluster_<entity_type>.ddf	Applies clustering rules
mdm_survive_<entity_type>.ddf	Applies survivorship rules.
add_update_<entity_type>.djf	Bulk-loads data into the hub.
mrn_add_<entity_type>.ddf	Adds new records to the hub from the Master Data Management user interface.
mdm_query_survivor_<entity_type>.ddf	Searches the hub for matching records based on specified criteria.
mdm_query_relationship_<entity_type>.ddf	Searches the hub for related records based on specified criteria.
mdm_query_cluster_<entity_type>.ddf	Searches for contributor records for a specified entity.
mrn_move_<entity_type>_records.ddf	Supports moving records between clusters.
mrn_revert_move_<entity_type>.ddf	Reclusters moved records so that they can cluster according to the conditions specified by the entity type.
mdm_archive_<entity_type>.djf	Removes retired records from the hub.
mdm_read_remediation_<entity_type>.ddf	Reads records used by SAS Data Remediation and SAS Task Manager Note: Must have access privileges to the record files.
mdm_read_workflow_<entity_type>.ddf	Reads transient records from the workflow file Note: Must have access privileges to the record files.
mrn_write_<entity_type>.ddf	Writes records used by SAS Data Remediation and SAS Task Manager. Note: Must have access privileges to the record files.

You can modify these templates to meet your needs. When modifying them, look for comments that provide information, as shown in the following examples.



SAS MDM provides sample jobs that demonstrate how to load records into the SAS MDM hub using batch interfaces. These jobs are named `add_update_<entity_type>.djf`, and can be generated using the Master Data Management user interface. The sample jobs generally read their data from a table in the SAS MDM database, perform the previously described [clustering](#) and [survivorship](#) steps, and then add the data to the hub.

- database credentials
- parameters controlling the use of bulk-load mechanisms such as the Oracle sqlldr or SQL Server bcp utilities
- parameters identifying the extract and standardization job to use during the load

- a parameter indicating the maximum number of records to read from the source

These parameters can be updated in the Echo node, or provided on the command line using `dmpexec`. For more information about running jobs from the command line, see the *DataFlux Data Management Studio User's Guide*.

The sample jobs use the metadata defined for the entity type to ensure that all the required fields (including the label attribute and the QKB locale) are present. Without these values, the data cannot be written to the database. Records that fail this check are sent to SAS Data Remediation, where they can be reviewed and corrected.

For optimized bulk loading, configure the jobs to use the appropriate database utility by specifying `IS_MAX_LOAD = Y` on the **Inputs** tab for the Echo node. Update the other inputs to specify your credentials, database server, database, and input file.

Note: Running more than one occurrence of an `add_update_<entity_type>.djf` job when `IS_MAX_LOAD = Y` is not supported. The DBMS-provided bulk-load utilities make changes directly to the underlying database storage that might conflict when running multiple occurrences of the job. If `IS_MAX_LOAD = N`, multiple occurrences of the job can be run simultaneously.

If you are using a SQL Server database, you need to generate the .bcp files used to load the staging tables and place them in `[SAS MDM Install Dir]\Solutions\SASMDM\samples\scripts`:

- `mdm_company_st.bcp`
- `mdm_individual_st.bcp`
- `mdm_part_st.bcp`

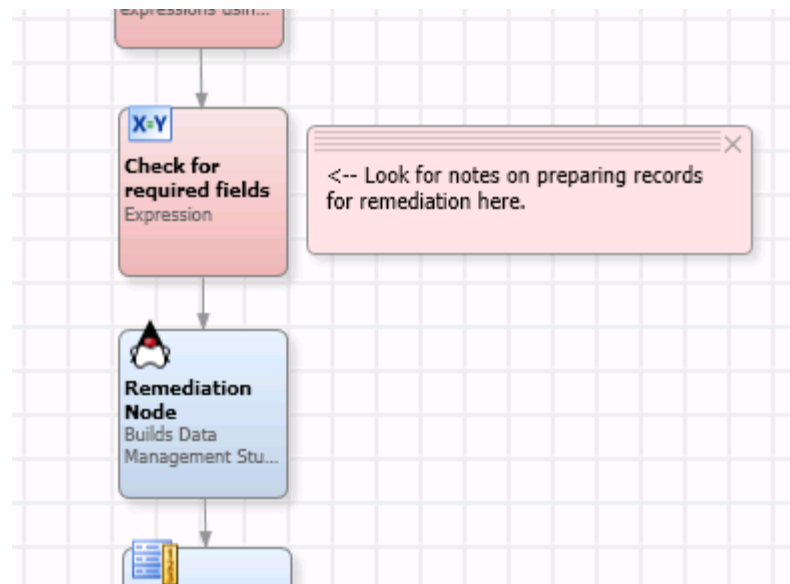
For more information about generating .bcp files and adjusting the generated files to match the input data files as prepared by the job, see the article “Non-XML Format Files (SQL Server)” on the Microsoft web site. The sample jobs generate their files by placing the field names in alphabetical order, which does not necessarily correspond to the table structure.

The `add_update` jobs can be run from the user interface, if your data steward has configured them this way. See [Chapter 8, “Using Source Systems,” on page 87](#) for details.

Managing Data Issues through Data Remediation

Jobs that are generated for each entity type have default locations where interaction with the data remediation environment is expected: the `mdm_extract_[entity type]` job, the `mdm_survive_[entity type]` job, and the `mr_add_[entity type]` job have a set of nodes that work together to flag data issues and send them to data remediation. These nodes are labeled Check for required fields and Remediation Node.

Figure 3.2 Remediation Node

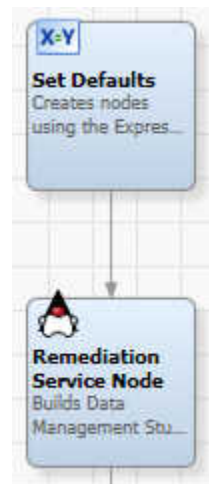


The expression code in the Check for required fields node flags records for data remediation based on user-defined business rules. Some rules are in place by default to capture data rows that are missing required fields, but you can add others as needed.

The flagged rows are passed to Remediation Node, which invokes a small Java application that stores them outside the hub.

The information is then normally submitted to the remediation service at the end of the job using a node labeled Remediation Service Node, which appears in the Gather Post-Live Stats node of the add_update_[entity type].djf batch job.

Figure 3.3 Remediation Service Node



Both nodes accept the MDM Staging ID as a key value to ensure that the correct set of records is being processed.

In some cases, the records are submitted when remediation node runs. This happens when no data processes through the node and the Submit when no data parameter is set to true.

The information that you supply in the Remediation Node determines how the data issues appear in data remediation. The following table describes the parameters.

Note: Some values are represented by macro variables, and you must not change them at this location.

Table 3.7 Remediation Node Parameters

Name	Value or Example	Notes
Staging ID Field	MDM_STAGING_ID	The staging ID used to coordinate actions across multiple remediation nodes and remediation service nodes.
Application	SAS MDM	Corresponds to application name defined in data remediation.
Name	mdm_extract_individual	Name of remediation package, which is a collection of records with issues.
Subject Area	INDIVIDUAL	Corresponds to subject area defined in data remediation.
Description	INDIVIDUAL Bulk Extract Staging Id % %MDM_STAGING_ID%%	Corresponds to package note.
Owner	%%DFINTL_DIS_USER%%	Pre-defined assignee for each issue.
Importance	medium	Pre-defined priority for each issue.
Issue Field	ISSUE_FIELD	Field where an issue has been identified.
Issue Description Field	ISSUE	Corresponds to issue type in data remediation.
Label Attribute Field	NAME	Label of field for issue name, also used to supply the value for issue name.
User-Defined Field 1	ADDRESS1	Data value from this field shows in the first user-defined field in data remediation. This label is also tied to the package.
User-Defined Field 2	CITY	Data value from this field shows in the second user-defined field in data remediation.
User-Defined Field 3	STATE_PROVINCE	Data value from this field shows in the third user-defined field in data remediation.
Workflow Name Field	WORKFLOW_NAME	Passes a workflow name to data remediation to initiate a workflow. The name must be a valid and active workflow available in the SAS Workflow environment.

Name	Value or Example	Notes
Submit when no data	true	Data value from this field shows in the third user-defined field in data remediation. For complex jobs, more than one remediation node can be used. The provided templates use one type of node to gather information, and a second type to submit remediation data to the SAS Data Remediation service. However, there are some circumstances where no data makes it through the first node, and the job stops before making it to the second node. In this situation, setting Submit when no data to true ensures that the gathered data is submitted to the data remediation service.

The following table describes the Remediation Service Node parameters.

Table 3.8 Remediation Service Node Parameters

Name	Value or Example	Notes
Staging ID Field	MDM_STAGING_ID	The staging ID used to coordinate actions across multiple remediation nodes and remediation service nodes.
Application	SAS MDM	Corresponds to application name defined in data remediation.
Name	add_update_individual	Name of remediation package.
Subject Area	INDIVIDUAL	Corresponds to subject area defined in data remediation.
Description	INDIVIDUAL Bulk Extract Staging Id % %MDM_STAGING_ID%%	Corresponds to package note.
Owner	%%DFINTL_DIS_USER%%	Pre-defined assignee for each issue.
Importance	medium	Pre-defined priority for each issue.
User-Defined Field 1	ADDRESS1	Data value from this field shows in the first user-defined field in data remediation. This label is also tied to the package.
User-Defined Field 2	CITY	Data value from this field shows in the second user-defined field in data remediation.
User-Defined Field 3	STATE_PROVINCE	Data value from this field shows in the third user-defined field in data remediation.

Managing Updates

It is possible to reprocess every row that might have changed in a weekly update using the same method, but this is not the most efficient way to handle changes. If each source system records changes to its own records and identifies new or deleted records, then only those changes are selected and processed as updates against the hub. The change files could be created using any Extraction, Transformation, and Loading (ETL) process, assuming it can access the files themselves.

The `mdm_extract_<entity_type>` job defines an input variable named `EXTRACT_PARAMETER` that is used for this purpose. The extract job can use this parameter to resume after the last load, assuming that a value exists that indicates where the last load left off. For example, if dates are associated with changes, you can use the latest date found in one run as the starting point for the next run. If records are being added and use a monotonically ascending numeric key, the extract job can use the largest value of the key from one run as the starting point for the next.

Other strategies are possible. SAS MDM does not interpret the key, but merely keeps track of it for use by the extract job.

Recovering from Load Problems

The sample jobs write incoming data into a staging table (one per entity type) before updating the hub. Only after the data has been successfully staged is it applied to the hub. If a problem occurs during the load operation, the transaction is rolled back. The records remain in the staging table, and the hub is unaffected.

The `MDM_PROCESS_CONTROL` table contains information about all the load jobs and their status, and can be used to determine the staging ID. Failing loads are identified by the column in the `MDM_PROCESS_CONTROL` table, which has a unique value for each load process. The values can be used to recover from the failure by either deleting those records from the staging table or restarting the go-live process.

Trimming Historical Data

The hub continues to expand with each insert and update because Retired records are maintained in the hub. No data is deleted during normal SAS MDM operation, although if the information is no longer needed, it can be moved to offline storage to free up resources.

The exact requirements for removing data from the hub can vary greatly. SAS MDM provides a sample job that is suitable for configuration. Within the job are notes about how to modify the job for your particular entity type. The `mdm_archive_<entity_type>.djf` job, in its original state, is used to generate archive files for the entity type. It cannot be accessed through the Master Data Management; you must run it manually. The `mdm_archive_<entity_type>.djf` job is generated using the Master Data Management interface. This job does the following:

- generates the archive files for expired audit records, expired error records, affected hierarchy records, expired cluster members, and expired transpose records.

- deletes any expired audit records, expired error records, affected hierarchy records, expired cluster members, and expired transpose records.
- cleans empty clusters.
- cleans up entities from the database.

Note: Relationship attributes are not archived by this job.

Loading Data

SAS MDM provides sample metadata and instance data to load into your hub. By default, the jobs to load the instance metadata are in `[SAS MDM Install Dir]\batch_jobs\sasmdm`. Installing the sample metadata and instance data is optional.

Table 3.9 *Metadata Jobs*

Job	Description
add_sample_meta.djf	Loads the required metadata for the samples.
add_ui_language_sample_<locale>.djf	Loads translatable metadata for the samples.

The jobs to load the instance data are generated on the DataFlux Data Management Server.

Import the jobs from the DataFlux Data Management Server into the following folders on your DataFlux Data Management Studio repository, mirroring their locations on the DataFlux Data Management Server:

- `batch_jobs\sasmdm`
- `data_services\sasmdm`

Execute the following jobs to load the sample data. You should examine and, if necessary, modify the Echo node at the top of the job to identify the data set that you want to load.

Table 3.10 *Instance Data*

Job	Description
add_update_individual.djf	Loads sample INDIVIDUAL data.
add_update_company.djf	Loads sample COMPANY data.
add_update_part.djf	Loads sample PART data.

Migrating from a Master Data Management Foundation Hub

Overview of Migration

If you have used DataFlux Data Management Studio to create a hub, you can migrate the metadata and instance data from that hub into SAS MDM. See the *Using Master Data Management Foundation* section of the *DataFlux Data Management Studio User's Guide* for more information about the Master Data Management Foundation feature.

Migrating Metadata

The metadata maintained by Master Data Management Foundation is a subset of the metadata maintained by SAS MDM. After migrating the metadata, you should review the entity type definition and augment the metadata with presentation metadata—for example, configuring which attributes appear in the Master Data Management search and editor interfaces. To migrate data from Master Data Management Foundation, you must configure a data source for the Master Data Management Foundation hub, and you must know the schema name for the hub.

Importing Source Systems and Languages

The source systems that provide data to the Master Data Management Foundation hub, and the language for the hub, might not be known to SAS MDM. You can import the source systems and languages defined in the Master Data Management Foundation hub by using the `migrate_mdf_meta.ddf` job, available in the `sample\templates\meta` directory. You must configure the Echo node at the top of the job, and ensure that the `MDF_DSN` and `MDF_SCHEMA` inputs correspond to your hub. When you run the job, it imports any source systems and languages that are not already known to the hub.

Migrating Instance Data

Data can be extracted from a Master Data Management Foundation hub and loaded into a SAS MDM hub. Generate the appropriate jobs and configure them according to your requirements.

4

Controlling Access

About Access Control	43
Overview	43
SAS MDM Capabilities	44
SAS MDM Roles	45
Defining Groups	46
Interaction with DataFlux Data Management Server Permissions	47
Managing Access	47
Designing Jobs	47

About Access Control

Overview

A centralized repository for use across an enterprise requires security. Organizations need the ability to secure access to data, to restrict access to people and systems with a legitimate business use for the data, and to audit changes to the data. Access control restricts access based on business requirements. By configuring access control, you can control which groups have access to different types of information. You can specify controls for the following:

- capabilities that are available in the user interface. For example, some users might be prevented from modifying records.
- complete restriction of access to information about an entity type. For example, a subset of users might have access to COMPANY data, but be unable to access PART data.
- restriction to attributes within an entity type. For example, a subset of users might have access to COMPANY data, but be unable to access contact information modeled within the COMPANY entity type. Access is either completely allowed or denied. The sole read-only support for attributes is the MDM_READONLY modifier or the specification of capabilities within the SAS Metadata Server that allows access to, but not modification of, data within the hub.

Every hub has a minimal level of access control. Users must be authenticated before using SAS MDM and have a minimal capability that grants access to the SAS MDM application. Users are registered through the SAS Management Console, where they are assigned roles and group membership.

If you do not want to use the SAS MDM access control capability, no action is required. By default, all access is granted. However, when an explicit grant is made to one of the controlled types of information, grants for that information become mandatory.

SAS MDM Capabilities

SAS MDM defines one capability, View Application, that provides access to the application. Every SAS MDM user must have the View Application capability.

SAS MDM defines five capabilities that govern functions related to data in the hub:

Table 4.1 *Capabilities Related to Hub Data*

Capability	Function
Manage Entities	Enables users to create, edit, and retire entities.
Manage Clusters	Enables users to address overmatch and undermatch issues in the clusters constructed by the hub.
Manage Relationships	Enables users to create and retire relationships between clusters.
View Source Systems	Provides UI access to load histories.
Run Source System Tools	Enables users to run source systems tools.

SAS MDM defines four capabilities that govern functions relating to the metadata model and administrative aspects of the hub:

Table 4.2 *Capabilities Related to Metadata and Administration*

Capability	Function
Manage Entity Types	Enables users to create, modify, and retire entity type definitions within the hub.
Manage Relationship Types	Enables users to create, modify, and retire relationship types within the hub.
Manage Source Systems	Enables users to create, modify, and retire source system definitions that are contributors to the hub.
Manage Languages	Enables users to create, modify, and retire language definitions used for data cleansing within the hub.

In addition, you can require that a user be a member of a specific group or groups to access records for a particular entity type, or for individual attributes of

those entity types. By default, all access is granted. However, when an explicit grant is made to one of the controlled types of information, grants for that information become mandatory, as shown in the following list:

- If no access rule is defined, access is granted.
- If an access rule is defined and the user is a member of at least one of the groups listed, access is granted.
- If an access rule is defined and the user is not a member of any of the groups listed, access is denied.

Note: For entity types, the label attribute must be accessible. For example, if the entity type has been explicitly granted, but the label attribute has been restricted, the entity type itself is restricted.

SAS MDM Roles

SAS MDM creates five new roles in SAS Management Console. These roles are tied to distinct capabilities in the data remediation application. Depending on assigned role of the user, the experience in SAS MDM changes. Certain features are available to those assigned one role, but the same features are not available to users in another role. Users can also create new groups and roles with distinct sets of capabilities.

Here are the roles for SAS MDM:

Table 4.3 SAS MDM Roles

Role	Capabilities
MDM: Administration (Default Group: Data Management Administrators)	View: Application; Source Systems Manage: Entities; Clusters; Relationships; Entity Types; Relationship Types; Source Systems; Languages Run: Source System Tools
MDM: Business Authorization (Default Group: Data Management Business Approvers)	View: Application; Source Systems Manage: Entities; Clusters; Relationships
MDM: Business Usage (Default Group: Data Management Power Users)	View: Application; Source Systems Manage: Entities; Relationships Run: Source System Tools
MDM: Business View (Default Group: Data Management Business Users)	View: Application; Source Systems
MDM: Stewardship (Default Group: Data Management Stewards)	View: Application; Source Systems Manage: Entities; Clusters; Relationships; Entity Types; Relationship Types Run: Source System Tools

Defining Groups

Groups are used to represent sets of users or a set of privileges within the SAS Metadata Server. Users can belong to multiple groups, and the set of roles and capabilities for a user is a combination of roles and capabilities from all the groups to which the user belongs.

When creating additional groups for access control, you should consider the needs of the users of the information stored within SAS MDM. The groups should be defined at the level appropriate to provide required access to the information, but no access beyond that. For example, consider these cases:

Table 4.4 *Employee Situations*

Employee	Department	Requires
Alice	Payroll	both employee address information and employee financial information.
Bob	Human Resources	employee address information, but does not require employee financial information.
Charlie	Manufacturing	no employee information.

This situation requires two groups:

Table 4.5 *Required Groups*

Employee	Group	Permissions
Alice	PAYMASTER	Grants access to the employee entity type and the financial attributes.
Bob	HR	Grants access to the employee entity type, but does not have any particular restrictions on attributes. Bob cannot access any financial attributes because he is not a member of the PAYMASTER group.
Charlie	Not assigned to a group	Charlie is not assigned to a group because he does not need to access employee data.

If another employee, David, requires access to financial attributes, a new group that has access to those attributes can be created. Alice and David would both be assigned that role, or a new role could be created just for David.

Note that granting groups access to entity types and attributes is complementary to assigning capabilities to a user. A user must still be assigned the View Application capability to be able to access the user interface. Assigning a user to a group with authorization for access to an entity type does not replace this capability.

You must decide whether it is more appropriate to assign roles to a user or to assign roles to a group, after which you assign users to the group.

Interaction with DataFlux Data Management Server Permissions

DataFlux Data Management Server enables you to configure permissions that either allow or deny access to jobs for users and groups. In some cases, entity-type-specific jobs can be controlled effectively by using permissions. In other cases, only one job provides access to all the entity types. Because of this difference, not all security configurations can be handled through permissions alone. In addition, permissions are insufficient to provide control over individual attributes because they operate at the job level. For these reasons, permissions should be considered a necessary but insufficient component of data security, which is augmented by the SAS MDM access control capability.

For more information about DataFlux Data Management Server permissions, see the *DataFlux Data Management Server Administrator's Guide*.

Managing Access

Access can be granted through batch interfaces. The `add_sample_meta.djf` job is an example of loading authorization information from text files. Access can also be granted interactively using the association tables available from the **Data Model** tab. Batch interfaces are more appropriate if you have to set up a large number of authorizations.

Note: Authorization changes made after a user logs on are not recognized until the user logs off and logs on again.

Designing Jobs

Access control operates in the Business tier of SAS MDM. This requires that extensions or modifications to the jobs provided by SAS MDM take access control into account, as follows:

- Jobs that can authorize access must use a variable named `MDM_USER_ROLES`, which is a comma-delimited string of roles assigned to the current user.

- Jobs that query the database directly must join the appropriate table to the roles table and ensure that the query factors in retirement status for authorization records. They must conform to the grant decisions described previously. For example, to determine the entity types that are accessible to a user with the MDM_BIZUSER role, the following query might be used:

```
select et.mdm_entity_type_id,et.mdm_entity_type from mdm_entity_types et where
mdm_entity_type_id in
(
  select etr.mdm_entity_type_id from mdm_entity_type_access etr
    where sysdate between etr.mdm_valid_from_dttm and etr.mdm_valid_to_dttm
        and lower (etr.mdm_group) in
union
select et2.mdm_entity_type_id from mdm_entity_types et2
  where et2.mdm_entity_type_id not in
    (select mdm_entity_type_id from mdm_entity_type_access
      where sysdate between mdm_valid_from_dttm and mdm_valid_to_dttm)
```

The two clauses in the sub-select check the following conditions:

- If an active grant exists for the user based on the roles, the entity type is included.
- If no active grants exist, the entity type is included.

Note: Group names are tested using the `in` predicate because any user can belong to more than one group. Compare group names in a case-insensitive manner.



Part 2

SAS MDM Interface

Chapter 5

<i>Accessing SAS MDM</i>	51
--------------------------------	-----------

5

Accessing SAS MDM

<i>Log On to Data SAS Management Console</i>	51
<i>SAS Data Management Console Home Page</i>	52
<i>Specifying Your Preferences</i>	54
Specifying Global Preferences	54
Specifying SAS Data Management Console Preferences	55
Specifying SAS MDM Preferences	55
<i>Overview of the SAS MDM Interface</i>	56
Menu Options	56
Tables	57
Fields	57

Log On to Data SAS Management Console

SAS MDM is accessed through the standard logon window for SAS applications. Logging on from this window opens SAS Data Management Console from which you can launch SAS MDM.

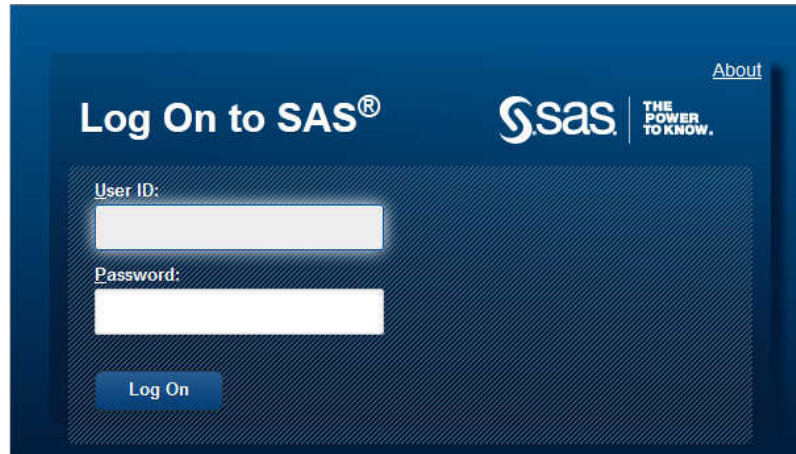
The port for all SAS Data Management web components is configured during installation. The default port number is 80. If your site is using the default port for these web components, you can access SAS Data Management Console from the following URL: `http://hostname/SASDataManagement`.

If your site is not using the default port for these web components, you must specify the port in the URL. For example, if the port is 7980, you can access SAS Data Management Console from the following URL: `http://hostname:7980/SASDataManagement`.

To log on to SAS Data Management Console:

- 1 Click the URL that is supplied by your system administrator, or paste it into the address field of your browser to display the SAS logon window:

Figure 5.1 Logon Window for SAS Data Management Console



- 2 In the **User ID** field, enter your user ID.
- 3 In the **Password** field, enter the password for your user ID.

Note: Your password is case-sensitive. Your user ID might be case-sensitive, depending on the operating system that is used to host the web application server. If you need assistance, contact your system administrator.

- 4 Click **Log On** to display SAS Data Management Console.

Note: If you log on to SAS MDM in one browser tab, then log on to SAS MDM or a different SAS web application in another browser tab, the same credentials are used automatically for subsequent authentication attempts.

To log off from SAS Data Management Console, click **Sign Out** in the upper right corner of the user interface:

Figure 5.2 SAS Data Management Console



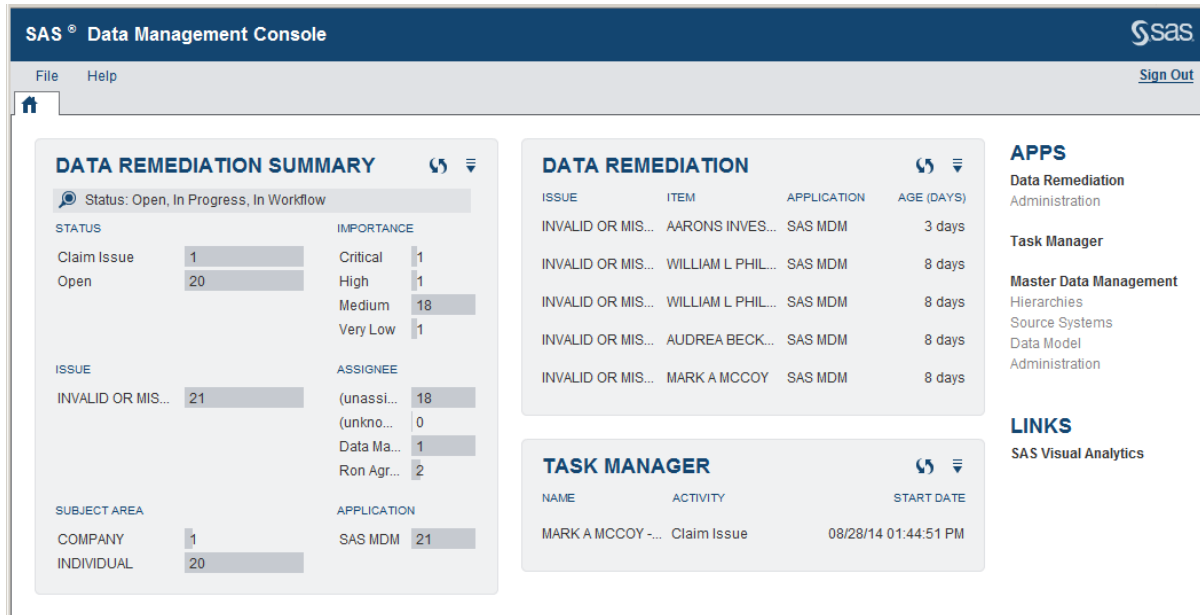
Note: When you select **Sign Out**, you are logged off from all tabs opened by the console.

SAS Data Management Console Home Page

The SAS Data Management Console home page enables you to launch the various components of SAS MDM.

Note: The following image is an example. Your specific content might be different.

Figure 5.3 SAS Data Management Console



- The **File** menu on the menu bar enables you to set preferences.
- The **Help** menu enables you to access user's guides and SAS on the web.
- The **DATA REMEDIATION** portlet lists, and enables you to open, current remediation tasks.
- The **DATA REMEDIATION SUMMARY** portlet provides summary statistics on issues tracked in the SAS Data Remediation databases.
- The **TASK MANAGER** portlet lists, and enables you to open, current task manager items.
- The **APPS** listings link you to the component applications of SAS MDM.

Note: The list of applications available to you varies according to your assigned role and to which SAS products you have installed.

- The **LINKS** listings provides access to other applications configured by your administrator.

Note: These are not necessarily SAS Applications. If SAS Visual Analytics is one of the links, see [Chapter 11, "Working with SAS Visual Analytics," on page 139](#) for more information about SAS Visual Analytics reports.

Here are the SAS MDM applications:

- **Data Remediation**
 - ☐ Administration
- **Task Manager**
- **Master Data Management**
 - ☐ Hierarchies
 - ☐ Source Systems
 - ☐ Data Model
 - ☐ Administration

Each of these applications is explained in detail in this document.

Specifying Your Preferences

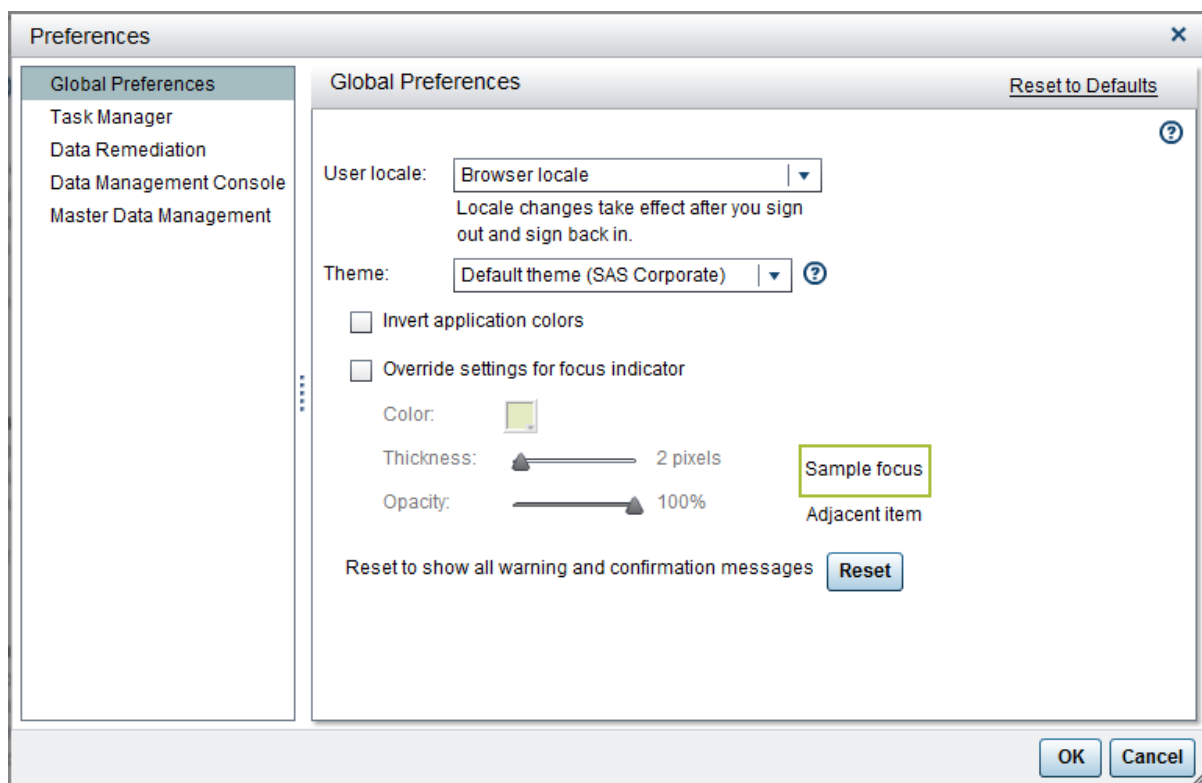
Specifying Global Preferences

You can specify global preferences to apply to all SAS web applications that are displayed with the Adobe Flash player. These preferences are set by each user.

To specify global preferences:

- 1 Select **File** ► **Preferences** to open the Preferences window.

Figure 5.4 Global Preferences



- 2 Click **Global Preferences** in the left pane.
- 3 Select a **User locale** to specify your language and geographic region.
Select a **Theme** to change the color scheme and other visual settings for all of your SAS web applications.
Select **Invert application colors** to invert all of the colors in your SAS web applications.
Select **Override settings for focus indicator** to change the color, thickness, and opacity of the focus in your SAS web applications.
- 4 Click **OK** to apply your changes.
- 5 Click **Reset to Defaults** to restore default settings.

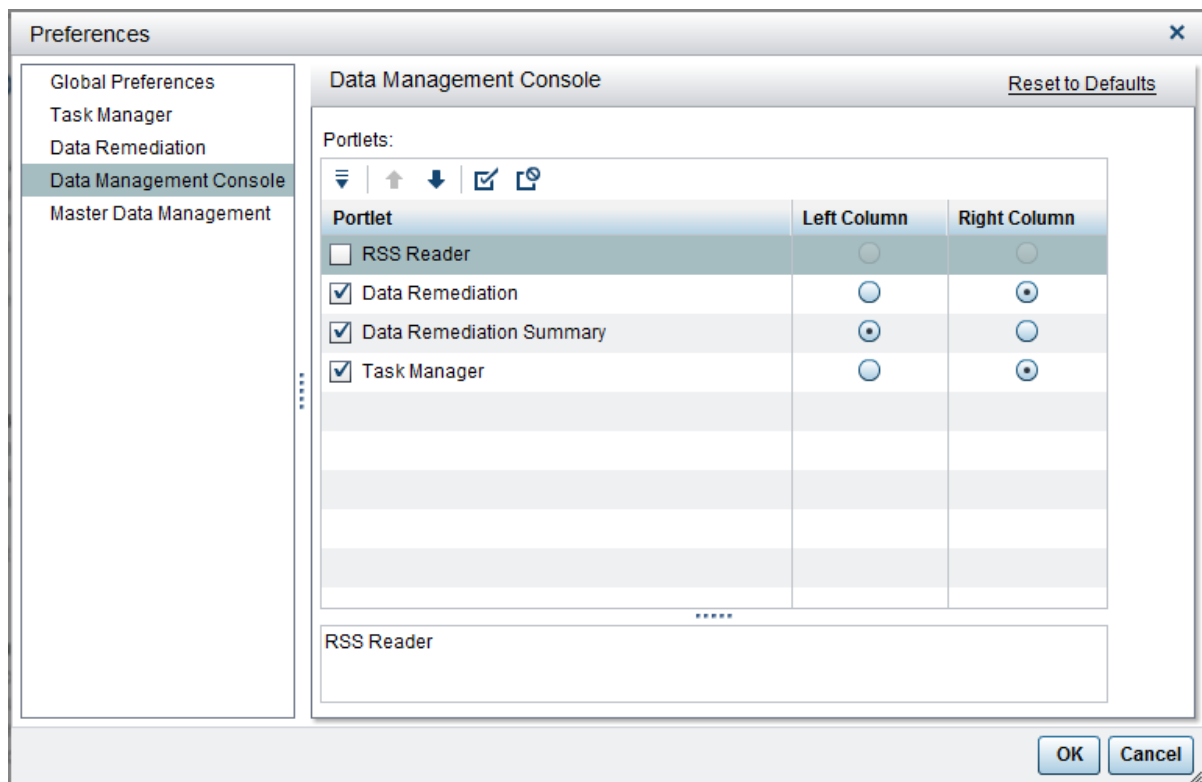
Note: If you changed the **User locale**, then you must log off and log back on to SAS MDM for the change to take effect.

Specifying SAS Data Management Console Preferences

To specify SAS Data Management Console preferences:

- 1 Select **File ► Preferences** to open the **Preferences** window.
- 2 Click **Data Management Console** in the left pane.

Figure 5.5 SAS Data Management Console Preferences



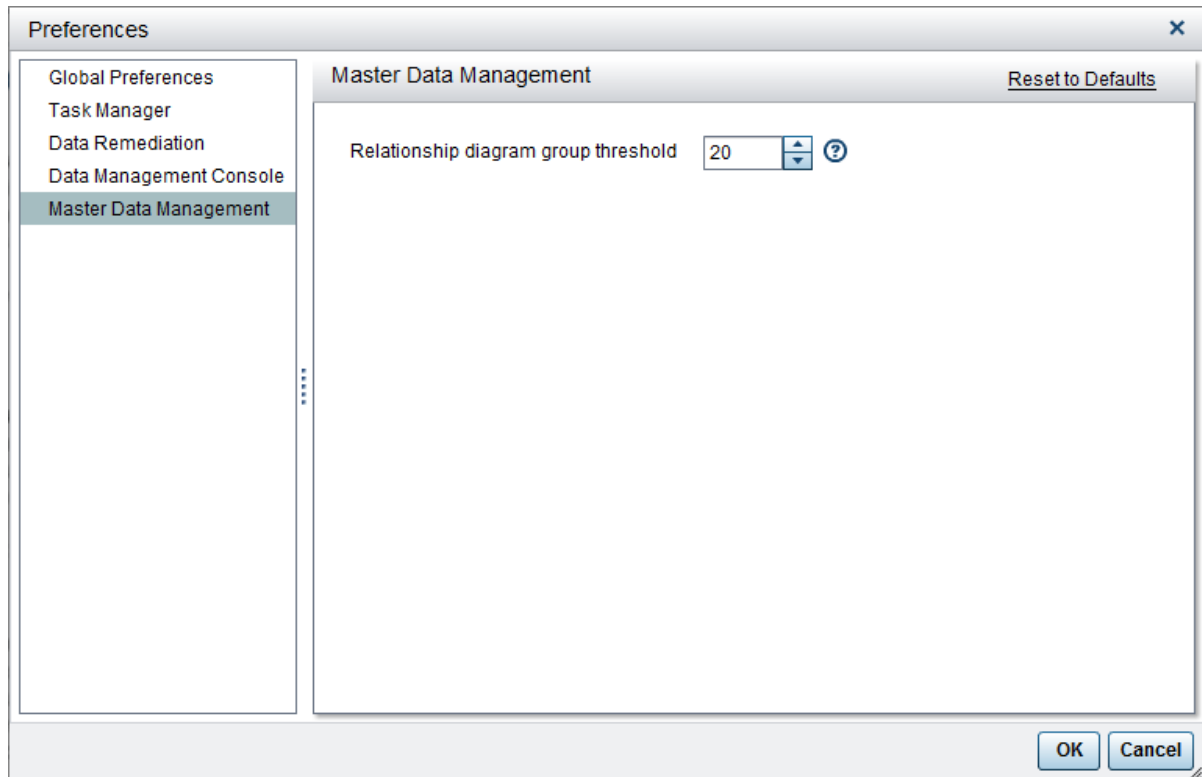
- 3 Specify the location of portlet panes that appear on SAS Data Management Console.
- 4 Click **OK** to apply your changes.
- 5 Click **Reset to Defaults** to restore default settings.

Specifying SAS MDM Preferences

To specify SAS MDM preferences:

- 1 Select **File ► Preferences** to open the Preferences window.
- 2 Click **Data Management Console** in the left pane.

Figure 5.6 SAS MDM Preferences



- 3 Specify the threshold for relationship diagram groups. Entities with a number of relationships greater than this threshold are presented as a group node on the relationship diagram. For more information, see [“Relationship Diagram Tab” on page 74](#).
- 4 Click **OK** to apply your changes.
- 5 Click **Reset to Defaults** to restore default settings.

Overview of the SAS MDM Interface

Menu Options

The SAS MDM interface has several common features. The toolbars of the interface contain an **Actions** menu . Clicking  displays the **Actions** menu and enables you to select specific options. Most options available from the **Actions** menu are also available through icons on the same toolbar. Holding the cursor over the icon displays the function of the icon. Clicking the icon performs the function.

You can click icons to expand and collapse elements of the interface. Click  or  to expand an item. Click  or  to collapse an item.

Tables

SAS MDM presents lists of objects in a tabular form. These tables show a subset of the information associated with the objects. All entries in a table are the same type of object. To open an object in an editor, double-click or right-click the row and select **Open** or **Edit** from the pop-up menu.

Figure 5.7 Sample Table

Entity Types

New Entity Type

	Type ▲	Display Name	Description
	COMPANY	Company	Company
	ENTITY	ENTITY	Abstract Entity Type
	INDIVIDUAL	Individual	Individual
	PART	Part	Part
	PARTY	Party	Party

To open an object, you can do any of the following:

- Click  from the toolbar.
- Click  and select **Open** or **Edit**.
- Double-click a table row.
- Right-click a table row and select **Open** or **Edit** from the pop-up menu.

To sort a table column in either ascending or descending order, click the column heading. Sorted columns display up or down arrows to indicate the sort direction.

Fields

Required fields on the interface are marked with an asterisk (*). Read-only fields appear dimmed.

Figure 5.8 Fields

▼ Entity Type Properties

Type: * ENTITY

Description: Abstract Entity Type

Parent Type: ▼

Valid From: 05/05/13 11:07:25 AM

Last Modified: 05/05/13 11:07:58 AM

Last Published: 05/05/13 11:07:25 AM

Valid To:

☒ Entity is an abstract entity (cannot be instantiated)

If you enter invalid information in a field, a red border appears around the field. You can hold the mouse over the red border to view information about the invalid field entry.

For date fields, you can either enter the date directly or click  to display a calendar. From the calendar, click a valid date to load it into the field. Click the navigation buttons to change the month and year. Click anywhere outside the calendar to hide it.

Part 3

SAS MDM Components

Chapter 6	
<i>Using Master Data Management</i>	61
Chapter 7	
<i>Using Hierarchies</i>	83
Chapter 8	
<i>Using Source Systems</i>	87
Chapter 9	
<i>Using Master Data Management Data Model</i>	93
Chapter 10	
<i>Using Master Data Management Administration</i>	127
Chapter 11	
<i>Working with SAS Visual Analytics</i>	139

6

Using Master Data Management

About Master Data Management	61
Overview	61
Managing Entities from the Master Data Management Tab	62
Searching Entities	62
Overview	62
Quick Search	63
Field Search	63
Deep Linking to Master Data	65
Working with Search Results	65
Overview	65
Using Tools	67
Editing Entities	68
Records Tab	69
Overview	69
The Entity Editor	70
Recluster Editor	72
Hierarchies Tab	74
Relationship Diagram Tab	74
Overview	74
Working with Relationships	77
History Tab	78
Creating New Entities	78
Creating a Remediation Issue	80

About Master Data Management

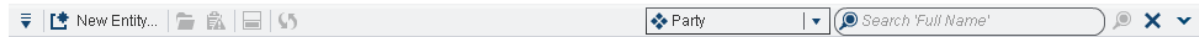
Overview

From the **Master Data Management** tab, business users and data administrators can search for, edit, create, and retire master data entities.

Note: The **Master Data Management** tab is available only to users who have View Application capability.

To access the **Master Data Management** tab, click **Master Data Management** on the SAS Data Management Console page. By default, the tab opens to the search toolbar:

Figure 6.1 Master Data Management Toolbar



Note: In the following descriptions, **Actions** menu items are accessed by clicking  from the toolbar. Most functions available from the **Actions** menu are also available from icons on the same toolbar. Clicking the icon performs the function.

Managing Entities from the Master Data Management Tab

You can perform the following actions from the **Master Data Management** tab:

- Search master data entities using either a quick search or a field search.
- Edit and retire master data entities returned from a search.
- Create a new master data entity for the hub.
- View, create, and retire relationships
- Create new remediation issues to be resolved in SAS Data Remediation.

The following sections describe these actions in detail.

Searching Entities

Overview

In order for a master data entity to be available for a search, the following must be true:

- The entity must be active (non-retired) and have been published for use.
- The sort of the entity must be greater than zero.
- The Entity type must define a label attribute. If the entity type is abstract, it is optional whether it specifies a label attribute. If an abstract entity type does not define a label attribute, it is not available for search operations.
- If the entity has been modeled as an abstract entity type, it must contain at least one concrete, derived entity type that meets these criteria.

Note: You can perform either a quick search or a field search.

Quick Search

To perform an entity quick search:

- 1 Select an entity type from the drop-down list on the right side of the search toolbar:

Figure 6.2 Quick Search Toolbar



The default attribute in the search field corresponds to the label attribute for that entity type. For example, when Party is the selected entity type, Full Name is the default attribute for the data search.

Note: You can also change the search method from the default of DataFlux Match to another type of match. See the description of [search methods on page 63](#).

- 2 Enter the search term into the search field in the toolbar
- 3 Do one of the following:
 - Click  to begin the search.
 - Click  to clear the search data.

Best records are returned in a table. See [“Working with Search Results” on page 65](#) for information about exploring data returned from the search.

Field Search

To perform an entity field search:

- 1 Click  on the right side of the toolbar to expand the search pane.

The search pane is constructed from the entity type's metadata. Only attributes that have been configured to appear by an administrator are shown in the pane.

Figure 6.3 Search Pane

Entity Type:

Party

▼ Contact

Full Name:

Primary Phone Number:

Primary Email Address:

► MDM Identifiers

▼ Address

Address Line 1:

City:

State/Province:

Postal Code:

Country:

- 2 Enter the appropriate search criteria in the fields of the search pane.
- 3 Click  to display the menu.
- 4 Click **Search Method** to display the search method options.

By default, the search is based on a DataFlux match. The search methods are:

DataFlux Match

uses match codes during searches and enables fuzzy matching capabilities. If match codes are not available for the fields, SAS MDM attempts to use standardized fields. If match codes or standardized fields are not available, SAS MDM searches for an exact match.

Similar Match

uses database-specific, approximate match functions. For Oracle and SQL Server, the soundex() function is used in combination with an exact match. This is because soundex() does not support some values, such as numbers and dates.

Case Insensitive

disregards the case of the search term.

Exact Match

searches for the input string exactly as it was entered.

5 Do one of the following:

- Click  to begin the search.
- Click  to clear the search data.

Search results are returned in a table. See [“Working with Search Results” on page 65](#) for information about exploring data returned from the search.

Deep Linking to Master Data

Deep linking refers to constructing a URL hyperlink that, once initiated, can take a user directly to master data in SAS MDM rather than requiring the user to search for it. For example, you can construct a URL link in an e-mail and send to a data steward. When the data steward clicks the URL link, it opens the SAS MDM application and loads the entity data encoded in the URL.

From within the appropriate environment, create a URL using the following syntax:

```
http://<your_host>/SASDataManagement/  
#type=<entity_type>&entity=<entity_ID>&module=MDM
```

For example, you might create the following:

```
http://localhost:8080/SASDataManagement/  
#type=COMPANY&entity=21682&module=MDM
```

Note: Creating a remediation issue submits a deep link along with the issue, which is accessible through the SAS Data Remediation web service API. For more information, see [“Creating a Remediation Issue” on page 80](#).

Working with Search Results

Overview

Successful master data searches return one or more entities (best records).

Figure 6.4 Search Results

1000 entities | Entity Type: PARTY | Search Method: DataFlux Match | Source System : Best Record

MDM Entity Type	MDM Cluster ID	Full Name	Address Line 1 (Standardized)	City
Company	26099	AARONS AEROSPACE	8604 Banana Bay Dr	ORLANDO
Company	31974	AARONS AEROSPACE	10974 Palestina	ASOTIN
Company	20729	AARONS AEROSPACE	130 St (14192)	AFFTON
Company	33078	AARONS AUTO DEALER	PO Box 56 (R1)	LEASBURG
Company	34051	AARONS CAR DEALER	200 Ave (2356)	ONA
Company	33859	AARONS CASINOS	400 Kasby St	ROCKFORD
Company	22511	AARONS COMPANIES	140 Ave (23426)	PENCE

Company: AARONS AEROSPACE			
Full Name:	AARONS AEROSPACE	County:	ORANGE
Website:	http://www.aarons.com	State/Province:	FL
Address Line 1:	8604 BANANA BAY DR	Postal Code:	33908
Address Line 2:		Country:	USA
City:	ORLANDO	Primary Phone Number:	(239) 238-8241
		Secondary Phone Number:	
		Fax Number:	(239) 238-3130
		Primary Email Address:	aarons@yahoo.com
		Secondary Email Address:	

The toolbar at the top of the results table contains an **Actions** menu with the following options:

New Entity

see [“Creating New Entities” on page 78](#) for a description of this option.

Open

opens the entity selected in the results table and displays cluster details about a new tab.

Create Remediation Issue

see [“Creating a Remediation Issue” on page 80](#) for a description of this option.

Retire

retires the selected records.

Tools

opens a dialog box that enables you to access available tools.

Search Method

see [Step 4 on page 64](#) for a description of this option.

Details Pane

displays or hides details of the selected record in a separate pane.

Show Retired Records

toggles between exposing or hiding retired records. When exposed, retired records appear dimmed.

Export

exports the results table in either CSV or PDF format.

Note: Not all characters can be represented in the fonts used in PDF export files. When a character cannot be represented in the PDF file, it is replaced with #. Only Latin characters can be rendered in exported PDF files.

Open New Search Tab

opens a separate tab for a new search.

Refresh

refreshes the table of results.

About SAS MDM

displays application version information.

Using Tools

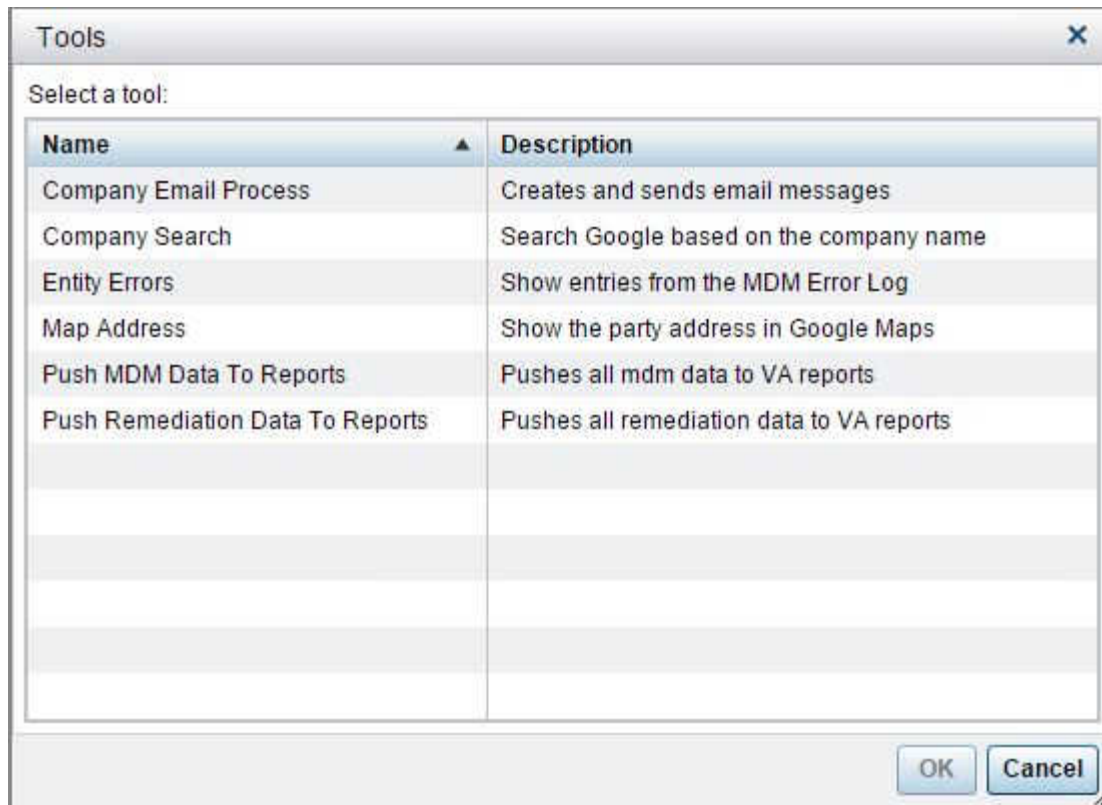
If your administrator has created tools for use with entities, the tools are available from the appropriate **Actions** menus. SAS MDM is installed with several tool examples that work in conjunction with the sample entity types and sample data. Your administrator must design and enable tools for use with new entity types.

Some of the possible uses of tools are:

- initiating a query for transactional data in external systems
- starting a batch process to load data or generate a report
- creating saved queries
- validating address data when editing records
- making changes to a group of records at the same time
- showing a selected address on a Google map
- performing a web search on selected data
- sending several rows of data via e-mail to a colleague for review

When you select **Tools** from the **Actions** menu, a dialog box with a list of available tools opens:

Figure 6.5 Tools Dialog Box



Note: The contents of the list vary depending the type of entity and what your administrator has made available.

Select the tool that you want to use, and then click **OK**. The results of this action vary according to the tool. For example, if you select **Company Search** while editing a Company entity, the tool opens a Google search for the company name in a separate browser tab.

Note: you must enable pop-ups in your browser to see tool results in a different tab.

Editing Entities

After entities are returned from a search, you can edit them. To edit an entity:

- 1 Select a record in the result table.
- 2 Click  and select **Open**.

The cluster of related records opens on a new tab:

Revert Move

reclusters selected contributor records that have been previously moved to another cluster. Reverting a move operation causes the record to recluster against the hub data using the defined cluster conditions. The **Revert Move** option is enabled only when a previously moved record is selected.

Note: If the record being reverted is the last contributor in the cluster, the entire cluster might be retired.

See “[Recluster Editor](#)” on page 72 for more information about recluster.

Show Retired Records

toggles between exposing or hiding retired records. When exposed, retired records appear dimmed.

Details Pane

displays or hides details of the selected record(s) in a separate pane.

Export

exports the records table in either CSV or PDF format.

Note: Not all characters can be represented in the fonts used in PDF export files. When a character cannot be represented in the PDF file, it is replaced with #. Only Latin characters can be rendered in exported PDF files.

Refresh

refreshes the table of records.

The Entity Editor

Overview

The entity editor enables you to make changes to the information for a master data entity.

To open the entity editor:

- 1 Select a record for editing on the **Records** tab.
- 2 Click  and select **Edit**. The entity editor opens.

The toolbar at the top of the entity editor contains an **Actions** menu with the following options:

Save

saves changes to the record.

Close Entity Editor

closes the entity editor. The editor can also be closed by clicking the left

navigation breadcrumb, such as



Create Remediation Issue

see “[Creating a Remediation Issue](#)” on page 80 for a description of this option.

Retire

retires the selected record.

Matrix Edit

toggles between standard Edit mode and matrix Edit mode.

Tools

opens a dialog box that enables you to access available tools.

Reset

resets the record to its initial state without saving any changes.

Standard Edit Mode

The following is a sample standard Edit mode screen:

Figure 6.7 Entity Editor

ABE BINSTEON >> Orion (59)

Save

Matrix Edit

MDM Identifiers

Contact

Full Name:

* ABE BINSTEON

Name Title:

Name Suffix:

Name Appendage:

Do not Call:

Yes

Last Contact:

March 16, 2010 08:20:29 AM

Notes:

Primary Phone Number:

(919) 212-2523

Standardized Primary Phone Number:

(919) 212 2523

Secondary Phone Number:

Standardized Secondary Phone Number:

Primary Email Address:

Secondary Email Address:

Personal Information

Address

Shipping Address

Drives

New Relationship

MDM Entity Type	MDM Cluster ID	Full Name	Do not Call	Last Contact	Address Line 1 (Standardized)	City

In standard Edit mode, you can change and save the values of individual properties for the selected record. If you change your mind after modifying values, click  to reset the all values in the form.

You can also view, add, or retire relationships using the tables at the bottom of the form. See [“Relationship Diagram Tab” on page 74](#) for more information.

Matrix Edit Mode

Matrix Edit mode enables you to move properties to create a new best record from the contributor records in a cluster.

To enter matrix Edit mode:

- Click  and select **Matrix Edit** or click  **Matrix Edit** from the toolbar. The tab changes to matrix Edit mode:

Figure 6.8 Matrix Edit

ABE BINSTEON

Orion (59)

Save

Matrix Edit

MDM Entity ID	Source System	Source System Record ID	Valid From	Valid To	Full Name	Do not Call	Address Line
3969	Orion	59	07/21/13 07:...		ABE BINSTEON	Yes	4900 Jay Matt

Source System	Source System Record ID	Valid From	Valid To	Full Name	Do not Call	Address Line 1 (Standardized)
Best Record	4275	07/21/13 07:0...		ABE BINSTEON	Yes	4900 Jay Matthews Ln
Orion	59	07/21/13 07:0...		ABE BINSTEON	Yes	4900 Jay Matthews Ln
Marketing	61	07/21/13 07:0...		ABE BINSTEON	No	4900 Jim Matthews Ln
ERP	61	07/21/13 07:0...		ABE BINSTEON	No	4900 Jay Matthews Rd

The tab remains in matrix Edit mode until you toggle it back to standard mode.

Note: Differences between properties in the record in the top table and records in the bottom table are automatically highlighted.

- 2 Select a property from a record in the bottom table that you want to move to the corresponding location of the record in the top table.
- 3 Click to move the selected property. Continue to move properties until the record in the top table contains the correct properties for new best record.
- 4 If you change your mind after moving a selected property, click to reset the properties.
- 5 Click Save to save the new best record when you have finished moving properties.

Recluster Editor

The recluster editor enables you to move records to new or different clusters. In some cases, the cluster conditions that have been configured for an entity might not work as intended. Records can be overmatched, resulting in clusters that are too large. Records can also be undermatched, resulting in too many clusters. In the case of overmatched records, you might need to move one or more contributor records to a new cluster. In the case of undermatched records, you might need to move one or more contributor records to an existing cluster.

Moving a record indicates that the cluster conditions are not processing the record as intended, so updates to the record are processed differently. Specifically, an update to a moved record (based on its source system ID and source system record ID) never causes clusters to merge together, even if the update matches multiple clusters. If SAS MDM allowed the merge, the move operation would be undone.

Similarly, if an incoming record matches both the original and target clusters, but does not match based on source system ID and source system record ID, the incoming record is placed in the original cluster. SAS MDM has a default preference for clusters without moved records compared to clusters with moved records.

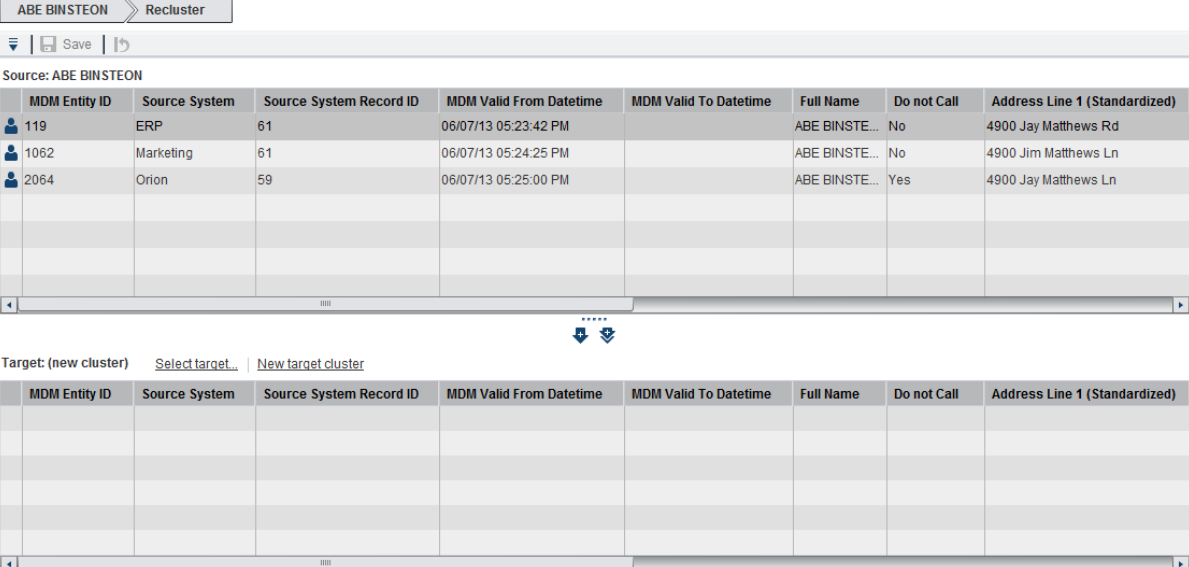
Note: The following apply to clusters:

- Only entities of the same type can be moved to an existing cluster. For example, you cannot move a COMPANY contributor record to an existing PART cluster.
- Survivor records are always generated as part of the operation of the hub and cannot be moved.
- Retired entities (contributor and survivor records) cannot be moved to either a new cluster or an existing cluster.
- Records cannot be moved into retired clusters.
- If you move the last contributor record out of a cluster, the entire cluster is retired.

To open the recluster editor:

- 1 Select a record for reclustering on the **Records** tab.
- 2 Click  and select **Recluster**. The recluster editor opens:

Figure 6.9 Recluster Editor



ABE BINSTEON > Recluster

Save

Source: ABE BINSTEON

MDM Entity ID	Source System	Source System Record ID	MDM Valid From Datetime	MDM Valid To Datetime	Full Name	Do not Call	Address Line 1 (Standardized)
119	ERP	61	06/07/13 05:23:42 PM		ABE BINSTE...	No	4900 Jay Matthews Rd
1062	Marketing	61	06/07/13 05:24:25 PM		ABE BINSTE...	No	4900 Jim Matthews Ln
2064	Orion	59	06/07/13 05:25:00 PM		ABE BINSTE...	Yes	4900 Jay Matthews Ln

Target: (new cluster) [Select target...](#) [New target cluster](#)

MDM Entity ID	Source System	Source System Record ID	MDM Valid From Datetime	MDM Valid To Datetime	Full Name	Do not Call	Address Line 1 (Standardized)

The toolbar at the top of the recluster editor contains an **Actions** menu with the following options:

Save

saves changes to the cluster.

Close Recluster Editor

closes the recluster editor. The editor can also be closed by clicking the left

navigation breadcrumb, such as

ABE BINSTEON > Recluster

Select target

opens a dialog box that enables you to search for and select a target cluster to which to move the selected records.

New Target Cluster

Clears the Target table and enables you to move records to create a new cluster.

Reset

resets the cluster to its initial state without saving any changes.

To move one or more records to another cluster:

- 1 Select either **Select target** or **New target cluster** from the **Actions** menu.
- 2 If you specified **Select target**, choose a target cluster.
- 3 Select one or more records in the Source table to be reclustered.
- 4 Click  to move selected records or  to move all records to the Target table.

Hierarchies Tab

The **Hierarchies** tab enables you to open and view hierarchies for the entity. This list shows only hierarchies where the entity is the root or top-level element.

Figure 6.10 Hierarchies Tab

Records	Hierarchies	Relationship Diagram	History
  			
Name		Description	Type
 Bueno Book Customers		Purchasers of parts from Bueno Books	Manufacturer to Purchaser

The toolbar at the top of the hierarchies table contains an **Actions** menu with the following options:

Open Hierarchy

opens the hierarchy selected in the hierarchies table and displays details about a new tab. See [“Searching Hierarchies” on page 83](#) for more information about hierarchy details.

Refresh

refreshes the table of hierarchies.

Relationship Diagram Tab

Overview

The **Relationship Diagram** Tab enables you to view and create relationships among entities. Entities are referred to as nodes on the relationship diagram.

The toolbar at the top of the **Relationship Diagram** tab contains an **Actions** menu with the following options:

Details Pane

displays or hides details of the selected node in a separate pane.

Fit to Area

centers the relationship diagram in its display pane.

Zoom Out

reduces the size of the relationship diagram.

Zoom In

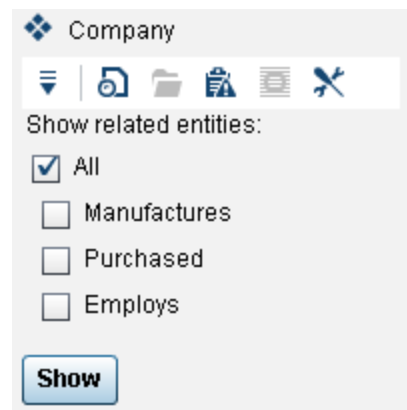
increases the size of the relationship diagram.

Reset Zoom

resets the relationship diagram to its original size.

The Node Control Pane enables you to select what type of relationships to display.

Figure 6.11 Node Control Pane



The toolbar at the top of the **Node Control Pane** tab contains an **Actions** menu with the following options:

Show Related Entities

searches for and displays entities related to the selected entity.

Open Entity

opens the selected entity (node) in an editor.

Create Remediation Issue

see [“Creating a Remediation Issue” on page 80](#) for a description of this option.

Hide Node on Diagram

removes the selected node from view on the diagram.

Tools

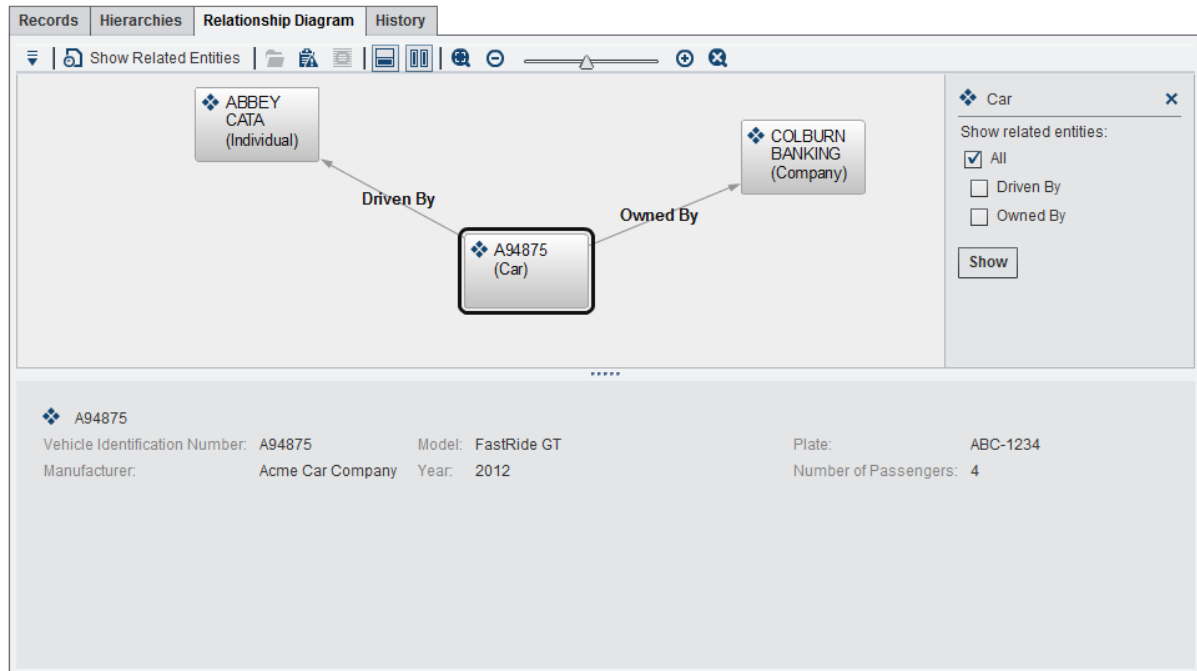
opens a dialog box that enables you to access available tools.

New Relationship

opens a dialog box for creating a new relationship.

After selecting relationship types, click **Show**. The related entities are displayed in nodes on the diagram:

Figure 6.12 Relationship Diagram



You can click and drag a node to modify the pattern of the diagram or press the Shift key while clicking and move the entire graph. You can also press the Control key while clicking to rotate the graph. From this point, you can select a node to create a new relationship.

To create a new relationship:

- 1 Select the node to which you want to create a new relationship.
- 2 Click  and select **New Relationship**. A dialog box appears:

Figure 6.13 New Relationship Dialog Box

The 'New Relationship' dialog box has a title bar with a close button. It contains the following fields and controls:

- Relationship type:** A dropdown menu currently showing 'Manufactures'.
- Entity to relate:** A dropdown menu currently showing 'Part ID', followed by an empty text input field and a 'Search' button.
- Search to find entities.** A label below the search field.
- Buttons:** 'Add' and 'Close' buttons at the bottom right.

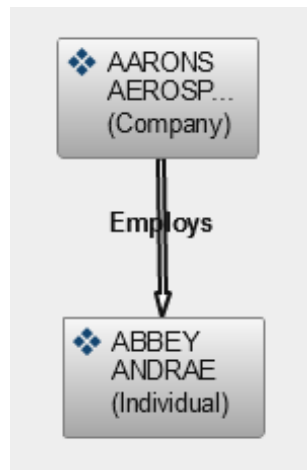
- 3 Fill in the appropriate fields and search to find entities.

- 4 Select the entity or entities with which to establish relationships. Then click **Add**.

Working with Relationships

You can perform actions on a relationship by selecting it in the relationship diagram:

Figure 6.14 Relationship Diagram



When a relationship is selected, the details pane is updated to show the source, target, type, and any configured attributes for the relationship:

Figure 6.15 Relationship Properties

The dialog box is titled 'Relationship - Employs'. It contains the following fields and controls:

- Relationship Properties**
- Source:** AARONS AEROSPACE (Company)
- Target:** ABBEY ANDRAE (Individual)
- Relationship Type:** Employs (indicated by a relationship icon)
- Status:** A dropdown menu with a downward arrow.
- Start Date:** A date input field with a calendar icon.
- Buttons:** OK, Clear, and Cancel.

The Node Control Pane also displays a toolbar that provides options to edit the attributes of the relationship, hide the node, or retire the relationship.

See “[Working with the Relationship Type Editor](#)” on page 113 for more information about the editing relationships.

History Tab

The **History** tab displays the history of insertions, movements, and retirements of records.

Figure 6.16 History Tab

Records | Hierarchies | Relationship Diagram | History

Number of days: 30

MDM Entity ID	Source System	Source System Rec..	Label	Modified By	Modified Date	Action
119	ERP	61	ABE BINSTEON		06/07/13 05:23:42 PM	INSERT
120	Best Record	120	ABE BINSTEON		06/07/13 05:23:42 PM	INSERT
120	Best Record	120	ABE BINSTEON	robarn	06/07/13 05:24:24 PM	RETIRE
1062	Marketing	61	ABE BINSTEON		06/07/13 05:24:25 PM	INSERT
1063	Best Record	1063	ABE BINSTEON		06/07/13 05:24:25 PM	INSERT
1063	Best Record	1063	ABE BINSTEON	robarn	06/07/13 05:25:00 PM	RETIRE
2064	Orion	59	ABE BINSTEON		06/07/13 05:25:00 PM	INSERT
2065	Best Record	2065	ABE BINSTEON		06/07/13 05:25:00 PM	INSERT

The toolbar at the top of the **History** tab contains an **Actions** menu with the following options:

Refresh

refreshes the table of records.

Export

exports the records table in either CSV or PDF format.

Note: Not all characters can be represented in the fonts used in PDF export files. When a character cannot be represented in the PDF file, it is replaced with #. Only Latin characters can be rendered in exported PDF files.

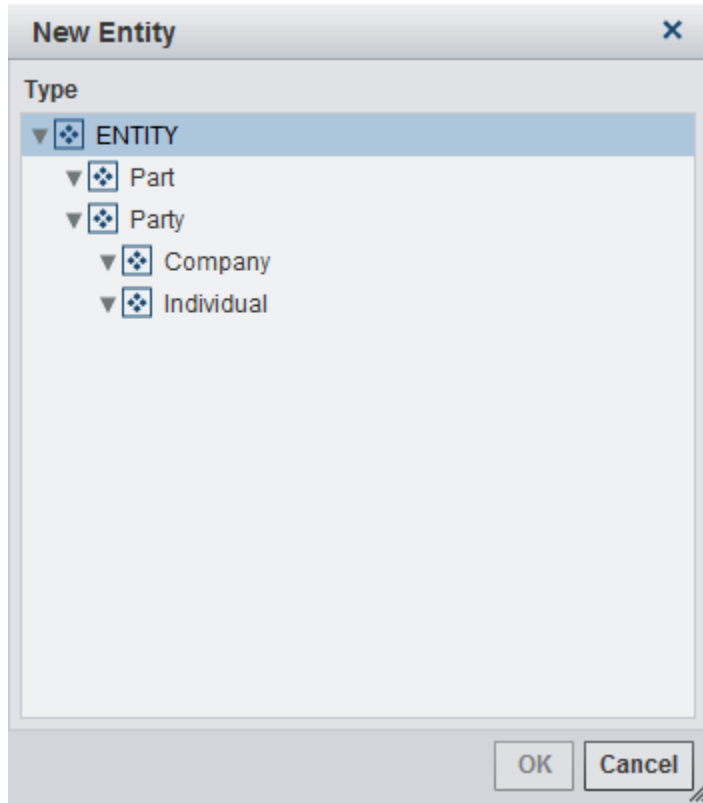
The toolbar also contains a **Number of Days** field to filter the table of records based on the number of days since the record was modified. Set this field to 0 to show all records.

Creating New Entities

To create a new entity:

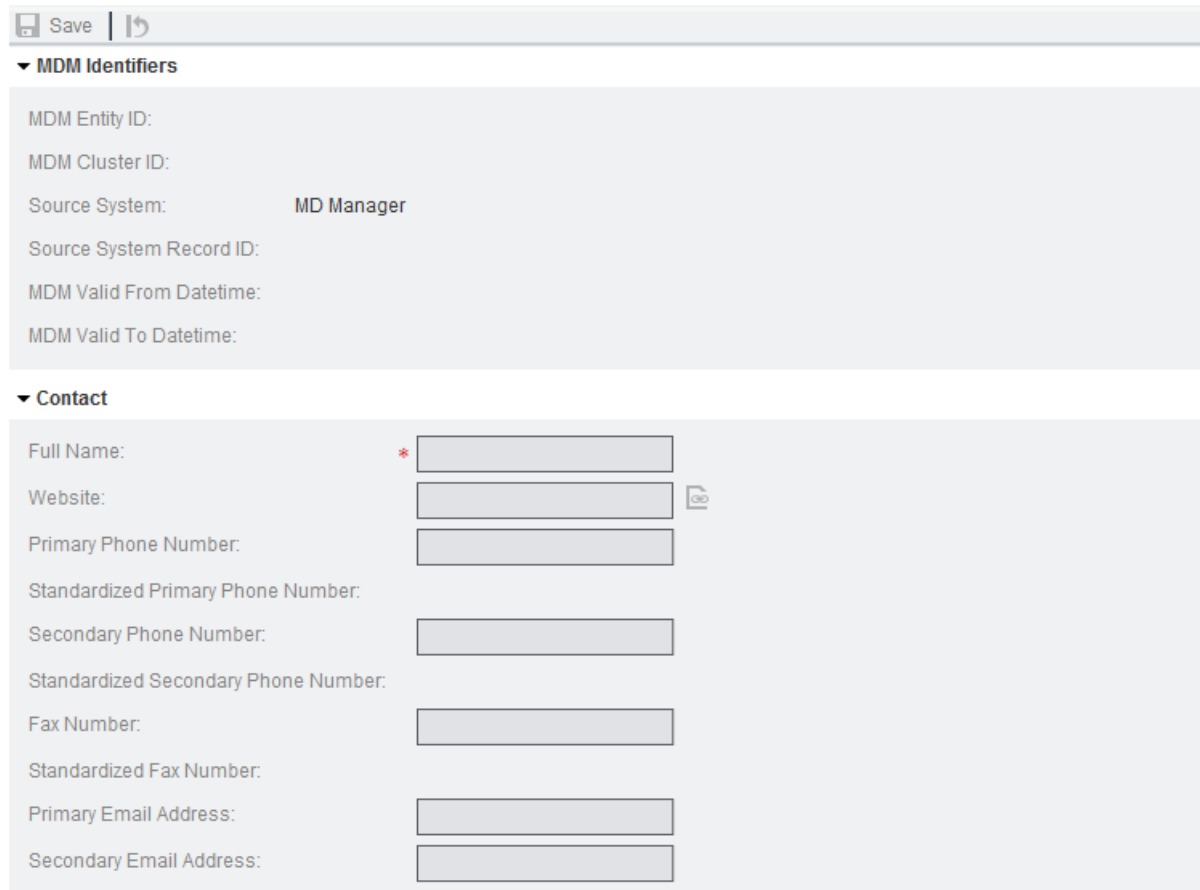
- 1 On the toolbar at the top of the **Master Data Management** tab, click  to display the **New Entity** dialog box:

Figure 6.17 New Entity Dialog Box



- 2 Select the type of entity. Then click **OK** to display the entity editor:

Figure 6.18 Entity Editor



The screenshot shows the Entity Editor interface. At the top, there is a toolbar with a 'Save' button (floppy disk icon) and a 'Clear' button (circular arrow icon). Below the toolbar, the interface is divided into two main sections: 'MDM Identifiers' and 'Contact'.

MDM Identifiers Section:

- MDM Entity ID: (text input field)
- MDM Cluster ID: (text input field)
- Source System: **MD Manager** (text input field)
- Source System Record ID: (text input field)
- MDM Valid From Datetime: (text input field)
- MDM Valid To Datetime: (text input field)

Contact Section:

- Full Name: (text input field with a red asterisk indicating a required field)
- Website: (text input field with a 'Go' button icon)
- Primary Phone Number: (text input field)
- Standardized Primary Phone Number: (text input field)
- Secondary Phone Number: (text input field)
- Standardized Secondary Phone Number: (text input field)
- Fax Number: (text input field)
- Standardized Fax Number: (text input field)
- Primary Email Address: (text input field)
- Secondary Email Address: (text input field)

3 After filling in the appropriate fields, do one of the following:

- Click  **Save** to save the new entity.
- Click  to clear the fields.

Creating a Remediation Issue

You can create a remediation issue to address invalid or missing data. This option is available by selecting **Create Remediation Issue** from the appropriate

 drop-down menu or by clicking  on the toolbar. Either action displays the **Create Remediation Issue** dialog box:

Figure 6.19 Create Remediation Issue

Create Remediation Issue

Package name: *

Subject area:

Issue: *

Importance:

Assignee:

Due date:

Notes

☐ Select a task template to address the issue:

Complete the following items in the dialog box. Then click **OK** to create the remediation issue.

Package name [required]

is a container for one or more issues that have been identified in SAS MDM. Using a unique name here enables you to see issues that were created at the same time grouped together in SAS Data Remediation.

Subject Area

is the name or label of a category or group within an application. This allows problems to be reported against distinct entity types in SAS MDM. In SAS Data Remediation, you can group by subject area to see all issues related to a particular SAS MDM entity type.

Issue [required]

select an issue type from the drop-down list or type in a new issue.

Importance

select an importance status from the drop-down list.

Assignee

click to select an assignee for the issue. The assignee list contains those individuals who have been defined as users of SAS Data Remediation in SAS Management Console.

Due Date

click to select a due date for resolution of the issue.

Notes

add any relevant notes concerning the issue.

Select a task template to address the issue

select this check box to use a template to address the issue. Then choose a template from the drop-down list. See the *SAS Data Remediation User's Guide* for more information.

Creating a data remediation issue sends the issue to the data remediation environment where it can be reviewed and corrected. If you select a task to associate with the data issue, a user-defined workflow can be associated with the data issue, and the issue is routed as defined by the workflow template. You can use SAS Data Remediation to review all outstanding data issues, assign them to others, correct them or act on them, depending on the custom tasks that were associated with the issue.

7

Using Hierarchies

Searching Hierarchies	83
Overview	83
Quick Search	83
Field Search	84

Searching Hierarchies

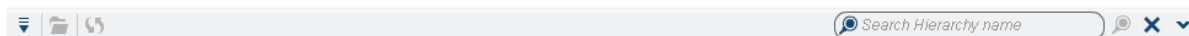
Overview

Hierarchies are used to display relationships between entities. For example, you can build hierarchies between a household and the people that constitute that household, or between a company and its suppliers.

Note: The **Hierarchies** tab is available only to users who have View Application capability.

To access hierarchy searches, click **Hierarchies** on the SAS Data Management Console page. By default, the tab opens to the search toolbar:

Figure 7.1 Hierarchy Search Toolbar



You can perform either a quick search or a field search.

Quick Search

To perform a hierarchy quick search:

- 1 Enter the hierarchy name into the search field.

Note: The search field is case-insensitive and, by default, searches on even a single letter as a wildcard. For example, searching on the letter *B* returns search results for any type of hierarchy with a name that contains either *B* or *b*.

- 2 Do one of the following:

- Click  to begin the search.
- Click  to clear the search data.

Hierarchies are returned in a table.

Field Search

To perform a hierarchy field search:

- 1 Click  on the right side of the toolbar to expand the search pane.
- 2 Enter appropriate search criteria in the fields of the search pane.

Figure 7.2 Search Pane

Hierarchy Type:

Manufacturer to Purchaser | ▼

Name:

Bueno Book Customers

Description:

Note: The search fields are case-insensitive and, by default, search on even a single letter as a wildcard. For example, searching on the letter **B** in the **Name** field returns search results for any type of hierarchy with a name that contains either **B** or **b**. If you want to limit the types of hierarchies returned, you must specify the hierarchy type in the **Hierarchy Type** field.

- 3 Do one of the following:
 - Click  to begin the search.
 - Click  to clear the search data.

Hierarchies are returned in a table.

Figure 7.3 Search Results

Search Hierarchy Type : 'Manufacturer to Purchaser'

▼ | 📁 | ↺

Search Hierarchy name

🔍 | ✕ | ▼

Name	Description	Root Entity	Type
 Aarons Aerospace Parts		AARONS AEROSPACE	Manufacturer to Purchaser

The toolbar at the top of the results table contains an **Actions** menu with the following options:

Open
opens the hierarchy selected in the results table on a new tab.

Export
exports the results table in either CSV or PDF format.

Note: Not all characters can be represented in the fonts used in PDF export files. When a character cannot be represented in the PDF file, it is replaced with #. Only Latin characters can be rendered in exported PDF files.

Open New Search Tab
opens a separate tab for a new search.

Refresh

refreshes the table of results.

About SAS MDM

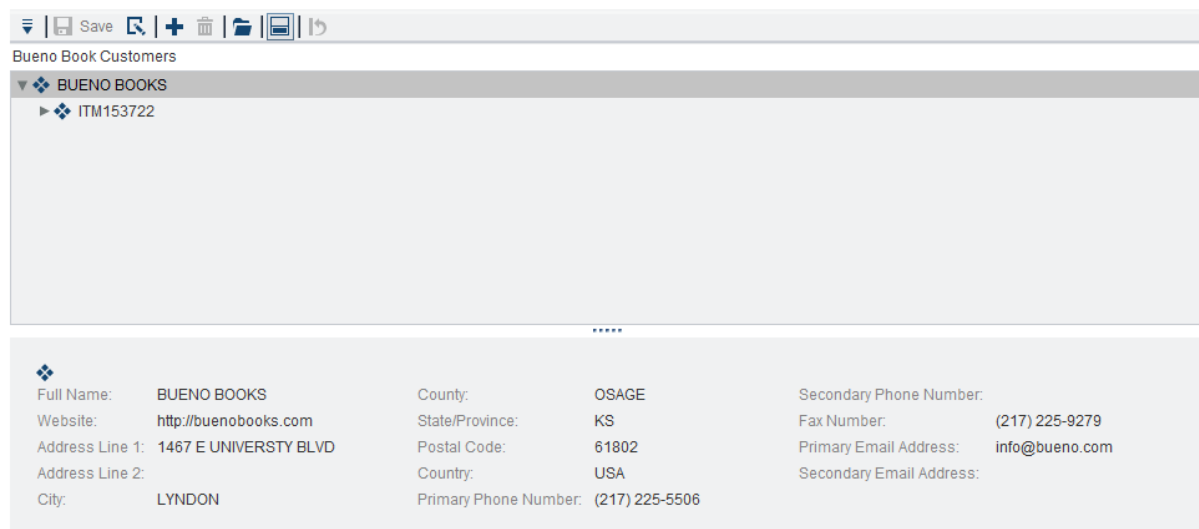
displays application version information.

After hierarchies are returned from a search, you can edit them. To edit a hierarchy:

- 1 Select a hierarchy in the result table.
- 2 Click  and select **Open**.

The hierarchy opens on a new tab:

Figure 7.4 Search Results



The toolbar at the top of the hierarchy editor contains an **Actions** menu with the following options:

Save

saves all changes that have been made to the hierarchy.

Name

opens a dialog box that enables you to change the name and description of the hierarchy. Naming hierarchies enables you to search on them later and locate them more easily. Names are not required.

Tools

opens a dialog box that enables you to access available tools for the entities in the hierarchy.

Add Entity (New Relationship)

opens a dialog box that enables you to add a new relationship, which also adds the new relationship to the relationship diagram. See [“Relationship Diagram Tab” on page 74](#) for more information.

Remove Entity (Retire Relationship)

retires the selected relationship, which also removes the relationship from the relationship diagram. See [“Relationship Diagram Tab” on page 74](#) for more information.

Open Entity

opens the selected item for editing on a separate tab.

Details Pane

displays or hides details of the selected item in a separate pane.

Reset

resets the record to its initial state without saving any changes.

8

Using Source Systems

About Source Systems	87
Working with Source Systems	88
Overview	88
Running Source System Tools	89

About Source Systems

From the **Source Systems** tab, you can view source systems and the entity types that they contribute. Selecting an entity type in the left pane enables you to run tools defined on the entity type. The right pane shows the execution history of add/update source system tools.

Note: The **Source Systems** tab is available only to users who have View Application and View Source System capabilities.

To access the **Source Systems** tab, click **Source Systems** on the SAS Data Management Console page. The following display appears:

Figure 8.1 Source Systems Tab

All Source Systems

Best Record

MD Manager

ERP

Part

Individual

Company

Marketing

Orion

Call Center UK

Call Center US

CRM

Add/Update History: ERP - Company

Number of days: 30

	Start Date	Elapsed TI...	Source Sy...	Entity	User	Records	Added to Hub	Send for Re...	Issues/Record
	08/28/14 1...	00:03:03	ERP	Company	Roger...	14629	14629	0	
.....									

Actions: ERP - Company

Tool	Description
▶ Remaining Records	Retrieve the number of company remaining records from the source system
▶ Add/Update	Extract, standardize, and prepare company data from the source system to MDM hub

Note:

- The source systems in the display are samples and might not match the definitions in your hub.
- If you have created a new source system definition (“[Working with the Source Systems Editor](#)” on page 129) within the same MDM session, it is not immediately displayed on the **Source Systems** tab. You must refresh your browser for the newly created source system definition to be displayed on the **Source Systems** tab.

Working with Source Systems

Overview

Available source systems are displayed in the left pane. Expanding the source system displays the entity types contributed by the source system. The upper right pane displays the Add/Update History of the item selected in the left pane.

Note: Only published entity types are displayed.

Selecting an entity type under a source system displays an Actions table in the lower right pane listing the source system tools available for the selected entity type.

The entity types contributed by the source system and the associated source system tools are defined by administrators using the source systems editor. See [“Working with the Source Systems Editor” on page 129](#) for more information about the source systems editor.

The Add/Update History table displays a snapshot of the results of each **Add/Update** source system tool execution. Clicking  displays the most recent result. You can limit the number of historical entries displayed on the Add/Update History table using the Number of days filter. The default value is 30.

The first column in the table indicates the status of the **Add/Update** tool execution:

-  indicates that the tool is still executing.
-  indicates an error during the execution. Holding the mouse pointer over the icon displays the error message.
- No icon indicates a successful execution.

Clicking  refreshes the status display of the currently executing entry.

The Add/Update History table gives users a quick view of the data quality coming into the hub. The columns in the table are as follows:

Start Date

the date and time at which the **Add/Update** tool was run.

Elapsed Time

the time elapsed since the **Add/Update** tool started. After execution, the total elapsed time that the tool ran.

Source System

the source system from which the entity type records are loaded.

Entity

the name of the entity type.

User

the name of the user running the tool.

Records

the number of records being processed.

Added to Hub

the number of records contributed to the hub.

Send for Remediation

the number of records that have issues and were sent for remediation.

Issues/Record

the average number of issues per record coming from the source system.

Note: Add/Update History is not applicable to the Best Record source system. Add/Update History contains entries for the MD Manager source system only when a new entity type record is created or updated within the SAS MDM hub.

Running Source System Tools

Overview

The tools that are available to you are configured by your administrator. By default, an add_update tool is associated with every entity type contributed by a source system. However, your administrator can change this or add new tools.

Note: Running source system tools is available only to users who have View Application and Run Source System Tools capabilities.

To run a source system tool:

- 1 Click  next to the tool name. A **Run Tool** dialog box displays the name, description, and input parameters, if any, of the tool.
- 2 Click **Run** in the dialog box.

Running the Add/Update Tool

Add/Update is a default source system tool. It is created by the source system when an administrator defines an entity type in the source system editor. The purpose of the **Add/Update** tool is to extract the entity type records from the source system, standardize, and prepare them, and then load them to the hub.

To run the **Add/Update** tool:

- 1 Click  next to the **Add/Update** tool name. A **Run Tool** dialog box is displayed.
- 2 In the **Run Tool** dialog box, enter a number greater than zero in the **LOAD_LIMIT** field. The **LOAD_LIMIT** field specifies how many records to load to the hub.

Note: The default value is 30000 records. You can change this to a number applicable to your data. If the **LOAD_LIMIT** field is left empty, the tool uses the **LOAD_LIMIT** value set within the job definition. For example, you can open the job sasmdm/add_update_part.djf in Data Management Studio and see the value of **LOAD_LIMIT** in the Echo node. The job or service associated with the tool is specified in the Job/service field on the **Tools** tab of the source system editor. See [“Using the Tools Editor” on page 131](#) for more information.

- 3 Click **Run** in the dialog box. The batch job begins and displays an **Add/Update** dialog box indicating that the batch tool started successfully.
- 4 Click **Close** in the dialog box.

Note: The dialog box indicates only that the tool started successfully. Clicking  on the Add/Update History table displays the current execution status.

The **Add/Update** tool loads the data incrementally by the number of records specified by the LOAD_LIMIT parameter. This incremental loading avoids issues created by loading large volumes of data in a system with limited system resources.

After all the data have been loaded, running the **Add/Update** tool again results in batch job termination returning the message **No more data to process**.

The sample jobs use the extract parameter to support the incremental load using the following process:

- 1 The data from the source system is ordered by Source System Record ID (SRC_SYS_REC_ID). The LAST_TOKEN_VALUE in the MDM_EXTRACT_TOKEN database table represents the last row number read from the source system for a particular entity type.
- 2 The next time the extract job is invoked by the **Add/Update** tool, it loads data from row number LAST_TOKEN_VALUE + 1.

Note: This works because the sample data uses numeric identifiers for the source system record ID. Depending on your data, you might have to modify the interpretation of the extract parameter.

If you want to reload the data after all of it has been processed, you can manually update the LAST_TOKEN_VALUE field of the specific source system and entity type in the MDM_EXTRACT_TOKEN database table. To reload from the beginning, you must set LAST_TOKEN_VALUE to 0, and then run the **Add/Update** tool again.

You can see, and modify, the definition of the **Add/Update** tool on the **Tools** tab of the source systems editor. See [“Working with the Source Systems Editor” on page 129](#) for more information.

Running the Remaining Records Tool

Remaining Records is provided as an example of a custom source system tool. The purpose of the **Remaining Records** tool is to retrieve the remaining number of entity type records in the source system, since the last load, that are available to load to the hub. **Remaining Records** is particularly useful for determining the number of records to use as the value for the LOAD_LIMIT field in the **Add/Update** tool.

Note: The sample Remaining Records tool is provided for only the following sample source systems and entity types:

Table 8.1 Remaining Records Availability

Source Systems	Entity Types
ERP, Marketing, Orion, and Call Center UK.	Company, Individual, and Part

To run the **Remaining Records** tool:

- 1 Click  next to the **Remaining Records** tool name. A **Run Tool** dialog box is displayed.

- 2 Click **Run** in the dialog box. The job begins and displays a **Remaining Records** dialog box indicating the number of records remaining to load to the hub.
- 3 Click **Close** in the dialog box.

You can see, and modify, the definition of the **Remaining Records** tool on the **Tools** tab of the source systems editor. See [“Working with the Source Systems Editor” on page 129](#) for more information.

9

Using Master Data Management Data Model

About Master Data Management Data Model	93
Overview	93
Importing and Exporting	95
Managing Definition Types	95
Understanding Entity Type Definitions	96
Working with the Entity Type Editor	96
Overview	96
Entity Type Editor Tabs	100
Using the Definition Editor	100
Using the Clustering Editor	107
Using the Tools Editor	108
Using the Hierarchy Types Tab	112
Using the Relationship Types Tab	112
Using the Source Systems Tab	112
Understanding Relationship Type Definitions	113
Working with the Relationship Type Editor	113
Overview	113
Relationship Type Editor Tabs	115
Using the Definition Editor	115
Using the Rules Editor	119
Understanding Hierarchy Type Definitions	120
Working with the Hierarchy Type Editor	121
Overview	121
Hierarchy Type Editor Tabs	122
Using the Properties Editor	122
Using the Levels Editor	123

About Master Data Management Data Model

Overview

From the **Data Model** tab, data administrators can maintain model information for the hub. This information includes the metadata definitions of entity types,

relationship types, and hierarchy types and is used to manage and present information in the hub and SAS MDM.

Note: The **Data Model** tab is available only to users who have View Application, Manage Entity Types, and Manage Relationship Types capabilities.

To access the **Data Model** tab, click the **Data Model** link under Master Data Management on the **Data Management Console** page. The following display appears:

Figure 9.1 Data Model Tab

☐ Highlight related types across lists

Entity Types

New Entity Type

Type	Display Name	Description	Last Published	Valid From	Valid To
PARTY	Party	Party	08/28/14 10:00:00 AM	08/28/14 10:00:00 AM	
INDIVIDUAL	Individual	Individual	08/28/14 10:00:04 AM	08/28/14 10:00:04 AM	
COMPANY	Company	Company	08/28/14 10:00:07 AM	08/28/14 10:00:07 AM	
PART	Part	Part	08/28/14 10:00:00 AM	08/28/14 10:00:00 AM	
ENTITY	ENTITY	Abstract Entity Type	08/28/14 09:58:52 AM	08/28/14 09:58:52 AM	

Relationship Types

New Relationship Type

Type	Display Name	Description	Source Entity ...	Target Entity T...	Last Published	Valid From	Valid To
MANUFACTURER	Manufactures	Company manufactures ...	Company	Part	08/28/14 10:00...	08/28/14 10:00...	
PURCHASER	Purchased	Party purchases Part	Party	Part	08/28/14 10:00...	08/28/14 10:00...	
EMPLOYS	Employs	Company employs Indivi...	Company	Individual	08/28/14 10:00...	08/28/14 10:00...	

Hierarchy Types

New Hierarchy Type

Type	Display Name	Description	Valid From	Valid To
MANUFACTURER TO PURCHASER	Manufacturer to Purchaser	Displays Manufacturer to Purchaser tree	08/28/14 09:59:55 AM	

Note: In the following descriptions, **Actions** menu items are accessed by clicking from the toolbar. Most functions available from the **Actions** menu are also available from icons on the same toolbar. Clicking the icon performs the function.

The toolbar at the top of the **Data Model** tab contains an **Actions** menu with the following options:

Highlight related types across lists

highlights related types across the lists in the tables. For example, selecting the COMPANY entity type highlights the MANUFACTURER and EMPLOYS relationship types, both of which refer to the COMPANY entity type.

Import Model

opens a dialog box from which you choose an entity definition file (.met file) for import.

Export Model

opens a dialog box from which you choose the parts of the current data model that are to be exported to an entity definition file (.met file). Only active objects are shown. You cannot export retired objects.

About SAS MDM

displays application version information.

Importing and Exporting

Users of the Master Data Management Foundation feature of DataFlux Data Management Studio can also export entity type definitions into .met files, which SAS MDM can import to create new entity types. For more information about the Master Data Management Foundation feature, see the *DataFlux Data Management Studio User's Guide*.

After you choose a .met file from **Import Model** dialog box, the contents of the file are parsed and presented in another dialog box from which you can choose the objects, such as entity types, to import into the hub. When objects are selected in this dialog box, their dependencies are automatically included as well. For example, selecting the sample MANUFACTURES hierarchy type references the PART and COMPANY entity types, which are derived from ENTITY and PARTY respectively.

When you import metadata using this interface, the objects in the hub are merged with the objects represented in the .met file. Importing does not cause any previously existing objects to be retired; importing only adds to the hub. Any objects that were previously retired in the hub but are present in the .met file are left unchanged.

Similarly, when you select objects in the **Export Model** dialog box, their dependencies are automatically included.

Managing Definition Types

Here are the metadata definition types managed from the **Data Model** tab:

entity types

user-defined entities and their attributes. Attributes are data elements for their entity types, in much the same way that columns are part of a table in a relational database.

relationship types

user-defined relationships that exist between the associated entity types.

hierarchy types

user-defined hierarchies of entities.

Each definition type table contains several entries. Selecting an entry under either of the definition types and clicking **Open** from its **Actions** menu opens an editor on a separate tab. For example, selecting COMPANY under **Entity Types** opens an entity editor:

Figure 9.2 Entity Editor Properties Tab

The individual components of the editors are described in the following sections.

Understanding Entity Type Definitions

Entity type definitions enable you to configure metadata for the hub and Master Data Management to define entity-specific attributes for clustering, match codes, and standardization. For step-by-step examples that demonstrate how to create, generate jobs for, and publish an entity type definition, see [Chapter 12, “Working with SAS MDM,”](#) on page 147.

Working with the Entity Type Editor

Overview

The entity type editor is used to create a new entity type definition, to edit an existing definition, or to retire a definition. From the entity type editor, you can collaborate on an entity type definition and publish the definition when you are ready to move it into production. This cycle can be repeated as necessary to achieve the correct definition for an entity type.

Note:

- A retired item becomes inactive, but is not deleted from the hub. It can be viewed as read-only, but cannot be edited. When an entity type is retired, all the entity types that are derived from it are also retired, along with the attributes, clustering conditions, and relationship types for the entity type or types.

- You cannot retire an entity type that is part of a hierarchy type. You must first disassociate the entity type from the hierarchy type. See [“Understanding Hierarchy Type Definitions”](#) on page 120 for more information.

The entity type editor is accessed from the **Entity Types** table:

Figure 9.3 Entity Types Table

Entity Types

▼ | + New Entity Type | 📁

Type	Display Name	Description	Last Published	Valid From	Valid To
⊞ ENTITY	ENTITY	Abstract Entity Type	06/07/13 04:18:37 ...	06/07/13 04:18:37 ...	
⊞ PARTY	Party	Party	06/07/13 04:19:59 ...	06/07/13 04:19:59 ...	
⊞ INDIVIDUAL	Individual	Individual	06/07/13 04:19:59 ...	06/07/13 04:19:59 ...	
⊞ COMPANY	Company	Company	06/07/13 04:19:59 ...	06/07/13 04:19:59 ...	
⊞ PART	Part	Part	06/07/13 04:19:59 ...	06/07/13 04:19:59 ...	

Note: Entity type definition and entity type are equivalent terms.

The toolbar at the top of the **Entity Types** table contains an **Actions** menu with the following options:

New Entity Type

creates a new entity type.

Open

opens the existing entity type selected in the table.

Retire

retires the entity type selected in the table.

Show Retired

displays the retired entity types in the table.

Generate Jobs

generates default batch jobs and data services in support of the entity type. The generation process uses the metadata that has been saved in the hub and publishes .djf and .ddf files to your DataFlux Data Management Server. Changes to the entity type metadata definition (including the metadata for any parent types) are not fully functional until the jobs for the type and any derived types have been regenerated.

Publish

makes the entity type active. This includes generating the tables used to store the data in your DBMS, as well as updating the metadata that the Master Data Management references when determining which entity types to display.

Export

exports the **Entity Types** table in either CSV or PDF format.

Note: Not all characters can be represented in the fonts used in PDF export files. When a character cannot be represented in the PDF file, it is replaced with #.

Selecting **New Entity Type** or **Open** from the **Actions** menu opens the entity type editor on a new tab. For example, opening the COMPANY definition displays the following:

Figure 9.4 Entity Type Editor

The toolbar at the top of the tab contains an **Actions** menu with the following options:

Save

saves the entity type. You can make changes to the entity type by adding attributes as needed.

Retire

retires the entity type.

Generate Jobs

generates default batch jobs and data services in support of the entity type. The generation process uses the metadata that has been saved in the hub and publishes .djf and .ddf files to your DataFlux Data Management Server. Changes to the metadata definition (including the metadata for any parent types) are not fully functional until the jobs for the type and any derived types have been regenerated.

Publish

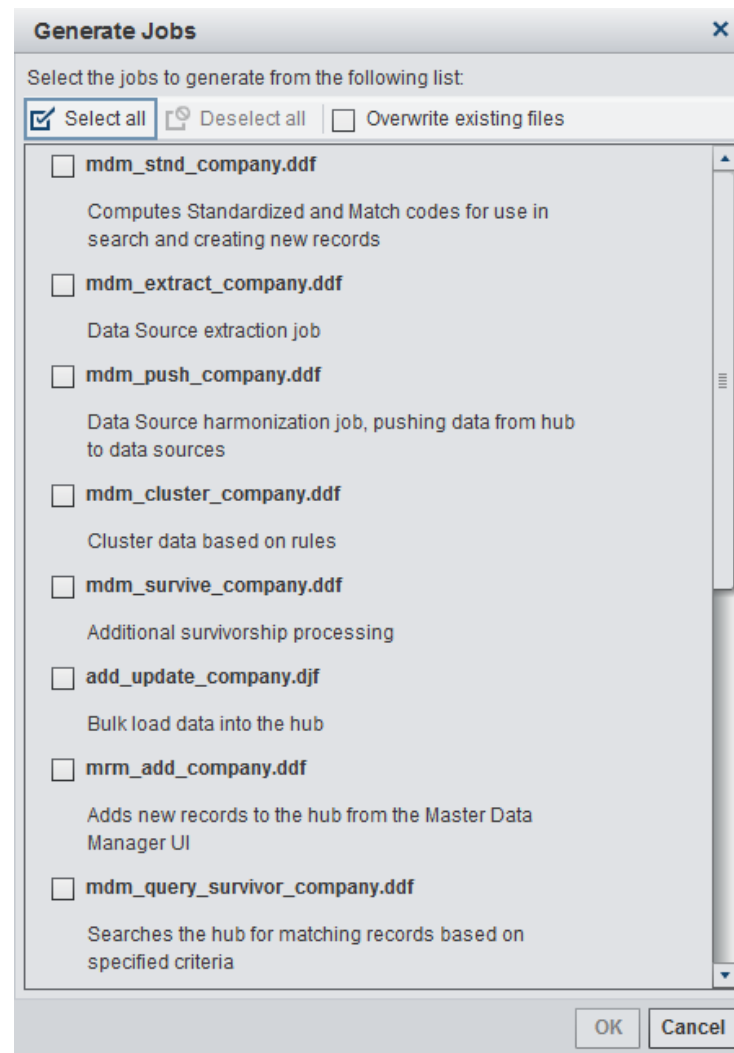
makes the entity type active. This includes generating the tables used to store the data in your DBMS, as well as updating the metadata that the Master Data Management references when determining which entity types to display.

Reset

discards changes and resets the entity type editor to its original state.

The **Generate Jobs** option presents a dialog box from which you can select the jobs that you want to generate:

Figure 9.5 Generate Jobs Dialog Box



For abstract entity types, not all jobs are required. For example, abstract entity types generally support search operations but do not support any of the other jobs. Also, you might not want to regenerate the jobs if you have customized them for your needs

Note: Use the **Overwrite existing files** check box with caution. If you select the **Overwrite existing files** check box, files, including customized files, are replaced on your DataFlux Data Management Server, if they exist.

If you are generating jobs from the **Entity Types** table and have selected more than one entity type in the table, the selections that you make apply to every entity type in the selection. The job names appear as '<MULTIPLE>' instead of the entity type to emphasize that you are generating jobs for multiple entity types.

These jobs generally need to be customized to incorporate your standardization and match code generation requirements, cluster conditions, and survivorship rules. Download the batch jobs and services to your DataFlux Data Management Studio environment, and modify them. Then, republish them on the server.

Note: Changes that have an impact on the model of the entity type in the SAS MDM target database must be republished.

The following guidelines apply to publishing, republishing, and saving:

- Publish is required when you create a new entity type.
- Republish is required when you add a new attribute.
- Republish is required when you change cluster conditions.
- Saving, not republishing, is required when you:
 - change the label, description, or sort order of the entity type
 - change the label, description, group, display orders, read-only bit, or constraint of the attribute
 - modify survivorship rules

To see the result of changes after publishing, republishing, or saving items, you must refresh or restart SAS MDM.

Entity Type Editor Tabs

The entity type editor contains the following tabs:

- **Definition**, from which you can view and modify the properties and attributes for an entity type.
- **Clustering**, from which you can create and modify cluster conditions for an entity type.
- **Tools**, from which you can create and modify tools for an entity type definition.
- **Hierarchy Types**, from which you can create and modify hierarchy types for an entity type definition.
- **Relationship Types**, from which you can create and modify relationship types for an entity type.
- **Source Systems**, from which you can view the sources that contribute this entity type to the hub. The **Source Systems** tab is not available for abstract entity types.

Using the Definition Editor

Overview

The definition editor is used to view and modify properties and attributes of an entity type. It is also used to specify which groups have permission for access to specific entities and attributes. To display the definition editor, click the **Definition** tab in the entity type editor.

Figure 9.6 Entity Types Definition Editor

The screenshot shows the Entity Types Definition Editor with the 'Definition' tab selected. On the left, a tree view lists attributes for the 'COMPANY' entity, including MDM_ENTITY_TYPE_ID, MDM_VALID_TO_DTTM, MDM_VALID_FROM_DTTM, MDM_LANGUAGE_ID, MDM_SRC_SYS_REC_ID, MDM_SRC_SYS_ID, MDM_ENTITY_CLUSTER_ID, MDM_ENTITY_ID, NAME, ADDRESS1, ADDRESS2, CITY, COUNTY, STATE_PROVINCE, POSTALCODE, COUNTRY, PHONE1, PHONE2, EMAIL1, EMAIL2, WEBSITE, and FAX. The main pane is divided into 'Properties' and 'Permissions' tabs, with 'Properties' currently active. The 'Entity Type Properties' section includes fields for Type (COMPANY), Description (Company), Parent Type (PARTY), Last Modified (01/26/15 09:55:45 AM), Valid From (01/26/15 09:54:38 AM), Last Published (01/26/15 09:54:38 AM), and Valid To. There is a checkbox for 'Entity is an abstract entity (cannot be instantiated)'. The 'Display Settings' section includes Display name (Company), Sort Order (300), and Attribute to use as entity label (NAME). The 'Remediation User-defined Fields' section lists three fields: ADDRESS1, PHONE1, and FAX. The 'Source System Harmonization' section includes a radio button selection for 'All source systems that can contribute records to clusters of this entity type', 'Source systems that contributed records to the clusters being harmonized', and 'Select source systems:'. Below this is a list of source systems with checkboxes: MD Manager, Call Center UK, Orion, Marketing, ERP, and Call Center US.

The definition editor includes a pane on the left, under the toolbar, displaying the entity (for example, COMPANY) and listing its attributes. On the right are two additional tabs, **Properties** and **Permissions**. The **Properties** tab displays the properties of the item selected in the left pane, either the entity or one of its attributes. The **Permissions** tab displays the permissions on the item selected in the left pane, either the entity or one of its attributes. By default, the definition editor opens to the **Properties** tab of the entity.

Entity Properties Tab

When the entity is selected in the left pane, the **Properties** tab displays the following fields:

Type [required]

the entity type that is used by the hub, bulk jobs, data services, stored procedures, job generation, database naming conventions, and processes that run outside of SAS MDM.

Description

a description of the entity type.

Parent Type

the parent entity type that contributes metadata to the entity type. All entity types ultimately derive from the ENTITY entity type, which is the default.

Valid From [read-only]

the date from which the entity type is active. This date is automatically set when a new entity type is created, and is automatically updated when the entity type is saved.

Last Modified [read-only]

the date and time at which the entity type was last updated. This date is automatically updated when the entity type is saved.

Last Published [read-only]

the date and time at which the entity type was last published.

Valid To [read-only]

the date on which the entity type is inactive or retired. This date is automatically updated when the entity type is retired.

Entity is an abstract entity (cannot be instantiated)

indicates whether the entity type is an abstract entity type. Selecting this option means that concrete instances of the entity type cannot be created. Deselecting this option means that concrete instances of the entity type can be created.

Display Name [required]

the entity type display name, listed under the Label column in the **Entity Types** table.

Sort Order [required]

the entity type's sort order in selection lists. To hide the entity type and its associated attributes in selection lists, enter either zero (0) or a negative value.

Attribute to use as entity label [required]

selecting an attribute from the drop-down list indicates that this attribute is to be used as the default for a SAS MDM search. The display name of this label attribute appears in the search field. A label attribute must be designated for each entity type. For example, you can choose to use either the company name or a stock ticker as the label attribute for a COMPANY entity type.

Note: The following apply to label attributes:

- A label attribute must be a string data type; non-string attributes do not appear in the drop-down list.
- Retired attributes cannot be used as a label attributes.
- Encrypted cannot be used as label attributes.
- To be selected as a label attribute, the **Type** field of that attribute must contain 100 characters or less.

User-defined field 1

The attribute that is provided to the Data Remediation Service for display. The user-defined fields are used to provide context about the record in remediation within the SAS Data Remediation user interface. These values are used during job generation to ensure that the data provided to SAS Data Remediation is specified consistently.

User-defined field 2

The attribute that is provided to the Data Remediation Service for display. The user-defined fields are used to provide context about the record in remediation within the SAS Data Remediation user interface. These values are used during job generation to ensure that the data provided to SAS Data Remediation is specified consistently.

User-defined field 3

The attribute that is provided to the Data Remediation Service for display. The user-defined fields are used to provide context about the record in remediation within the SAS Data Remediation user interface. These values are used during job generation to ensure that the data provided to SAS Data Remediation is specified consistently.

Source System Harmonization

enables you to select which source systems to update when harmonizing SAS MDM hub clusters:

- All source systems that can contribute records to clusters of this entity type.
- Source systems that contributed records to the clusters being harmonized.
- Specified source systems.

During the entity type job generation, a push job template is also generated. Data Stewards can modify the push job template to push data from the hub to data sources as specified by the **Source System Harmonization** setting. When the push job is in place, changes made in the hub to the entity type are harmonized to the data sources.

SAS MDM provides a sample scenario to demonstrate source system harmonization for the INDIVIDUAL sample entity type. The source system harmonization sample requires you to set up the SAS MDM sample data as described in the *SAS MDM Administrator's Guide*. Assume that Call Center US needs to collect all the survivor records of individuals from all data sources. Every time an individual survivor record is generated, SAS MDM must push the generated survivor record to the Call Center US data source. The Call Center US data source in the SAS MDM sample is represented by a table named CC_US_INDIVIDUAL in the SAS MDM database, and is created during the INDIVIDUAL entity type record harmonization process. In a real business scenario, this can be a table in a remote database, a text file, or any other form of data output. In the **Source System Harmonization** setting for the INDIVIDUAL entity type, you can see that Call Center US is specified as selected source system to which to harmonize data when the hub sends an update. The sample push job for this is mdm_push_individual.ddf. When you load INDIVIDUAL entity type data, update existing Individual records, or create new INDIVIDUAL records from the hub, the survivors records are harmonized to the sample output table CC_US_INDIVIDUAL.

Attribute Properties Tab

When an attribute is selected in the left pane, the following fields are displayed on the **Properties** tab:

Figure 9.7 Attributes Fields

The screenshot shows the SAS MDM interface for defining attributes. On the left, a tree view lists attributes under the 'Company' entity, including 'MDM_ENTITY_TYPE_ID', 'MDM_VALID_TO_DTTM', 'MDM_VALID_FROM_DTTM', 'MDM_LANGUAGE_ID', 'MDM_SRC_SYS_REC_ID', 'MDM_SRC_SYS_ID', 'MDM_ENTITY_CLUSTER_ID', 'MDM_ENTITY_ID', 'NAME', 'ADDRESS1', 'ADDRESS2', 'CITY', 'COUNTY', 'STATE_PROVINCE', 'POSTALCODE', 'COUNTRY', 'PHONE1', 'PHONE2', 'EMAIL1', 'EMAIL2', 'WEBSITE', 'FAX', and 'ATTRIBUTE'. The 'ATTRIBUTE' attribute is selected. On the right, the 'Properties' tab is active, showing fields for 'Type' (set to 'ATTRIBUTE'), 'Group', 'Description', 'Data Type' (set to 'STRING'), 'Length' (set to '50'), 'Constraint', 'Valid From', and 'Valid To'. There are also checkboxes for 'Attribute is required', 'Attribute is read only', and 'Encrypt attribute'. Below these are 'Display Settings' for 'Display name' (set to 'ATTRIBUTE'), 'Search form order', 'Search results order', 'Details panel order', 'Cluster table order', and 'Record editor order', all set to '0'.

Attributes that share the same Group field value are visually grouped as parent and child nodes. The attributes and the higher-level groups are listed according to their respective display order values.

The toolbar over the left pane contains an **Actions** menu with the following options:

New Attribute

creates a new attribute.

New Flux Attribute

creates a new flux attribute, which contains standardized or match code values, based on the attribute selected in the attributes pane. A flux attribute is always associated with another attribute. SAS MDM assigns a default name for a flux attribute by adding a 'DF_' prefix and a '_MC' (match code) suffix to the parent attribute name. For flux attributes containing standardized values, change the default '_MC' suffix to a more meaningful suffix such as '_STND'. See [“Enabling Data Quality” on page 16](#) for more information.

Note: Flux attributes cannot be added to inherited attribute types.

Retire Attribute Type

retires the selected attribute. An attribute cannot be retired if it is used in either a clustering type match condition or a relationship type match condition.

Note: For information about considerations when retiring required attributes, see the discussion of the MDM_REQUIRED modifier in [“Attribute Modifiers” on page 29](#).

The attribute fields on the **Properties** tab are as follows:

Note: If the attribute is inherited from another entity type, you can change only the display properties. Other values must be changed in the context where the attribute is defined.

Type [required]
is the attribute type.

Description
is the attribute description.

Group
groups similar attributes together in the entity editor. The groups are the collapsible and expandable field sets in the entity editor.

Note: If you enter a new Group value (one that is not already known to SAS MDM), and you subsequently want to use a different locale for a Master Data Management session, you must log on specifying the different locale and create the Group value in the language of that locale. If you enter an existing Group value for which the corresponding value for a different locale already exists, the value of the other locale is used without requiring further modifications.

Data Type
maps the attribute to a given data type: STRING, TEXT, LIST, DATE, TIMESTAMP, BOOLEAN, INTEGER, REAL, or URL. The data type determines how the attribute data is stored and accessed in the database. By default, all newly defined attributes are set to the STRING data type.

Length
specifies the character length of certain types of data, depending on the type selected in the **Data Type** field. Fields with LIST, STRING, or TEXT data types require a default length with a minimum length of 1. Fields with other data types ignore this length.

Constraint
indicates whether the attribute has a regular expression constraint that requires the incoming attribute data to match the defined regular expression. If the attribute data does not match this constraint, the data is marked as invalid. Typical attributes that have constraints include phone numbers, addresses, and e-mail addresses. Fields with a LIST data type have built-in constraints that limit specific characters (for example, M, F, and U for Gender). Semicolons (;) act as delimiters. Other constraints specify the allowable format for input characters as regular expressions. For more information about regular expressions, see the *DataFlux Expression Language Reference Guide for Data Management Studio*.

Valid From [read-only]
indicates the date from which the attribute is active. This date is automatically set when a new attribute is created, and is automatically updated when the attribute is saved.

Valid To [read-only]
indicates the date on which the attribute is inactive or retired. This date is automatically updated when the attribute is retired.

Attribute is required
selecting this option indicates that the attribute is required in the entity editor.

Attribute is read only
selecting this option indicates that the attribute is not editable in the entity editor. Read-only fields and labels appear dimmed. In most cases, except the select menus, the text inputs are shown as static text. The select menus are shown as disabled menus.

Select Groups

opens a dialog box from which you can select groups to have permissions on the selected attribute or entity.

Remove Groups

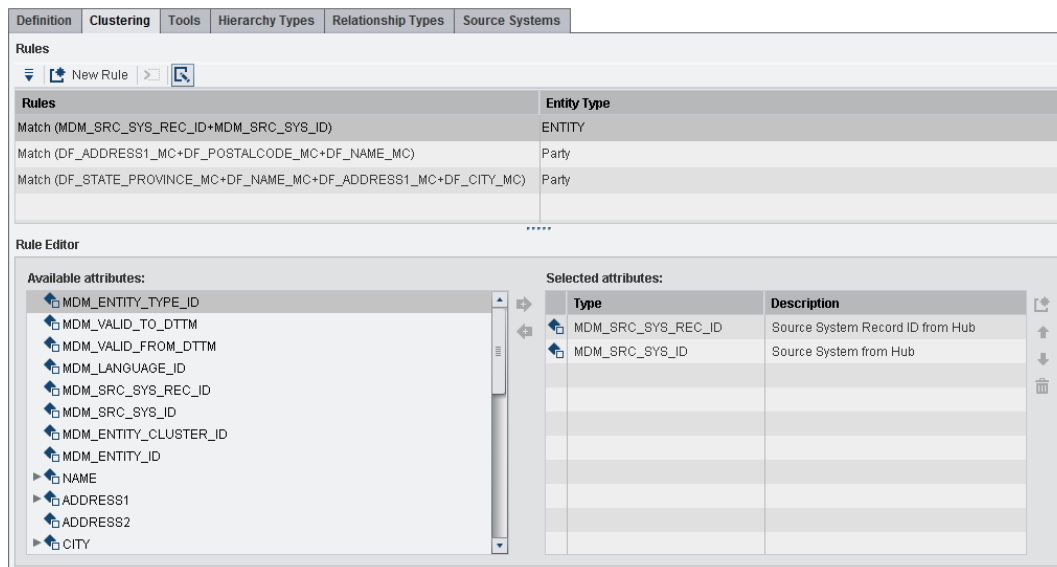
removes groups that have been selected in the groups list.

Using the Clustering Editor

Overview

The clustering editor is used to create and modify cluster conditions for an entity type. A rule specifies certain attributes for the cluster condition. Records that match the attributes specified in the rule are clustered together. To display the clustering editor, click the **Clustering** tab in the entity type editor.

Figure 9.9 Clustering Editor



The toolbar on the **Clustering** tab contains an **Actions** menu with the following options:

New Rule

creates a new rule with no attributes associated with it.

Retire

retires the selected rule from the list of rules associated with the entity type.

Show Rule Editor

displays or hides the **Available** and **Selected** panes.

When you use the **Generate Jobs** option from the entity type editor, the cluster conditions that you define in the clustering editor are automatically added to the jobs that require this information.

Rules Panes

The Rules pane of the editor lists the cluster conditions that have been defined for an entity type. Select a condition in the list to load it into the Rules Editor

panes where you can add attributes to and remove attributes from the condition. To move attributes between the Available and Selected lists, either double-click the attribute or click the arrow icons.

The condition list includes conditions that are inherited by the entity type (for example, the source system ID and source system record ID condition from ENTITY). However, inherited conditions cannot be modified in the clustering editor. To update an inherited condition, you must open an entity type editor on the entity type where the condition is defined.

It is possible that the data for a clustering rule does not always appear in the same field. For example, the entity type defines multiple e-mail addresses and records should match regardless of the field where the e-mail address appears. In this situation, you can create rules that match by specifying attribute groups. Each group can match within itself or with other attribute groups within the same rule. For example, to match company records based on the name field and

either their phone or fax number, click  to create a group like this:

Figure 9.10 New Group

Selected attributes:		
	Type	Description
	NAME	Full Name from Src Sys
	DF_FAX_MC	Fax Number Match Code from Hub
	---Group Separator---	
	NAME	Full Name from Src Sys
	DF_PHONE1_MC	Primary Phone Number Match Code from Hub

Note that both groups have the same number of attributes and that the fields that we want to compare are in the same relative position within the groups.

In the rule editor, rules that use cross-field matching are presented like this, where the comma delimits each group within the rule:

Figure 9.11 Cross-Field Matching

Match or cross match (NAME+DF_FAX_MC,NAME+DF_PHONE1_MC)

Using the Tools Editor

Overview

The tools editor is used to create and modify tools that run from the SAS MDM interface. Some uses of the tools editor are as follows:

- initiating a query for transactional data in external systems
- starting a batch process to load data or generate a report
- creating saved queries
- validating address data when editing records

- making changes to a group of records at the same time
- showing a selected address on a Google map
- performing a web search on selected data
- sending several rows of data via e-mail to a colleague for review

To define a new tool, select an entity, and then choose a data job, process job, or real-time service from the DataFlux Data Management Server. Indicate the output type of the tool and the number of rows to be returned. This defines the locations from which the tool is available. SAS MDM attempts to align entity attributes with tool input parameters. If they do not align, the user is prompted at run time to supply data values for the unsupplied attributes. Once saved on the **Tools** tab, the tool becomes available in SAS MDM wherever entity data tables or entity form editors are used. Output can be returned to existing fields in a form, new data tables, a message, or a new browser tab (if a URL is the tool output). SAS MDM does not validate the logic of the job.

To display the tools editor, click the **Tools** tab in the entity type editor.

Figure 9.12 Tools Editor Properties

The screenshot shows the 'Tools' tab in the SAS MDM interface. On the left, a list of tools is displayed, with 'Company Email Process' selected. The main area shows the 'Properties' tab for this tool. The properties include:

- Display name:** * Company Email Process
- Description:** Creates and sends email messages
- Inheritance:** Company
- Job/service:** * sasmdm/tool_company_email_process.djf
- Job/service type:** Process Service
- # Input rows:** 0-1
- Result:** Status Only
- Sort Order:** * 10
- Contact person:**
- Valid From:** 12/10/13 10:10:47 PM
- Valid To:**

The tools editor includes a pane on the left, under the toolbar, that displays the tool list.

Note: The contents of the list vary depending the type of entity and what your administrator has made available.

On the right are two additional tabs, **Properties** and **Permissions**. The **Properties** tab displays the properties of the tool selected in the left pane, and the **Permissions** tab displays the permissions on the tool.

The toolbar on the **Tools** tab contains an **Actions** menu with the following options:

New Tool

creates a new tool type.

Retire

removes tools that have been selected in the tools list.

By default, the tools editor opens to the **Properties** tab.

Properties Tab

The toolbar on the **Properties** tab contains an **Actions** menu with the following options:

Inputs and Outputs

Opens a window that displays input and output fields. This window is available after you have selected a job or service. The window can be used to see the mapping of entity attributes to the service parameters.

Show MDM User Prompt - With Input

Opens a window that displays the prompt that the user sees when running the tool with inputs. This window is available after you have selected a job or service.

Show MDM User Prompt - With No Inputs

Opens a window that displays the prompt that the user sees when running the tool without inputs. For example, if a user does not select a row of data but still invokes the tool, this prompt is displayed. This window is available after you have selected a job or service.

When a tool is selected in the left pane, the **Properties** tab displays the following fields:

Display Name [required]
the tool display name.

Description
a description of the tool.

Inheritance [read-only]
the parent entity type.

Jobs/service [required]
the job or service to run for the tool. Click  to open a dialog box and select a job or service.

Jobs/service type [read-only]
displays the type of job or service selected, such as data service, process service, or batch job.

Input rows
the number of rows of input data for the tool to use. Batch jobs and Process Services support only 0-1 input rows.

Result
the type of result to be returned by the tool. Select one of the following options from the drop-down list:

- Attribute Values (Data Services). Attribute Value tools can modify fields in an Entity editor form. Modified fields are highlighted.
- Table (Data Services). Table tools return tabular information that is presented in a new tab.

Note: If you want to enable Open Entity functionality on the results table that contains entity information, include the MDM_ENTITY_TYPE_ID field and the MDM_ENTITY_CLUSTER_ID field in the output from your job or service.

- URL (Process Services, Data Services). URL tools return a URL that is opened in a new browser tab or window. You might need to enable pop-ups to see the results of URL tools.

- **Status Only** (Process Services, Data Services). Status Only tools return a status message that is presented as a message pop-up within the user interface.
- **Job start notification only** (Batch Jobs). Job start notifications are status messages returned by batch jobs. These notifications indicate that the batch tool has started successfully. Completion messages might be generated by the tool, but no further updates are available within the user interface.

The available options depend on whether the tool is a data service, process service, or batch job.

Note: Be aware of the following:

- Tools that return URLs require the browser to allow pop-ups.
- Attribute value tools do not highlight the background color for updated fields for drop-down lists and timestamp fields.

Sort Order [required]

the sort order of the tool in selection lists. To hide the tool in selection lists, enter either zero (0) or a negative value.

Contact person

person to contact regarding the tool.

Valid From [read-only]

the date from which the tool is active. This date is automatically set when a new tool is created, and is automatically updated when the tool is saved.

Valid To [read-only]

the date on which the tool is inactive or retired. This date is automatically updated when the tool is retired.

Permissions Tab

The **Permissions** tab displays the permissions on the tool selected in the left pane. Permissions determine which groups of users can access the tool. General permissions for entity types apply. Therefore, if a group has access to a tool but does not have access to the entity on which the tool is defined, the group cannot access the tool.

type editor **Source Systems** tab lists source systems from which the contributor records originate for the entity type in the hub. The **Source Systems** table on the main **Master Data Management** ► **Administration** tab lists source systems from which the contributor records originate for all entities in the hub. See [“Working with the Source Systems Editor” on page 129](#) for more information about the source systems editor.

Understanding Relationship Type Definitions

Relationship type definitions describe the types of relationships that can exist between the associated entity types. Relationship type definitions specify source and target entity types and have optional match conditions that can be used to determine whether two entities are related. For step-by-step examples that demonstrate how relationship types are established, see [Chapter 12, “Working with SAS MDM,” on page 147](#).

Relationship types can also be enhanced with relationship type attributes.

Working with the Relationship Type Editor

Overview

The relationship type editor is used to create a new relationship type definition, to edit an existing definition, or to retire a definition. It is also used to create and new relationship type attribute or retire an attribute.

Note:

- A retired item becomes inactive, but is not deleted from the hub. It can be viewed as read-only, but cannot be edited.
- You cannot retire a relationship type that is part of a hierarchy type. You must first disassociate the relationship type from the hierarchy type. See [“Understanding Hierarchy Type Definitions” on page 120](#) for more information.

The relationship type editor is accessed from a **Relationship Types** table:

Figure 9.14 Relationship Types Table

New Relationship Type						
Type	Display Name	Description	Source Entity Type	Target Entity Type	Valid From	Valid To
MANUFACTURER	Manufactures	Company manufactures Part	Company	Part	06/30/13 07:50:3...	
PURCHASER	Purchased	Party purchases Part	Party	Part	06/30/13 07:50:3...	
EMPLOYS	Employs	Company employs Individual	Company	Individual	06/30/13 07:50:3...	

Note: Relationship type definition and relationship type are equivalent terms.

The toolbar at the top of the **Relationship Types** table contains an **Actions** menu with the following options:

New Relationship Type

creates a new relationship type.

Open

opens the existing relationship type selected in the table.

Retire

retires the relationship type selected in the table.

Show Retired

displays retired relationship types in the table.

Generate Jobs

generates default batch jobs and data services in support of the relationship type. The generation process uses the metadata that has been saved in the hub and publishes .djf and .ddf files to your DataFlux Data Management Server. Changes to the relationship type metadata definition are not fully functional until the jobs for the type and any derived types have been regenerated.

Publish

makes the relationship type active. This includes generating the tables used to store the data in your DBMS, as well as updating the metadata that the Master Data Management references when determining which entity types to display.

Export

exports the **Relationship Types** table in either CSV or PDF format.

Note: Not all characters can be represented in the fonts used in PDF export files. When a character cannot be represented in the PDF file, it is replaced with #.

Selecting **New Relationship Type** or **Open** from the **Actions** menu opens the relationship type editor on a new tab. For example, opening Purchaser displays the following:

Figure 9.15 Relationship Types Editor

The screenshot shows the 'Relationship Types Editor' window. At the top is a toolbar with icons for Save, Settings, and Help. Below the toolbar are two tabs: 'Definition' (selected) and 'Rules'. The 'Definition' tab is divided into a left sidebar and a main content area. The sidebar shows a tree view with 'EMPLOYS' selected, which has sub-items 'EMP_STATUS' and 'START_DATE'. The main content area is titled 'Relationship Type Properties' and contains two sections: 'Relationship Type Properties' and 'Display Settings'. The 'Relationship Type Properties' section includes fields for Type (EMPLOYS), Description (Company employs Individual), Source Entity Type (COMPANY), Target Entity Type (INDIVIDUAL), Last Modified (01/26/15 09:55:46 AM), Valid From (01/26/15 09:54:39 AM), Last Published (01/26/15 09:54:39 AM), and Valid To. The 'Display Settings' section includes fields for Display name (Employs), Inverse Display Name (Employed By), Sort Order (30), and Inverse Sort Order (35).

The toolbar above the **Definition** and **Rules** tabs contains an **Actions** menu with the following options:

Save

saves the relationship type.

Retire

retires the relationship type.

Generate Jobs

generates default batch jobs and data services in support of the relationship type. The generation process uses the metadata that has been saved in the hub and publishes .djf and .ddf files to your DataFlux Data Management Server. Changes to the relationship type metadata definition are not fully functional until the jobs for the type and any derived types have been regenerated.

Publish

makes the relationship type active. This includes generating the tables used to store the data in your DBMS, as well as updating the metadata that the Master Data Management references when determining which entity types to display.

Reset

discards changes and resets the relationship type editor to its original state.

Relationship Type Editor Tabs

The relationship type editor contains the following tabs:

Definition

enables you to view and modify properties for a relationship type or relationship type attribute.

Rules

enables you to create rules that define match conditions for a relationship type.

Using the Definition Editor

Overview

The definition editor is used to view and modify properties of relationship types or attributes. To display the definition editor, click the **Definition** tab in the relationship type editor.

Relationship Type Fields

When the relationship type is selected in the left pane, the following fields are displayed:

Figure 9.16 Relationship Type Fields

The screenshot shows a software interface with two tabs: 'Definition' and 'Rules'. The 'Definition' tab is active. On the left, there is a tree view with a folder icon and a plus sign, containing the item 'NEW_RELATIONSHIP_TYPE'. To the right of the tree is a 'New Attribute' button. The main area displays the 'Relationship Type Properties' and 'Display Settings' for the selected item.

Relationship Type Properties

- Type: * NEW_RELATIONSHIP_TYPE
- Description:
- Source Entity Type: * PARTY
- Target Entity Type: * PARTY
- Last Modified:
- Valid From:
- Last Published:
- Valid To:

Display Settings

- Display name: * NEW_RELATIONSHIP_TYPE
- Inverse Display Name: * NEW_RELATIONSHIP_TYPE
- Sort Order: * 30
- Inverse Sort Order: * 30

Type [required]

the relationship type name. Relationship type names must be unique.

Description

the relationship type description.

Source Entity Type[required]

the source entity type of the relationship. Changing the source entity type removes any match conditions that have been defined for the relationship type.

Target Entity Type[required]

the target entity type of the relationship.

Valid From [read-only]

the date from which the relationship type is active. This date is automatically set when a new relationship type is created and is automatically updated when the relationship type is saved.

Valid To [read-only]

the date on which the relationship type is inactive or retired. This date is automatically updated when the relationship type is retired.

Display Name [required]

the display name of the relationship type, listed under the Label column in the **Relationship Types** table.

Inverse Display Name [required]

the display name used by the target entity type when presenting the relationship.

Sort Order [required]

the form display order of the relationship type in the entity editor for the source entity type.

Inverse Sort Order [required]

the form display order of the relationship type in the entity editor for the source target type.

Note: The **Sort Order** and **Inverse Sort Order** fields control whether the relationship type appears in the entity editor. To prevent the relationship type from appearing in the entity editor, enter either zero (0) or a negative value in these fields. To show the relationship type in the entity editor, enter a value greater than zero. Include sufficient padding to allow for future relationship types to be added or for relationship types to be reorganized.

Attribute Fields

For each relationship type that you create, you can define any number of relationship attributes. Relationship type attributes are optional.

The toolbar on the **Definition** tab contains an **Actions** menu with the following options:

New Attribute

creates a new attribute.

Retire Attribute Type

retires the selected attribute.

Figure 9.17 Attribute Fields

The screenshot shows the Relationship Type Editor interface. On the left, a tree view shows 'NEW_RELATIONSHIP_TYPE' expanded, with 'ATTRIBUTE' selected. The main area is divided into two sections: 'Attribute Type Properties' and 'Display Settings'.

Attribute Type Properties:

- Type: * ATTRIBUTE
- Group:
- Description:
- Data Type: STRING
- Length: * 50
- Constraint:
- Valid From:
- Valid To:
- ☐ Attribute is read only
- ☐ Encrypt attribute

Display Settings:

- Display name: * ATTRIBUTE
- Details panel order: * 0
- Record editor order: * 0

When the relationship type attribute is selected in the left pane, the following fields are displayed:

Type [required]

the attribute name.

Description

the attribute description.

Group

groups similar attributes together in the relationship type editor. The groups are the collapsible and expandable field sets in the relationship type editor.

Note: If you enter a new Group value (one that is not already known to SAS MDM), and you subsequently want to use a different locale for a Master Data Management session, you must log on and specify the different locale and create the Group value in the language of that locale. If you enter an existing

Group value for which the corresponding value for a different locale already exists, the value of the other locale is used without requiring further modifications.

Data Type

maps the attribute to a given data type: STRING, TEXT, LIST, DATE, TIMESTAMP, BOOLEAN, INTEGER, REAL, or URL. The data type determines how the attribute data is stored and accessed in the database. By default, all newly defined attributes are set to the STRING data type.

Length

specifies the character length of certain types of data, depending on the type selected in the **Data Type** field. Fields with LIST, STRING, or TEXT data types require a default length with a minimum length of 1. Fields with other data types ignore this length.

Constraint

indicates whether the attribute has a regular expression constraint that requires the incoming attribute data to match the defined regular expression. If the attribute data does not match this constraint, the data is marked as invalid. Typical attributes that have constraints include phone numbers, addresses, and e-mail addresses. Fields with a LIST data type have built-in constraints that limit specific characters (for example, M, F, and U for Gender). Semicolons (;) act as delimiters. Other constraints specify the allowable format for input characters as regular expressions. For more information about regular expressions, see the *DataFlux Expression Language Reference Guide for Data Management Studio*.

Valid From [read-only]

indicates the date from which the attribute is active. This date is automatically set when a new attribute is created, and is automatically updated when the attribute is saved.

Valid To [read-only]

indicates the date on which the attribute is inactive or retired. This date is automatically updated when the attribute is retired.

Attribute is read only

selecting this option indicates that the attribute is not editable in the entity editor. Read-only fields and labels appear dimmed. In most cases, except the select menus, the text inputs are shown as static text. The select menus are shown as disabled menus.

Encrypt Attribute

selecting this option indicates that the attribute is encrypted in the database. This field can be modified only if the attribute type or the entity type has not been published. After the attribute or entity type is published, this field becomes a read-only field.

Display Name [required]

the attribute display name.

Details panel order [required]

the display order of the attribute in the details panel.

Record editor order [required]

the specific column order of the attribute in the record editor.

Note: Relationship attributes cannot be inherited.

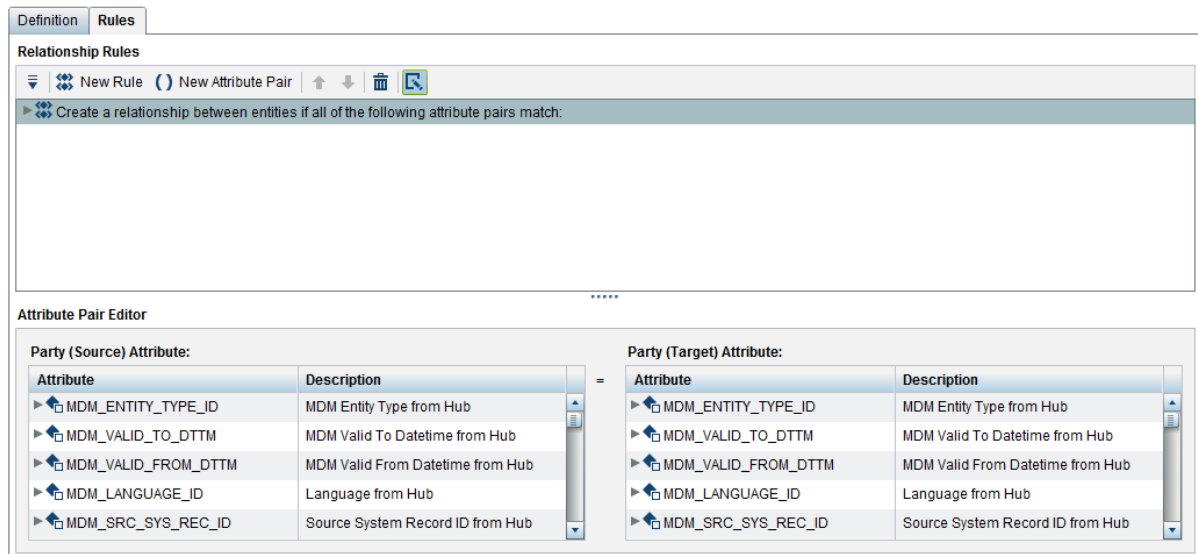
Using the Rules Editor

Overview

The rules editor is used to create rules that define match conditions for a relationship type. You can identify the attributes in the source and target entity types that are used to relate instances of their entity types automatically. All the attributes matches in a rule are used when determining if two entities are related.

The properties editor is used to view and modify properties of the relationship type. To display the rules editor, click the **Rules** tab in the relationship type editor.

Figure 9.18 Relationship Type Editor Rules Tab



The toolbar on the **Rules** tab contains an **Actions** menu with the following options:

New Rule

creates a new rule folder in the **Rules** pane.

New Attribute Pair

creates an empty attribute pair to which you add source and target attributes.

Move Up

moves the selected attribute pair up to a different rule folder.

Move down

moves the selected attribute pair down to a different rule folder.

Delete

deletes the selected item in the **Rules** pane, either a rule folder or an attribute pair.

Show Attribute Pair Editor

displays or hides the Attribute Pair Editor.

Creating and Deleting Rules

To create a rule with attribute pairs:

- 1 Select **New Rule** from the **Actions** menu. A new rule folder appears in the **Rules** pane.
- 2 Select **New Attribute Pair** from the **Actions** menu. A new, empty attribute pair appears in the selected rule folder.
- 3 Select a source attribute and a target attribute in the **Attribute Pair Editor** to add to the selected empty attribute pair.
- 4 Continue to add attribute pairs. To change the folder location of attribute pairs, select the attribute pair. Then, select **Move Up** or **Move Down** from the **Actions** menu.

To delete a rule or individual attribute pairs:

- 1 Select the rule or attribute pair that you want to delete.
- 2 Select **Delete** from the **Actions** menu.

Note: Be aware of the following requirements:

- When creating match conditions, the source and target attributes must have comparable data types. For example, an Integer attribute cannot be compared against a String attribute.
- The source and target attributes must be unique within each group. If redundant rules exist within a group, they are automatically merged when you save the entity type.

Understanding Hierarchy Type Definitions

Hierarchy type definitions enable you to define hierarchy types that are used to build hierarchies of entities. Hierarchy types are built on relationship types that you have already defined. Only defined relationships appear in the hierarchy type editor. The relationship types used to build the hierarchy type determine the types of entities that can exist within a hierarchy and the structure in which they appear.

Consider hierarchy types as defined views of relationship types. For example, if you define relationship types that show the connection between a supply company, the parts that it sells, and the people that it employs, you can create a hierarchy type that shows only the supplier-to-part relationship in a hierarchical view. This view does not show the people employed by the supplier even though the supplier-employee relationship exists.

Some constraints and defined behaviors exist for hierarchy types:

- Only one relationship type can be specified for each hierarchy level.
- Entities that are related to each other either through the data elements defined for the relationship type or through explicit linking of the entities in the Master Data Management application are reflected in hierarchies. If you relate new entities, those changes are reflected in hierarchies in which those relationships are defined as levels.

- You cannot retire an entity type or relationship type that is part of a hierarchy type. You must first disassociate the entity type or relationship type from the hierarchy type.

Working with the Hierarchy Type Editor

Overview

The hierarchy type editor is used to create a new hierarchy type definition, to edit an existing definition, or to retire a definition.

Note: A retired item becomes inactive, but is not deleted from the hub. It can be viewed as read-only, but cannot be edited.

The hierarchy type editor is accessed from a **Hierarchy Types** table:

Figure 9.19 Hierarchy Types Table

Hierarchy Types

 New Hierarchy Type
 

Type	Display Name	Description	Valid From	Valid To
 MANUFACTURER TO PURCHASER	Manufacturer to Purchaser	Displays Manufacturer to Purchaser tr...	12/04/13 02:37:05 PM	

Note: Hierarchy type definition and hierarchy type are equivalent terms.

The toolbar at the top of the **Hierarchy Types** table contains an **Actions** menu with the following options:

New Hierarchy Type

creates a new hierarchy type.

Open

opens the existing hierarchy type selected in the table.

Retire

retires the hierarchy type selected in the table.

Show Retired

displays retired hierarchy types in the table.

Selecting **New Hierarchy Type** or **Open** from the **Actions** menu opens the hierarchy type editor on a new tab. For example, opening Manufacturer to Purchaser displays the following:

Figure 9.20 Hierarchy Types Editor

The screenshot shows the 'Properties' tab of the Hierarchy Types Editor. At the top, there is a toolbar with a dropdown arrow, a 'Save' button, and a circular arrow icon. Below the toolbar, the 'Properties' tab is active, showing the following fields:

- Type: * MANUFACTURER TO PURCHASER
- Display name: * Manufacturer to Purchaser
- Description: Displays Manufacturer to Purchaser
- Valid From: 12/03/13 09:35:24 PM
- Valid To:

The toolbar above the **Properties** tab contains an **Actions** menu with the following options:

Save

saves the hierarchy type.

Retire

retires the hierarchy type.

Reset

discards changes and resets the hierarchy type editor to its original state.

Hierarchy Type Editor Tabs

The hierarchy type editor contains the following tabs:

Properties

enables you to view and modify properties for a hierarchy type.

Levels

enables you to add or delete hierarchy levels.

Using the Properties Editor

The properties editor is used to view and modify properties of the hierarchy type. To display the properties editor, click the **Properties** tab in the hierarchy type editor.

Figure 9.21 Relationship Type Editor Properties Tab

Properties		Levels
Type:	*	MANUFACTURER TO PURCHASER
Display name:	*	Manufacturer to Purchaser
Description:		Displays Manufacturer to Purchaser
Valid From:		12/03/13 09:35:24 PM
Valid To:		

The properties editor contains the following fields:

Type [required]

the hierarchy type name. Hierarchy type names must be unique.

Display Name [required]

the display name of the hierarchy type.

Description

a description of the hierarchy type.

Valid From [read-only]

the date from which the hierarchy type is active. This date is automatically set when a new hierarchy type is created and is automatically updated when the hierarchy type is saved.

Valid To [read-only]

the date on which the hierarchy type is inactive or retired. This date is automatically updated when the hierarchy type is retired.

Using the Levels Editor

The levels editor enables you to add levels to a hierarchy. To display the levels editor, click the **Levels** tab in the hierarchy type editor.

Figure 9.23 New Level Dialog Box



The dialog box is titled "New Level" and has a close button (X) in the top right corner. It contains a "Location:" label followed by a dropdown menu currently showing "Below current selection". Below this is a table with three columns: "Source", "Relationship", and "Target". The first row of the table is highlighted in blue and contains the values "Party", "Purchased", and "Part". There are 15 rows in total, with the first row being the header and the remaining 14 rows being empty. At the bottom right of the dialog box are two buttons: "OK" and "Cancel".

Source	Relationship	Target
Party	Purchased	Part

- 3 Select the relationship.
- 4 Select **Below current selection** or **Above current selection** from the drop-down list.
- 5 Click **OK**.

10

Using Master Data Management Administration

About Master Data Management Administration	127
Overview	127
Managing Sources Systems and Languages	128
Working with the Source Systems Editor	129
Overview	129
Source Systems Editor Tabs	131
Using the Definition Editor	131
Using the Tools Editor	131
Working with the Language Editor	135
Overview	135
Using the Language Editor	137

About Master Data Management Administration

Overview

From the **Administration** tab, data administrators configure source systems and languages that are used in the hub and by Master Data Management.

Note: The **Administration** tab is available only to users who have View Application, Manage Source Systems, and Manage Languages capabilities.

To access the **Administration** tab, click the **Administration** link under **Master Data Management** on the SAS Data Management Console page. The following display appears:

Figure 10.1 Administration Tab

Source Systems

New Source System

Display Name	Description	Valid From	Valid To
Best Record	Best records constructed by the Hub	06/07/13 04:18:37 PM	
MD Manager	Master Data Manager	06/07/13 04:18:37 PM	
ERP	ERP (US)	06/07/13 04:19:56 PM	
Marketing	Marketing Database (US)	06/07/13 04:19:56 PM	
Orion	Orion	06/07/13 04:19:56 PM	
Call Center UK	Call Center (UK)	06/07/13 04:19:56 PM	
Call Center US	Call Center (US)	06/07/13 04:19:56 PM	
CRM	CRM (US)	06/07/13 04:19:56 PM	

.....

Languages

New Language

Display Name	ISO Locale	QKB Locale	Default	Valid From	Valid To
English (US)	USA	ENUSA	Y	06/07/13 04:19:00 PM	

Note: In the following descriptions, **Actions** menu items are accessed by clicking from the toolbar. Most functions available from the **Actions** menu are also available from icons on the same toolbar. Clicking the icon performs the function.

Managing Sources Systems and Languages

Here are the items managed from the **Administration** tab:

source systems

source systems from which the contributor records in the hub originate.

languages

languages that SAS MDM supports and their associated QKBs and ISO locales.

Each table contains several entries. Selecting an entry under either of the tables and clicking **Open** from its **Actions** menu opens an editor on a separate tab. For example, selecting **English (US)** under **Languages** opens a language editor:

Figure 10.2 Language Editor

Display Name: * English (US)

MDM QKB Locale: * ENUSA

ISO Locale: * USA

☒ Default

Valid From: 06/21/13 10:11:14 PM

Valid To:

The individual components of the editors are described in the following sections.

Working with the Source Systems Editor

Overview

The source systems editor is used to create a new source system definition, to edit an existing definition, or to retire a definition. The source system is used to specify from which system entity data and cluster information originally came.

Note: A retired item becomes inactive, but is not deleted from the hub. It can be viewed as read-only, but cannot be edited.

The source systems editor is accessed from the **Source Systems** table.

Figure 10.3 Source Systems Table

Source Systems

New Source System

Display Name	Description	Valid From	Valid To
Best Record	Best records constructed by the Hub	06/07/13 04:18:37 PM	
MD Manager	Master Data Manager	06/07/13 04:18:37 PM	
ERP	ERP (US)	06/07/13 04:19:56 PM	
Marketing	Marketing Database (US)	06/07/13 04:19:56 PM	
Orion	Orion	06/07/13 04:19:56 PM	
Call Center UK	Call Center (UK)	06/07/13 04:19:56 PM	
Call Center US	Call Center (US)	06/07/13 04:19:56 PM	
CRM	CRM (US)	06/07/13 04:19:56 PM	

Note: The source systems shown in the image are samples provided with SAS MDM and might not match the definitions in your hub.

The following sample source systems are included with SAS MDM and are required for correct operation:

Best Record

a source system representing the hub. All survivor records appear in this source system.

MD Manager

a source system representing Master Data Management. All records created by Master Data Management appear in this source system.

The toolbar at the top of the **Source Systems** table contains an **Actions** menu with the following options:

New Source System

creates a source system definition.

Open

opens the existing source system definition selected in the table.

Retire

retires the source system definition selected in the table.

Show Retired

displays retired source system definitions in the table.

Export

exports the **Source Systems** table in either CSV or PDF format.

Note: Not all characters can be represented in the fonts used in PDF export files. When a character cannot be represented in the PDF file, it is replaced with #.

Selecting **New Source System** or **Open** from the **Actions** menu opens the source system editor on a new tab. For example, opening the ERP definition displays the following editor:

Figure 10.4 Source Systems Editor

Definition Tools

Display name: * ERP

Description: ERP (US)

Contact person:

Path:

Valid From: 10/29/14 11:07:30 AM

Valid To:

Select the entity types that this source system contributes records to:

	Entity Type	Display Name	Description
☒ +	PART	Part	Part
☒ +	INDIVIDUAL	Individual	Individual
☒ +	COMPANY	Company	Company
☐ +			
☐ +			
☐ +			
☐ +			
☐ +			

The toolbar at the top of the tab contains an **Actions** menu the following options:

Save

saves the source system definition.

Retire

retires the source system definition.

Reset

discards changes and resets the source systems editor to its original state.

Source Systems Editor Tabs

The source systems editor contains the following tabs:

Definition

enables you to view and modify the definition of a hierarchy type.

Tools

enables you to create and modify tools.

Using the Definition Editor

The definition editor contains the following fields:

Display Name [required]

the source system definition display name, listed under the Name column in the **Source Systems** table.

Description

the source system definition description.

Contact Person

the person to be notified about issues with the source system.

Path [read-only]

path is applicable only for source system data that is migrated from DataFlux Master Data Management Foundation. It is used to correlate the source system with its representation in the Master Data Management Foundation hub, where it is referred to as a source table.

Valid From [read-only]

the date from which the source system definition is active. This date is automatically set when a new source system definition is created and is automatically updated when the language definition is saved.

Valid To [read-only]

the date on which the source system definition is inactive or retired. This date is automatically updated when the source system definition is retired.

The table at the bottom of the editor enables you to select to which entity types the sources system contributes records. Entity types selected here determine tool availability on the **Tools** tab.

Using the Tools Editor

Overview

The tools editor is used to create and modify tools that run from the SAS MDM interface. Some uses of the tools are as follows:

- initiating a query for transactional data in external systems
- starting a batch process to load data or generate a report
- creating saved queries
- validating address data when editing records
- making changes to a group of records at the same time

- showing a selected address on a Google map
- performing a web search on selected data
- sending several rows of data via e-mail to a colleague for review

To define a new tool:

- 1 Select an entity.
- 2 Click **New Tool**.
- 3 Enter a display name.
- 4 Choose a data job, process job, or real-time service available from DataFlux Data Management Server.
- 5 Indicate the output type of the tool.
- 6 Set the sort order for the tool.

SAS MDM attempts to align entity attributes with tool input parameters. If they do not align, you are prompted at run time to supply data values for the attributes. Once saved on the **Tools** tab, the tool becomes available in SAS MDM wherever entity data tables or entity form editors are used. Output can be returned to existing fields in a form, new data tables, a message, or a new browser tab (if a URL is the tool output). SAS MDM does not validate the logic of the job.

To display the tools editor, click the **Tools** tab in the source system editor.

Figure 10.5 Tools Editor Properties

The screenshot shows the SAS MDM Tools Editor interface. On the left, a tree view lists tools for three entities: Part, Individual, and Company. Each entity has four tool types: Remaining Records, Extract, Standardize, and Add/Update. The 'Add/Update' tool for the 'Part' entity is selected. The main pane on the right shows the 'Properties' tab for this tool. The fields are as follows:

Display name:	* Add/Update
Description:	Extract, standardize, and prepare part data from the source system to MDM hub
Job/service:	* sasmdm/add_update_part.djf
Job/service type:	Batch Job
Result:	Job start notification only
Sort Order:	* 1
Contact person:	
Valid From:	10/29/14 11:07:34 AM
Valid To:	

The tools editor includes a pane on the left, under the toolbar, displaying the tool list. On the right are two additional tabs, **Properties** and **Permissions**. The **Properties** tab displays the properties of the tool selected in the left pane, and the **Permissions** tab displays the permissions on the tool.

The toolbar on the **Tools** tab contains an **Actions** menu with the following options:

New Tool

creates a new tool type

Retire

removes tools that have been selected in the tools list.

By default, the tools editor opens to the **Properties** tab.

The Properties Tab

When a tool is selected in the left pane, the **Properties** tab displays the following fields:

Display Name [required]

the tool display name.

Description

a description of the tool.

Jobs/service [required]

the job or service to run for the tool. Click  to open a dialog box and select a job or service.

Jobs/service type [read-only]

displays the type of job or service selected, such as data service, process service, or batch job.

Result

the type of result to be returned by the tool. Select one of the following options from the drop-down list:

- **Attribute Values (Data Services).** Attribute Value tools can modify fields in an Entity editor form. Modified fields are highlighted.
- **Table (Data Services).** Table tools return tabular information that is presented in a new tab.
- **URL (Process Services, Data Services).** URL tools return a URL that is opened in a new browser tab or window. You might need to enable pop-ups to see the results of URL tools.
- **Status Only (Process Services, Data Services).** Status Only tools return a status message that is presented as a message pop-up within the user interface.
- **Job start notification only (Batch Jobs).** Job start notifications are status messages returned by batch jobs. These notifications indicate that the batch tool has started successfully. Completion messages might be generated by the tool, but no further updates are available within the user interface.

The available options depend on whether the tool is a data service, process service, or batch job.

Note: Tools that return URLs require that the browser allow pop-ups.

Sort Order [required]

the sort order of the tool in selection lists. To hide the tool in selection lists, enter either zero (0) or a negative value. The tool selection list can be viewed in the Actions table by selecting **Master Data Management** ► **Source Systems** from the Data Management Console. See [“Working with Source Systems” on page 88](#) for more information.

Contact person

person to contact regarding the tool.

Valid From [read-only]

the date from which the tool is active. This date is automatically set when a new tool is created, and is automatically updated when the tool is saved.

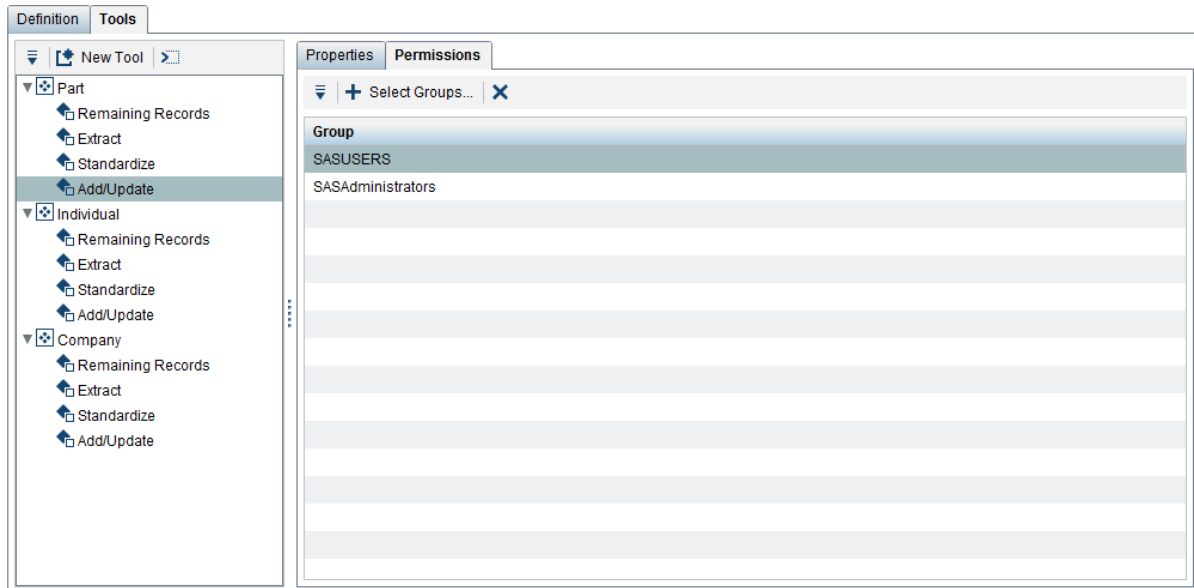
Valid To [read-only]

the date on which the tool is inactive or retired. This date is automatically updated when the tool is retired.

Permissions Tab

The **Permissions** tab displays the permissions on the tool selected in the left pane. Permissions determine which groups of users can access the tool. General permissions for entity types apply. Therefore, if a group has access to a tool but does not have access to the entity on which the tool is defined, the group cannot access the tool.

Figure 10.6 Tools Editor Permissions



For more information about groups, see [“Defining Groups” on page 46](#).

The toolbar on the **Permissions** tab contains an **Actions** menu with the following options:

Select Groups

opens a dialog box from which you can select groups and assign permissions on the selected tool.

Remove Groups

removes groups that have been selected in the groups list.

Default Source System Tools

When you select an entity type on the Source Systems **Definition** tab, SAS MDM generates default source system tools for the selected entity type. The default source system tools, displayed on the **Tools** tab, are as follows:

Standardize

an embedded tool to standardize entity type data from the source system, used internally by the Add/Update tool.

Extract

an embedded tool to extract entity type data from the source system, used internally by the Add/Update tool.

Add/Update

an executable tool to prepare, standardize, and extract data from the source system to the SAS MDM hub.

The jobs for these tools are created by SAS MDM when you generate jobs for an entity type from the entity type editor. See [“Working with the Entity Type Editor” on page 96](#) for more information about the entity type editor.

Custom Source System Tools

You can create custom source system tools. The Remaining Records tool is an example. The Remaining Records tool retrieves the remaining number of entity type records available to be loaded from a source system.

Note: The sample Remaining Records tool is provided for only the following sample source systems and entity types:

Table 10.1 Remaining Records Availability

Source Systems	Entity Types
ERP, Marketing, Orion, and Call Center UK.	Company, Individual, and Part

Working with the Language Editor

Overview

The language editor is used to create a new language definition, to edit an existing definition, or to retire a definition. You can register languages that are used to associate entity data with a given locale. Standardized values and match codes are computed based on the language associated with the entity data.

Note:

- To use the language locale features with standardizations and match codes, you must have a valid QKB license for the language that you want to use. The English (US) language is included with SAS MDM by default.
- A retired item becomes inactive, but is not deleted from the hub. It can be viewed as read-only, but cannot be edited.

The language editor is accessed from the **Languages** table.

Figure 10.7 Languages Table

Languages

≡ | + New Language | 📁

Name	ISO Locale	QKB Locale	Default	Valid From	Valid To
English (US)	USA	ENUSA	Y	05/05/13 11:07:41 AM	

The toolbar at the top of the **Languages** table contains an **Actions** menu with the following options:

New Language

creates a language definition.

Open

opens the existing language definition selected in the table.

Retire

retires the language definition selected in the table.

Show Retired

displays retired language definitions in the table.

Export

exports the **Languages** table in either CSV or PDF format.

Note: Not all characters can be represented in the fonts used in PDF export files. When a character cannot be represented in the PDF file, it is replaced with #.

About SAS MDM

displays application version information.

Selecting **New Language** or **Open** from the **Actions** menu opens the language type editor on a new tab. For example, opening the English (US) language definition displays the following editor:

Figure 10.8 Languages Editor

⌵ | 💾 Save | ↺

Display Name: * English (US)

MDM QKB Locale: * ENUSA

ISO Locale: * USA

☒ Default

Valid From: 06/21/13 10:11:14 PM

Valid To:

The toolbar at the top of the tab contains an **Actions** menu with the following options:

Save

saves the language definition.

Retire

retires the language definition.

Reset

discards changes and resets the language editor to its original state.

Using the Language Editor

The language editor contains the following fields:

Display Name [required]

the language definition display name, listed under the Name column in the **Languages** table.

MDM QKB Locale [required]

QKB locale abbreviation that matches the QKB locale. The format is [ISO 3166-1-alpha-2] [ISO 3166-1-alpha-3] (two-letter country code followed by three-letter country code). For example:

- United States = [EN][USA] = ENUSA
- Germany = [DE][DEU] = DEDEU
- France = [FR][FRA] = FRFRA

MDM ISO Locale [required]

ISO 3166-1-alpha-3 code (three-letter country code). For example:

- United States = USA
- Germany = DEU
- France = FRA

Default [required]

selecting this check box indicates that the language definition is the default used by SAS MDM. Only one default language definition can exist for the system.

Valid From [read-only]

the date from which the language definition is active. This date is automatically set when a new language definition is created and is automatically updated when the language definition is saved.

Valid To [read-only]

the date on which the language definition is inactive or retired. This date is automatically updated when the language definition is retired.

11

Working with SAS Visual Analytics

<i>SAS Visual Analytics Reports</i>	139
Overview	139
Components	139
Sub-Reports	140
<i>Creating and Viewing Reports for SAS Visual Analytics</i>	141
Starting SAS LASR Analytic Server	141
Creating the SAS MDM Report	141
Viewing the SAS MDM Report	142
Creating the SAS Data Remediation Report	142
Viewing the SAS Data Remediation Report	143
<i>Database Connections to the SAS MDM Database</i>	143

SAS Visual Analytics Reports

Overview

A single report with sub-reports is available for use with SAS Visual Analytics.

The report and metadata delivered for these SAS Visual Analytics reports is based on the sample entity types available for SAS MDM.

Note: If you attempt to load more data or allow more users than SAS guidelines specify, you are likely to see performance degradation. See *SAS Visual Analytics: Users' Guide* for information regarding SAS Visual Analytics server sizing and in-memory table size limitations.

Components

Several components are required to make the report properly viewable to end users:

HubOverview

is the SAS Visual Analytics report file. It is installed for you in your SAS metadata environment in this location: `..\Products\SAS MDM\Reports`.

data tables

the HubOverview requires several data tables from the SAS MDM database to be available to load into the memory of the SAS Visual Analytics environment. The metadata for these tables can be found in this location:

```
..\Products\SAS MDM\Data Sources\MDM LASR
```

The data tables are as follows:

- BATCH_LOAD_STATS
- MDM_ENTITY_RPT_VW

Sub-Reports

The HubOverview report contains the following sub-reports on different tabs:

metrics by entity type

displays the following counts broken down by entity type:

- row count
- active clusters
- retired clusters
- active contributors
- active survivors
- retired survivors
- retired contributors

metrics by entity type and source system

displays active and retired counts broken down by entity type and source system. Report data can be filtered using the **Entity Types** and **Source Systems** drop-down lists at the top of the report.

batch load statistics

displays time-lined and cross-tab reports of batch loads to the hub broken down by entity type and source system. Data can be filtered using range-slider and drop-down controls at the top of the report.

Metrics reported for each load are as follows:

- new contributors
- new survivors
- new clusters
- updated clusters
- collapsed clusters
- issues sent to remediation
- records sent to remediation

Creating and Viewing Reports for SAS Visual Analytics

Starting SAS LASR Analytic Server

Data Management LASR Analytic Server must be started before pushing data. To start SAS LASR Analytic Server:

- 1 Log on to SAS MDM. You must have Data Management Administrators or Data Management Stewards permissions.
- 2 From the SAS Data Management Console, click **SAS Visual Analytics** beneath LINKS to open SAS Visual Analytics.
- 3 From the SAS Visual Analytics main menu, select **Administrator**.
- 4 On the **Manage Environment** tab, select **LASR ► LASR Servers**.
- 5 Select the check box for Data Management LASR Analytic Server.
- 6 Start Data Management LASR Analytic Server.

Creating the SAS MDM Report

SAS MDM provides a HubOverview report that runs on SAS Visual Analytics. Data for the report can be pushed from the SAS MDM hub to SAS Visual Analytics using a SAS MDM tool. To create the SAS MDM report:

- 1 Log on to SAS MDM. You must have Data Management Administrators or Data Management Stewards permissions.
- 2 From SAS MDM, run the Push MDM Data To Reports tool:
 - a Open **Master Data Management** from the SAS Data Management Console.
 - b Select **Tools** from the **Actions** menu.
 - c Select and execute **Push MDM Data To Reports**.

See [“Using Tools” on page 67](#) for more information about using tools. The tool prompts you for following input:

LASR_HOST

the fully qualified host name of the SAS LASR Analytic Server.

LASR_SERVER_PORT

port on which SAS LASR Analytic Server is running. Port values for Data Management SAS LASR Analytic Server can range from 10100 to 10109.

LASR_AUTH_URL

authorization URL for SAS LASR Analytic Server. See SAS LASR Analytic Server Authorization documentation for details. The default value is similar to: `http://server:port/SASLASRAuthorization`.

DB_NAME

the TNS Name for Oracle or the ODBC connection name for SQL Server.
See “[Database Connections to the SAS MDM Database](#)” on page 143 for information about this value.

DB_SCHEMA

the database schema name.

DB_USER

the database user name.

DB_PASSWORD

the database password.

DB_TYPE

the database type, Oracle or Sql Server.

Click **OK**.

- 3 Check the SAS Data Management Server logs to confirm completion of the job.

Viewing the SAS MDM Report

To view the HubOverview report:

- 1 From the SAS Visual Analytics main menu, select **View Report**.
- 2 On the **View Report** tab, select **File ► Open**.
- 3 In the **Open** dialog box, select **SAS Folders ► Products ► SAS MDM ► Reports ► HubOverview**, and then click **Open**.

Creating the SAS Data Remediation Report

SAS MDM provides a RemediationReport report, in conjunction with SAS Data Remediation, that runs on SAS Visual Analytics. Data for the report can be pushed from the SAS Data Remediation database to SAS Visual Analytics using a SAS MDM tool. To create the SAS Data Remediation Report:

- 1 Log on to SAS MDM. You must have Data Management Administrators or Data Management Stewards permissions.
- 2 From SAS MDM, run the Push Remediation Data To Reports tool:
 - a Open **Master Data Management** from the SAS Data Management Console.
 - b Select **Tools** from the **Actions** menu.
 - c Select and execute **Push Remediation Data To Reports**.

See “[Using Tools](#)” on page 67 for more information about using tools. The tool prompts you for following input:

LASR_HOST

the fully qualified host name of the SAS LASR Analytic Server.

LASR_SERVER_PORT

port on which SAS LASR Analytic Server is running. Port values for Data Management SAS LASR Analytic Server can range from 10100 to 10109.

LASR_AUTH_URL

authorization URL for SAS LASR Analytic Server. See SAS LASR Analytic Server Authorization documentation for details. The default value is similar to: `http://server:port/SASLASRAuthorization`.

DB_HOST

the fully qualified host name for the remediation database.

DB_PORT

the port number for the Postgres connection.

DB_NAME

the remediation database name.

DB_USER

the database user name.

DB_PASSWORD

the database password.

Click **OK**.

- 3 Check the SAS Data Management Server logs to confirm completion of the job.

Viewing the SAS Data Remediation Report

To view the RemediationReport report:

- 1 From the SAS Visual Analytics main menu, select **View Report**.
- 2 On the **View Report** tab, select **File** ► **Open**.
- 3 In the **Open** dialog box, select **SAS Folders** ► **Products** ► **SAS MDM** ► **Reports** ► **RemediationReport**, and then click **Open**.

Database Connections to the SAS MDM Database

SAS MDM is installed using SAS provided ODBC drivers, which must be used with SAS MDM. For SAS Visual Analytics reports, the SAS provided ODBC drivers cannot be used when connecting to SQL Server databases containing the SAS MDM data for the reports. You must create a new data source name (DSN) using ODBC drivers supplied by your database vendor to connect to data in SAS MDM from the SAS Visual Analytics environment. For SQL Server, you need a new connection using the vendor-supplied ODBC driver. Use this `tnsname` entry or ODBC connection name as the **DB_NAME** value when you use the SAS MDM tool for creating SAS Visual Analytics HubOverview report. See [“DB_NAME” on page 142](#) for a description of this value.



Part 4

SAS MDM Examples

Chapter 12

Working with SAS MDM 147

12

Working with SAS MDM

Example 1: Creating an Abstract Entity Type	147
Overview of Creating an Abstract Entity Type	147
Open the Entity Type Editor	148
Define Entity Type Properties	148
Define Attributes	149
Define Flux Attribute	153
Add Match Code Generation to the Standardization Job	154
Define Cluster Conditions	156
Generate Jobs	157
Publish the Entity Type Definition	158
Define Relationship Types	159
Define Relationship Type Attributes	160
Generate Jobs	163
Publish the Relationship Type Definition	163
Example 2: Creating a Concrete Entity Type	164
Overview of Creating a Concrete Entity Type	164
Open the Entity Type Editor	164
Define Entity Type Properties	164
Define Attributes	165
Generate Jobs	168
Add Match Code Generation to the Standardization Job	169
Publish the Entity Type Definition	172
Example 3: Creating Entity Relationships	172
Overview of Creating Entity Relationships	172
Create Entity	172
Create Entity Relationships	175
Explore Entity Relationships	177

Example 1: Creating an Abstract Entity Type

Overview of Creating an Abstract Entity Type

The example presented in this section creates an abstract entity type named VEHICLE with VIN (vehicle identification number), MANUFACTURER, MODEL, YEAR, and PLATE attributes, as well as a DF_VIN_MC flux attribute. The example defines a single cluster condition based on the DF_VIN_MC flux attribute, and defines two relationship types: Owner and Driver.

Because VEHICLE is modeled as an abstract entity type, instances of this entity type cannot be created. Instead, concrete entity types must be created to use this metadata in the hub. The example presented in [“Example 2: Creating a Concrete Entity Type” on page 164](#) creates a concrete entity type named Car, which inherits metadata from the VEHICLE entity type.

Open the Entity Type Editor

To open the entity type editor:

- 1 Click **Data Model** on the SAS Data Management Console page. The **Data Model** tab opens.
- 2 Click **New Entity Type** on the **Entity Types** toolbar to open the entity type editor. The entity type editor opens on a new tab.

The following sections explain how to complete the fields on the entity type editor **Definition**, **Clustering**, and **Relationship Types** tabs.

Define Entity Type Properties

The definition editor is used to view and modify properties and attributes for an entity type definition.

To use the definition editor:

- 1 Click the **Definition** tab in the entity type editor to display the **Properties** tab.
- 2 Enter `VEHICLE` as the entity type in the **Type** field.
- 3 Enter `Abstract Vehicle type; extend with concrete type` in the **Description** field.
- 4 Leave the **Parent Type** field set to its default value, `ENTITY`. This value specifies the parent entity type that contributes metadata to the VEHICLE entity type.

Note: The **Valid From**, **Last Modified**, **Last Published**, and **Valid To** fields in the **Entity Type Properties** pane are blank. The **Valid From**, **Last Modified**, and **Last Published** fields are automatically completed when you save and publish the entity type definition in later steps. The **Valid To** field remains blank until the entity type definition is retired.

- 5 Select the **Entity is an abstract entity** check box. Selecting this option indicates that VEHICLE is an abstract entity type, meaning that instances of this entity type cannot be created.
- 6 Enter `vehicle` in the **Display Name** field as the label for the entity type.
- 7 Enter `1` in the **Sort Order** field. This causes the VEHICLE entity type to appear near the top of the selection lists in the Master Data Management user interface.

Figure 12.1 Definition Tab

The screenshot shows the 'Definition' tab of an entity type editor. On the left, a tree view shows the 'VEHICLE' entity type with its attributes: MDM_ENTITY_TYPE_ID, MDM_VALID_TO_DTTM, MDM_VALID_FROM_DTTM, MDM_LANGUAGE_ID, MDM_SRC_SYS_REC_ID, MDM_SRC_SYS_ID, MDM_ENTITY_CLUSTER_ID, and MDM_ENTITY_ID. The main panel is divided into two sections: 'Properties' and 'Permissions'. The 'Properties' section is further divided into 'Entity Type Properties', 'Display Settings', 'Remediation User-defined Fields', and 'Source System Harmonization'. The 'Entity Type Properties' section includes fields for Type (VEHICLE), Description (Abstract Vehicle type; extend with concrete type), Parent Type (ENTITY), Last Modified, Valid From, Last Published, and Valid To. A checkbox 'Entity is an abstract entity (cannot be instantiated)' is checked. The 'Display Settings' section includes fields for Display name (Vehicle), Sort Order (1), and Attribute to use as entity label. The 'Remediation User-defined Fields' section includes three user-defined fields. The 'Source System Harmonization' section includes a radio button 'All source systems that can contribute records to clusters of this entity type' and a list of source systems: MD Manager, ERP, Marketing, Orion, Call Center UK, and Call Center US.

8 Click **Save** on the entity type editor toolbar to save the changes.

Define Attributes

Use the following task to add five new attributes to the VEHICLE entity type: VIN, MANUFACTURER, MODEL, YEAR, and PLATE.

To define attributes:

- 1 Click **New Attribute** on the toolbar to add a new attribute. The new attribute appears in the attribute list on the left pane of the **Definition** tab.

Figure 12.2 New Attribute

Note:

- The **Length** field is not applicable for INTEGER attributes.
- The **Valid From** and **Valid To** fields are blank. The **Valid From** field is automatically completed when you save and publish the entity type definition in later steps. The **Valid To** field remains blank until the entity type definition is retired.

- 2 Enter data from the Attribute 1 Value column of the following table in the appropriate locations on the **Properties** tab.

Note:

- For check box options, **Yes** means select the check box. **No** means clear the check box.
- The **Search form order** values for the five new attributes are lower than 100, which is the lowest **Search form order** value for attributes that are inherited from the ENTITY (top-level) entity type. This causes the five new attributes to appear above the MDM Identifiers group containing the inherited attributes in the search fields on the **Master Data Management** tab.

Table 12.1 New Attributes

Attribute Option	Attribute 1 Value	Attribute 2 Value	Attribute 3 Value	Attribute 4 Value	Attribute 5 Value
Type	VIN	MANUFACTURER	MODEL	YEAR	PLATE
Group	Vehicle Information	Vehicle Information	Vehicle Information	Vehicle Information	Vehicle Information

Attribute Option	Attribute 1 Value	Attribute 2 Value	Attribute 3 Value	Attribute 4 Value	Attribute 5 Value
Description	Vehicle Identification Number	Vehicle manufacturer	Vehicle model designation	Year of manufacture	Current plate number
Data Type	String	String	String	Integer	String
Length	50	50	50	none	10
Constraint	(leave blank)	(leave blank)	(leave blank)	(leave blank)	(leave blank)
Required	No	No	No	No	No
Read-only	No	No	No	No	No
Encrypt	No	No	No	No	No
Display name	Vehicle Identification Number	Manufacturer	Model	Year	Plate
Search form order	90	92	94	96	98
Search results order	90	92	94	96	98
Details panel order	90	92	94	96	98
Cluster table order	90	92	94	96	98
Record editor order	90	92	94	96	98

Figure 12.3 VIN Attribute

The screenshot shows the SAS MDM interface with the 'Definition' tab selected. On the left, a tree view shows the hierarchy: Vehicle > MDM_ENTITY_TYPE_ID > MDM_VALID_TO_DTTM > MDM_VALID_FROM_DTTM > MDM_LANGUAGE_ID > MDM_SRC_SYS_REC_ID > MDM_SRC_SYS_ID > MDM_ENTITY_CLUSTER_ID > MDM_ENTITY_ID > VIN. The 'VIN' attribute is selected. The main panel shows the 'Properties' tab for the 'VIN' attribute. The 'Attribute Type Properties' section includes: Type: VIN, Group: Vehicle Information, Description: Vehicle Identification Number, Data Type: STRING, Length: 50, Constraint: (empty), Valid From: (empty), Valid To: (empty). There are checkboxes for 'Attribute is required', 'Attribute is read only', and 'Encrypt attribute', all of which are unchecked. The 'Display Settings' section includes: Display name: Vehicle Identification Number, Search form order: 90, Search results order: 90, Details panel order: 90, Cluster table order: 90, and Record editor order: 90.

- 3 Repeat this task to add each of the four remaining new attributes specified in the table (MANUFACTURER, MODEL, YEAR, and PLATE).
- 4 Select Vehicle at the top of the **Definition** tab, which displays the properties of Vehicle on the **Properties** tab.
- 5 Select VIN from the **Attribute to use as entity label** drop-down list.

Figure 12.4 Label Attribute

The screenshot shows the SAS MDM interface with the 'Definition' tab selected. On the left, a tree view shows the hierarchy: Vehicle > MDM_ENTITY_TYPE_ID > MDM_VALID_TO_DTTM > MDM_VALID_FROM_DTTM > MDM_LANGUAGE_ID > MDM_SRC_SYS_REC_ID > MDM_SRC_SYS_ID > MDM_ENTITY_CLUSTER_ID > MDM_ENTITY_ID > VIN > MANUFACTURER > MODEL > YEAR > PLATE. The 'Vehicle' entity is selected. The main panel shows the 'Properties' tab for the 'Vehicle' entity. The 'Entity Type Properties' section includes: Type: VEHICLE, Description: Abstract Vehicle type; extend with concrete type, Parent Type: ENTITY, Valid From: (empty), Last Modified: 08/29/13 11:51:56 AM, Last Published: (empty), Valid To: (empty), and a checked checkbox for 'Entity is an abstract entity (cannot be instantiated)'. The 'Display Settings' section includes: Display name: Vehicle, Sort Order: 1, and 'Attribute to use as entity label: VIN'. A dropdown menu is open for 'Attribute to use as entity label', showing the following options: VIN, MDM_SRC_SYS_REC_ID, MANUFACTURER, MODEL, and YEAR.

Note: Only one attribute can be designated as a label attribute in an entity type definition. The label attribute can be inherited from the parent entity type by derived entity types. A label attribute is always required for concrete entity types. The label attribute is defined here so that all types derived from VEHICLE use VIN as their label attribute.

- 6 Click **Save** on the entity type editor toolbar.

Define Flux Attribute

A flux attribute, based on the VIN attribute, is used for field mapping in the standardization job for the VEHICLE entity type (mdm_stnd_vehicle.ddf).

To define a flux attribute:

- 1 Select the VIN attribute in the attributes list of the attribute editor. On the attribute editor toolbar, click  and select **New Flux Attribute** from the menu. A flux attribute named DF_VIN_MC appears in the attributes list. SAS MDM automatically assigns the attribute name by adding a 'DF_' prefix and a '_MC' (match code) suffix to the original attribute name.
- 2 Enter data from the following table in the appropriate locations on the **Properties** tab.

Table 12.2 Flux Attribute

Attribute Option	Attribute Value
Type	DF_VIN_MC
Group	Vehicle Information
Description	VIN Match Code
Data Type	String
Length	60
Constraint	(leave blank)
Required	No
Read-only	No
Encrypt	No
Display name	VIN Match Code
Search form order	0
Search results order	0
Details panel order	0
Cluster table order	0

Attribute Option	Attribute Value
Records editor order	0

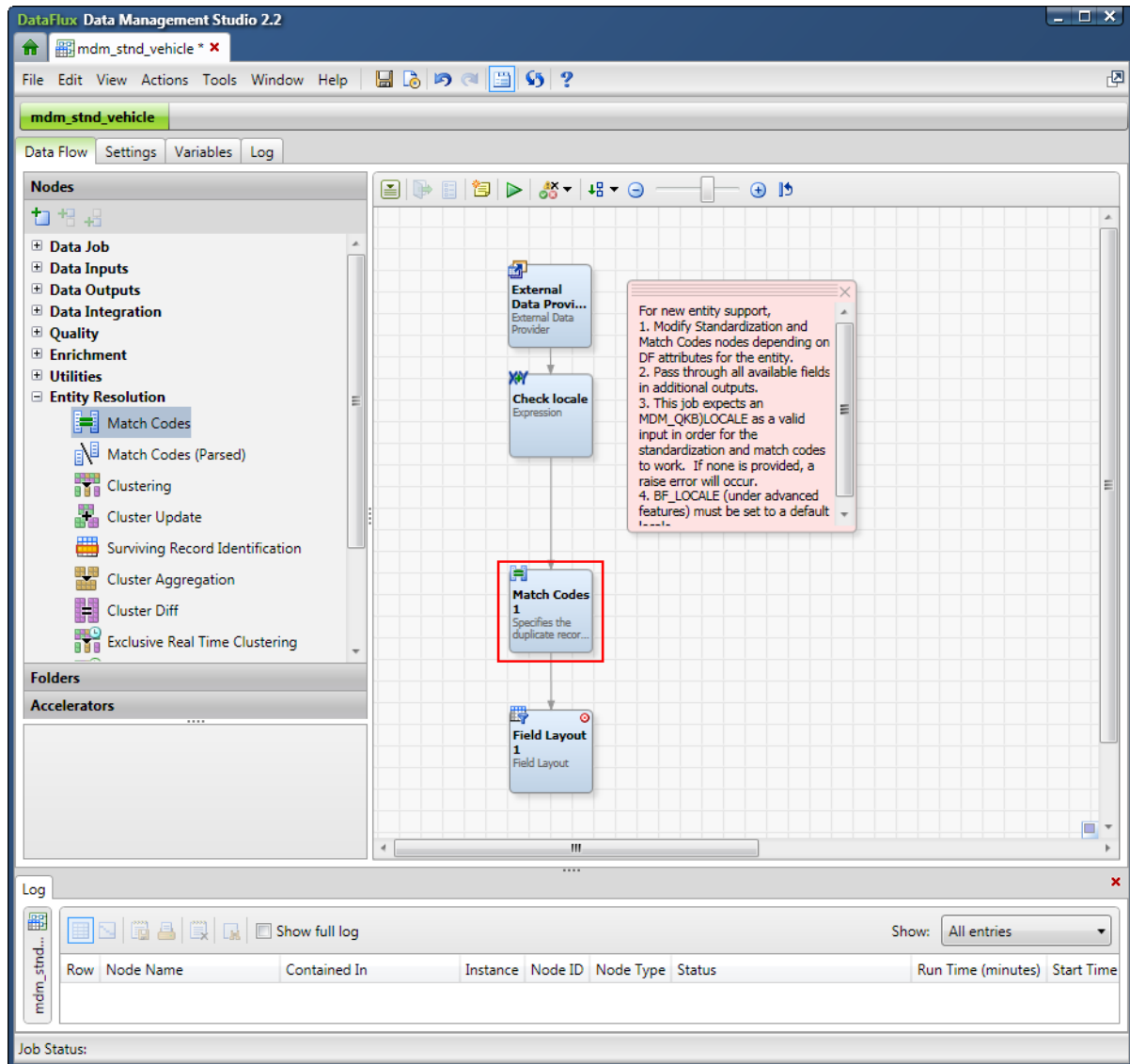
- 3 Click **Save** on the entity type editor toolbar to save the changes.

Add Match Code Generation to the Standardization Job

The steps in this section add match code generation to the standardization job for the VEHICLE entity type (mdm_stnd_vehicle.ddf).

- 1 Select and log on from the DataFlux Data Management Server **Data Management Servers** riser in DataFlux Data Management Studio. Expand the **Real-Time Data Services** folder and the **SAS MDM** folder, and locate the mdm_stnd_vehicle.ddf job.
- 2 Right-click the mdm_stnd_vehicle.ddf job and select **Export**. From the Export To Folder dialog box, export the job to your SAS MDM repository folder.
- 3 Locate and open the mdm_stnd_vehicle.ddf job in the SAS MDM repository folder on the **Folders** riser in DataFlux Data Management Studio.
- 4 Add a **Match Codes** node between the **Check locale** node and the **Field Layout 1** node. The **Match Codes** node generates match codes for the selected fields when the job is run. Match codes enable fuzzy matching of fields.

Figure 12.5 Match Codes Node



- 5 Double-click the **Match Codes** node to display its properties dialog box. In the properties dialog box, do the following:
 - a Double-click the VIN match code field in the **Available** list to transfer it to the **Selected** list.
 - b Select **Text** in the **Definition** column.
 - c Leave the **Sensitivity** value set to its default value of 85.
 - d Enter **DF_VIN_MC** in the **Output Name** column.
 - e Select the **Generate null match codes for blank field values** check box. Selecting this option causes null match codes to be created for null or blank input field values. This prevents null or blank field values from being clustered together.

Figure 12.6 Match Codes Properties

Match Codes Properties

Name:

Locale:

☐ Allow generation of multiple matchcodes per definition for each sensitivity

Match code fields

Available:

- vehicle_offset
- MDM_QKB_LOCALE
- MDM_ENTITY_CLUSTER_ID
- MDM_ENTITY_ID
- MDM_ENTITY_TYPE_ID
- MDM_SRC_SYS_ID
- MDM_SRC_SYS_REC_ID
- MDM_VALID_FROM_DTTM
- MDM_VALID_TO_DTTM
- MANUFACTURER
- MODEL
- PLATE
- VIN**
- YEAR

Selected:

Field Name	Definition	Sensitivity	Output Name
VIN	Text	85	DF_VIN_MC

☒ Generate null match codes for blank field values

☐ Preserve null values

- 6 Save and close the `mdm_stnd_vehicle.ddf` job.
- 7 On the **Data Management Servers** riser, right-click the **Real-Time Data Services** folder and select **Import**. Import the modified `mdm_stnd_vehicle.ddf` job from your SAS MDM repository folder to the DataFlux Data Management Server.

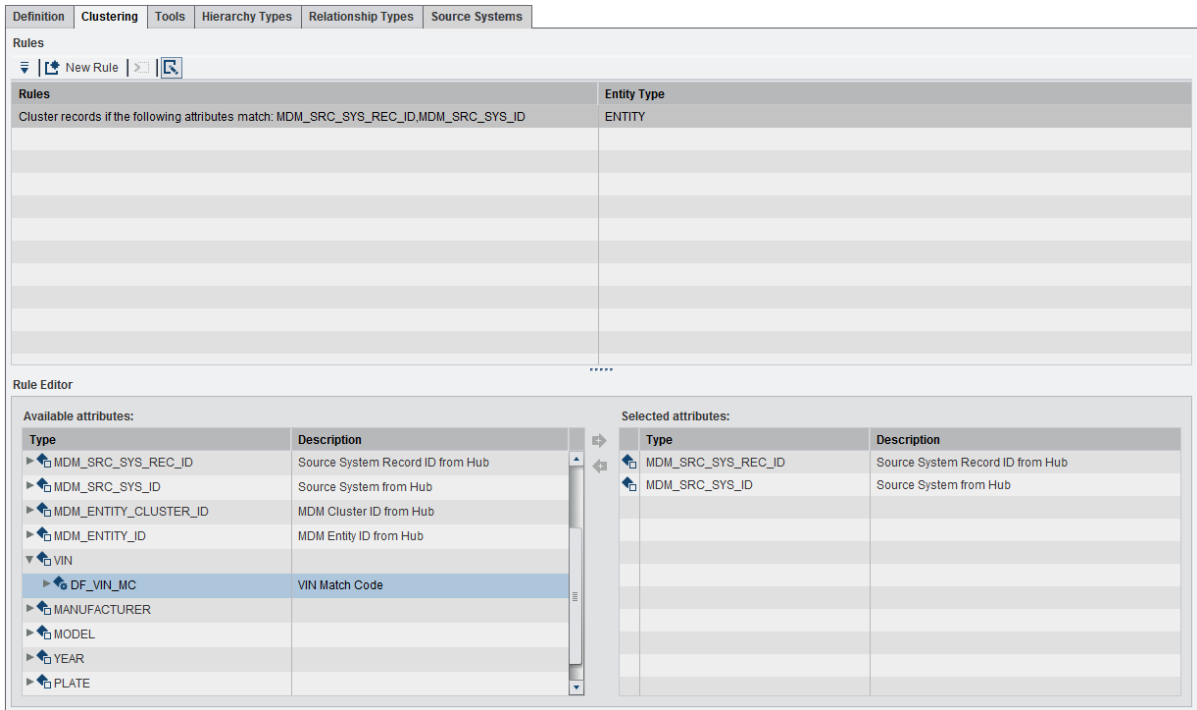
Define Cluster Conditions

The clustering editor is used to create and modify cluster conditions for an entity type definition. Records with the same attribute values specified in the cluster condition are clustered together. A default cluster condition is inherited from the ENTITY (top-level) entity type. The five previously added attributes appear in the **Available** list, as well as the flux attribute (DF_VIN_MC).

To use the clustering editor:

- 1 Click the **Clustering** tab in the entity type editor to display the clustering editor.

Figure 12.7 Clustering Tab



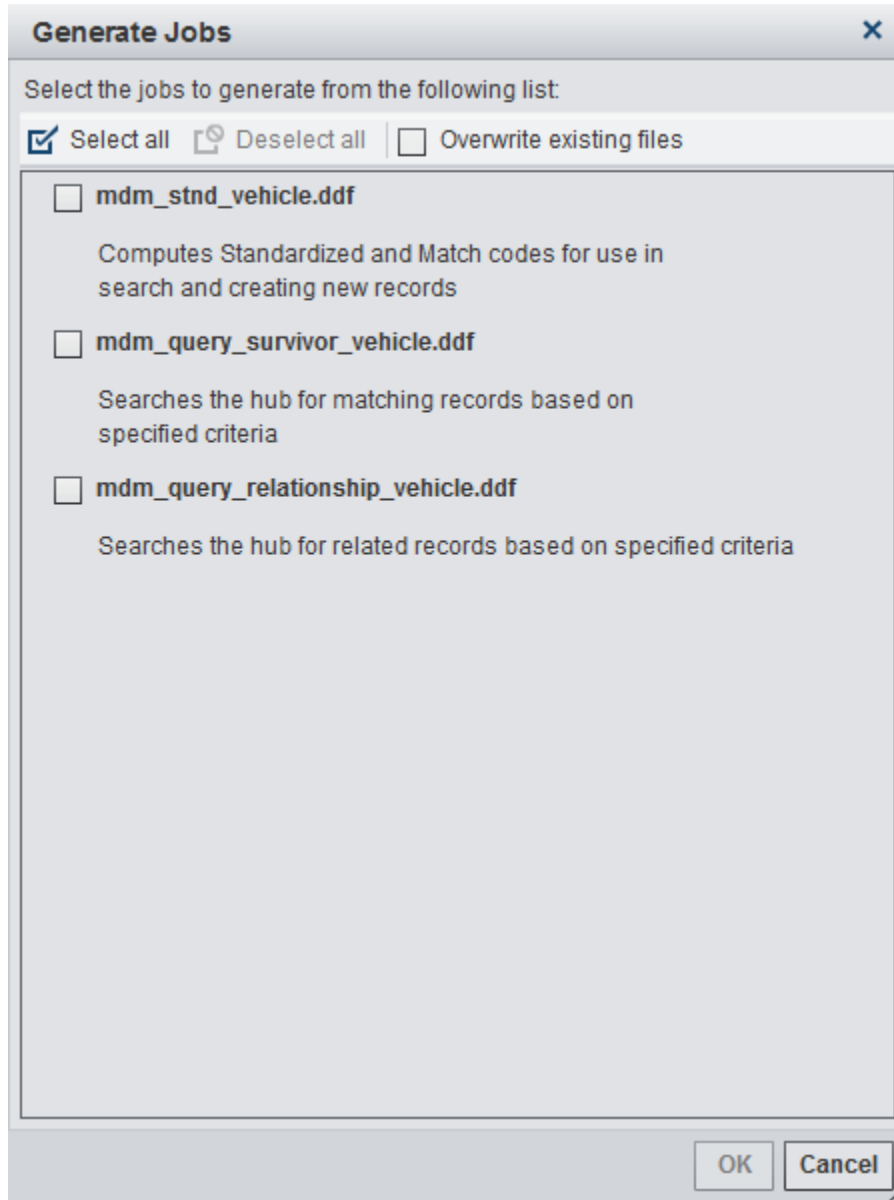
- 2 Click **New Rule** on the clustering editor toolbar. A new entry, **<No attributes in rule>**, appears in the **Rules** list.
- 3 In the **Available** list, select the **DF_VIN_MC** flux attribute. Then click  to transfer **DF_VIN_MC** to the **Selected** list.
- 4 Click **Save** on the entity type editor toolbar to save the changes. The flux attribute now appears in the **Rules** list.

Generate Jobs

After you have created and saved the VEHICLE entity type definition, you generate default batch jobs and data services in support of the entity type. To generate jobs and services:

- 1 Click  on the toolbar of the entity type editor. The Generate Jobs dialog box appears:

Figure 12.8 Generate Jobs Dialog Box



- 2 Select the check boxes for all three available jobs, `mdm_stnd_vehicle.ddf`, `mdm_query_survivor_vehicle.ddf`, and `mdm_query_relationship_vehicle.ddf`, in the Generate Jobs dialog box.
- 3 Click **OK** to generate the jobs. The job generation process uses the metadata that has been saved in the hub and publishes the jobs to your DataFlux Data Management Server.

Publish the Entity Type Definition

Publishing the entity type definition makes the definition active. This includes generating the tables used to store the data in your DBMS, as well as updating the metadata that Master Data Management references when determining which entity types are presented in the user interface.

To publish the definition:

- 1 Click  on the toolbar of the entity type editor. The Publish dialog box appears.
- 2 Click **OK** in the confirmation dialog box to publish the definition.

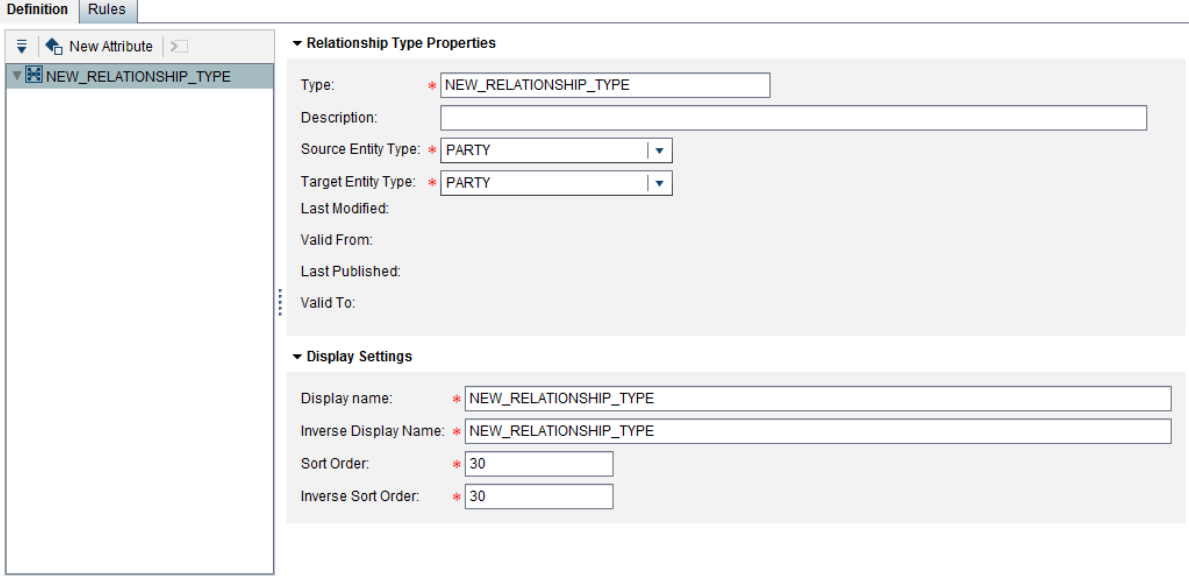
Define Relationship Types

The relationship type editor is used to create and modify relationship types for an entity type definition. Relationship types describe the types of relationships that can exist between the associated entity types. Relationship types define source and target entity types and have optional match conditions that can be used to determine whether two entities are related.

To use the relationship type editor:

- 1 Click the **Relationship Types** tab in the entity types editor.
- 2 Click **New Relationship Type** on the relationship type toolbar to add a new relationship type. The relationship type editor opens on a new tab.

Figure 12.9 Definition Editor



The screenshot shows the 'Definition Editor' with two tabs: 'Definition' and 'Rules'. The 'Definition' tab is active, showing a list of entity types on the left, including 'NEW_RELATIONSHIP_TYPE'. The main area displays the 'Relationship Type Properties' and 'Display Settings' for this new type.

Relationship Type Properties

- Type: * NEW_RELATIONSHIP_TYPE
- Description:
- Source Entity Type: * PARTY
- Target Entity Type: * PARTY
- Last Modified:
- Valid From:
- Last Published:
- Valid To:

Display Settings

- Display name: * NEW_RELATIONSHIP_TYPE
- Inverse Display Name: * NEW_RELATIONSHIP_TYPE
- Sort Order: * 30
- Inverse Sort Order: * 30

Note: The **Valid From** and **Valid To** fields are blank. The **Valid From** field is automatically completed when you save and publish the entity type definition in later steps. The **Valid To** field remains blank until the entity type definition is retired.

- 3 Enter data from the Relationship Type 1 Value column of the following table in the appropriate locations on the tab.

Table 12.3 Relationship Type Values

Definition Editor Field	Relationship Type 1 Value	Relationship Type 2 Value
Type	DRIVER	OWNER

Definition Editor Field	Relationship Type 1 Value	Relationship Type 2 Value
Description	Used to associate people with vehicles that they drive.	Used to associate people with vehicles that they own.
Source Entity Type	VEHICLE	VEHICLE
Target Entity Type	INDIVIDUAL	PARTY
Display Name	Driven By	Owned By
Inverse Display Name	Drives	Owns
Sort Order	10	20
Inverse Sort Order	10	20

Figure 12.10 DRIVER Relationship Type

The screenshot shows the SAS MDM Definition Editor interface. On the left, a tree view shows the 'DRIVER' relationship type selected. The main panel is divided into two sections: 'Relationship Type Properties' and 'Display Settings'. In the 'Relationship Type Properties' section, the 'Type' is set to 'DRIVER', the 'Description' is 'Used to associate people with vehicles that they drive', the 'Source Entity Type' is 'VEHICLE', and the 'Target Entity Type' is 'INDIVIDUAL'. The 'Display Settings' section shows the 'Display name' as 'Driven By', the 'Inverse Display Name' as 'Drives', the 'Sort Order' as '10', and the 'Inverse Sort Order' as '10'. The interface includes tabs for 'Definition' and 'Rules' at the top.

Note: The **Target Entity Type** for the DRIVER relationship type is INDIVIDUAL, because only a person can drive a vehicle.

- 4 Click **Save** on the relationship type editor toolbar to save the new DRIVER relationship.
- 5 Repeat this task to add the new OWNER relationship.

Note: The **Target Entity Type** for the OWNER relationship type is PARTY. Both INDIVIDUAL and COMPANY are concrete subtypes of PARTY, because both individuals and companies can own a vehicle.

Define Relationship Type Attributes

Use the following task to add two new attributes to the DRIVER relationship type: LICENSE_TYPE and PRIMARY_DRIVER.

To define attributes:

- 1 Click **New Attribute** on the toolbar to add a new attribute. The new attribute appears in the attribute list on the left pane of the **Definition** tab.

Figure 12.11 New Relationship Type Attribute

Note:

- The **Length** field is not applicable for INTEGER attributes.
- The **Valid From** and **Valid To** fields are blank. The **Valid From** field is automatically completed when you save and publish the entity type definition in later steps. The **Valid To** field remains blank until the entity type definition is retired.

- 2 Enter data from the Attribute 1 Value column of the following table in the appropriate locations on the **Definition** tab.

Note: For check box options, **yes** means select the check box. **no** means clear the check box.

Table 12.4 New Relationship Type Attributes

Attribute Option	Attribute 1 Value	Attribute 2 Value
Type	LICENSE_TYPE	PRIMARY_DRIVER
Group	Details	Details
Description	Denotes the type of license the driver has for the vehicle	Indicates whether the individual is the primary driver for the vehicle.
Data Type	LIST	BOOLEAN

Attribute Option	Attribute 1 Value	Attribute 2 Value
Length	20	1
Constraint	Student;Full;Restricted;Commercial	(leave blank)
Read-only	No	No
Encrypt	No	No
Display name	License Type	Primary Driver Flag
Details panel order	10	20
Record editor order	10	20

Figure 12.12 LICENSE_TYPE Attribute

The screenshot shows the 'Definition' tab of the relationship type editor. On the left, a tree view shows 'DRIVER' expanded, with 'LICENSE_TYPE' selected. The main panel displays the 'Attribute Type Properties' and 'Display Settings' for 'LICENSE_TYPE'.

Attribute Type Properties

- Type: * LICENSE_TYPE
- Group: Details
- Description: Denotes the kind of license the driver has for the vehicle
- Data Type: LIST
- Length: * 20
- Constraint: Student;Full;Restricted;Commercial
- Valid From:
- Valid To:
- ☐ Attribute is read only
- ☐ Encrypt attribute

Display Settings

- Display name: * License Type
- Details panel order: * 10
- Record editor order: * 10

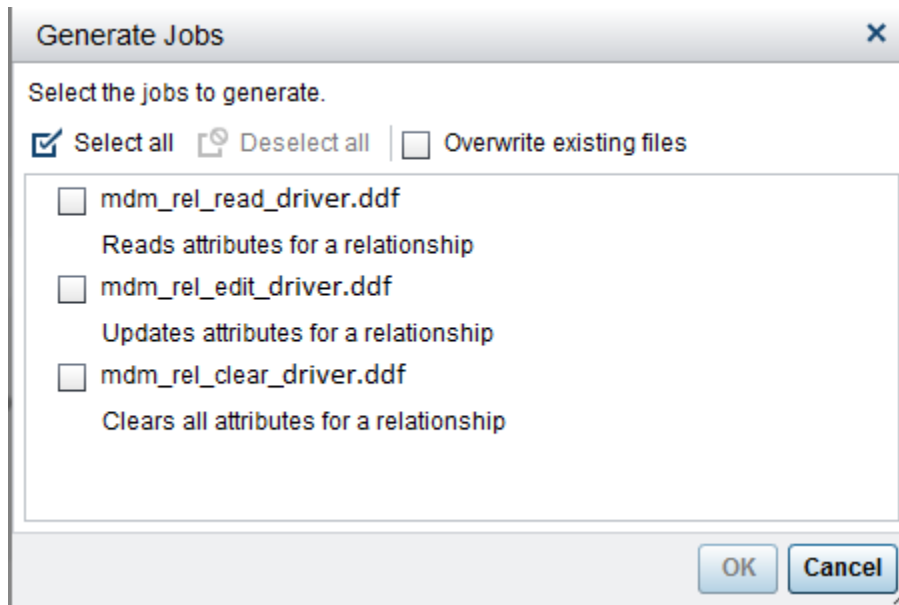
- 3 Click **Save** on the relationship type editor toolbar to save the LICENSE_TYPE attribute.
- 4 Repeat this task to add the new PRIMARY_DRIVER attribute specified in the table.
- 5 Click **Save** on the relationship type editor toolbar.

Generate Jobs

After you have created and saved the DRIVER relationship type definition, you generate default data services in support of the review type. To generate services:

- 1 Click  on the toolbar of the entity type editor. The Generate Jobs dialog box appears:

Figure 12.13 Generate Jobs Dialog Box



- 2 Select the check boxes for all three available jobs, `mdm_rel_read_driver.ddf`, `mdm_rel_edit_driver.ddf`, and `mdm_rel_clear_driver.ddf`, in the Generate Jobs dialog box.
- 3 Click **OK** to generate the jobs. The job generation process uses the metadata that has been saved in the hub and publishes the jobs to your DataFlux Data Management Server.

Publish the Relationship Type Definition

Publishing the relationship type definition makes the definition active. This includes generating the tables used to store the data in your DBMS, as well as updating the metadata that Master Data Management references when determining which relationship types are presented in the user interface.

To publish the definition:

- 1 Click  on the toolbar of the entity type editor. The Publish dialog box appears.
- 2 Click **OK** in the confirmation dialog box to publish the definition.

Example 2: Creating a Concrete Entity Type

Overview of Creating a Concrete Entity Type

The example presented in this section builds on the Vehicle example described in “[Example 1: Creating an Abstract Entity Type](#)”. Because the Vehicle entity type is modeled as abstract, instances of this entity type cannot be created. Instead, concrete entity types must be created from this metadata in the hub. This example creates a concrete entity type named Car, which inherits metadata from the Vehicle entity type.

Open the Entity Type Editor

To open the entity type editor:

- 1 Click **Data Model** on the SAS Data Management Console page. The **Data Model** tab opens.
- 2 Click **New Entity Type** on the **Entity Types** toolbar to open the entity type editor. The entity type editor opens on a new tab.

Define Entity Type Properties

To use the definition editor:

- 1 Click the **Definition** tab in the entity type editor to display the **Properties** tab.
- 2 Enter **CAR** as the entity type in the **Type** field.
- 3 Enter **Automobile** in the **Description** field.
- 4 Select **VEHICLE** in the **Parent Type** field. This value specifies the parent entity type that contributes metadata to the Car entity type.

Note: The **Valid From**, **Last Modified**, **Last Published**, and **Valid To** fields in the **Entity Type Properties** pane are blank. The **Valid From**, **Last Modified**, and **Last Published** fields are automatically completed when you save and publish the entity type definition in later steps. The **Valid To** field remains blank until the entity type definition is retired.

- 5 Clear the **Entity is an abstract entity** check box.
- 6 Enter **car** in the **Display Name** field as the label for the entity type.
- 7 Enter 2 in the **Sort Order** field. This causes the CAR entity type to appear after the VEHICLE entity type in the selection lists in the Master Data Management user interface.

Figure 12.14 Definition Tab

Definition Clustering Tools Hierarchy Types Relationship Types Source Systems

New Attribute

CAR

- MDM_ENTITY_TYPE_ID
- MDM_VALID_TO_DTTM
- MDM_VALID_FROM_DTTM
- MDM_LANGUAGE_ID
- MDM_SRC_SYS_REC_ID
- MDM_SRC_SYS_ID
- MDM_ENTITY_CLUSTER_ID
- MDM_ENTITY_ID
- VIN
- MANUFACTURER
- MODEL
- YEAR
- PLATE

Properties Permissions

Entity Type Properties

Type: * CAR

Description: Automobile

Parent Type: VEHICLE

Last Modified:

Valid From:

Last Published:

Valid To:

☐ Entity is an abstract entity (cannot be instantiated)

Display Settings

Display name: * VEHICLE

Sort Order: * 2

Attribute to use as entity label: VIN

Remediation User-defined Fields

User-defined field 1:

User-defined field 2:

User-defined field 3:

Source System Harmonization

Select which source systems to push data to when harmonizing with MDM hub clusters:

☒ All source systems that can contribute records to clusters of this entity type

☐ Source systems that contributed records to the clusters being harmonized

☐ Select source systems:

Display Name
☐ MD Manager
☐ ERP
☐ Marketing
☐ Orion
☐ Call Center UK
☐ Call Center US

8 Click **Save** on the entity type editor toolbar to save the changes.

Define Attributes

Use the following task to add a Capacity attribute to the Car entity type.

To define the attribute:

- 1 Click **New Attribute** on the toolbar to add a new attribute. The new attribute appears in the attribute list on the left side of the **Definition** tab.

Figure 12.15 New Attribute

Note:

- The **Length** field is not applicable for INTEGER attributes.
- The **Valid From** and **Valid To** fields are blank. The **Valid From** field is automatically completed when you save and publish the entity type definition in later steps. The **Valid To** field remains blank until the entity type definition is retired.

- 2 Enter data from the following table in the appropriate locations on the **Properties** tab.

Note:

- For check box options, **Yes** means select the check box. **No** means clear the check box.
- The **Search Form Order** value for the new attribute is lower than 100, which is the lowest **Search Form Order** value for attributes that are inherited from the ENTITY (top-level) entity type. The value 98 causes the Capacity attribute to appear immediately after the Plate attribute in the Search fields on the **Master Data Management** tab.

Table 12.5 Capacity Attribute

Attribute Option	Attribute Value
Type	CAPACITY
Group	Vehicle Information
Description	Number of Passengers
Data Type	Integer

Attribute Option	Attribute Value
Length	(leave blank)
Constraint	(leave blank)
Required	No
Read-Only	No
Encrypt	No
Display name	Number of Passengers
Search form order	99
Search results order	99
Details panel order	99
Cluster table order	99
Record editor order	99

- 3 Click **Save** on the entity type editor toolbar to save the new CAPACITY attribute. The CAPACITY attribute now appears in the attributes list.

Figure 12.16 Capacity Attribute

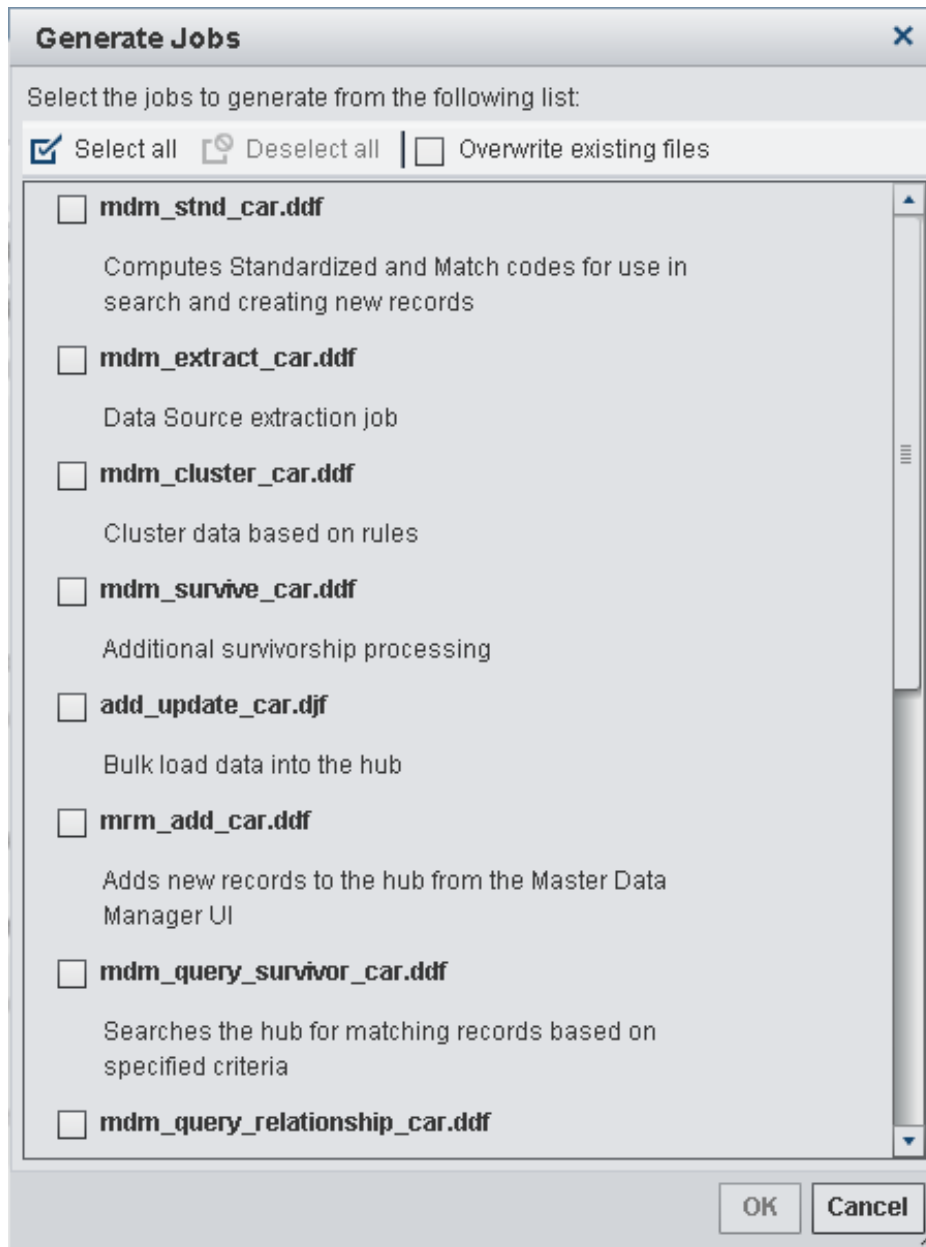
The screenshot shows the 'Definition' tab of the entity type editor. On the left, a tree view shows the 'Car' entity with various attributes, including 'CAPACITY' which is selected. The main panel shows the 'Properties' tab for the 'CAPACITY' attribute. The 'Attribute Type Properties' section includes fields for Type (CAPACITY), Group (Vehicle Information), Description (Number of Passengers), Data Type (INTEGER), Length (0), Constraint, Valid From, and Valid To. Below this are checkboxes for 'Attribute is required', 'Attribute is read only', and 'Encrypt attribute'. The 'Display Settings' section includes fields for Display name (Number of Passengers), Search form order (99), Search results order (99), Details panel order (99), Cluster table order (99), and Record editor order (99).

Generate Jobs

After you have created and saved the CAR entity type definition, generate default batch jobs and data services in support of the entity type. To generate jobs and services:

- 1 Click  on the toolbar of the entity type editor. The Generate Jobs dialog box appears:

Figure 12.17 Generate Jobs Dialog Box



- 2 Select **Check All** in the Generate Jobs dialog box.

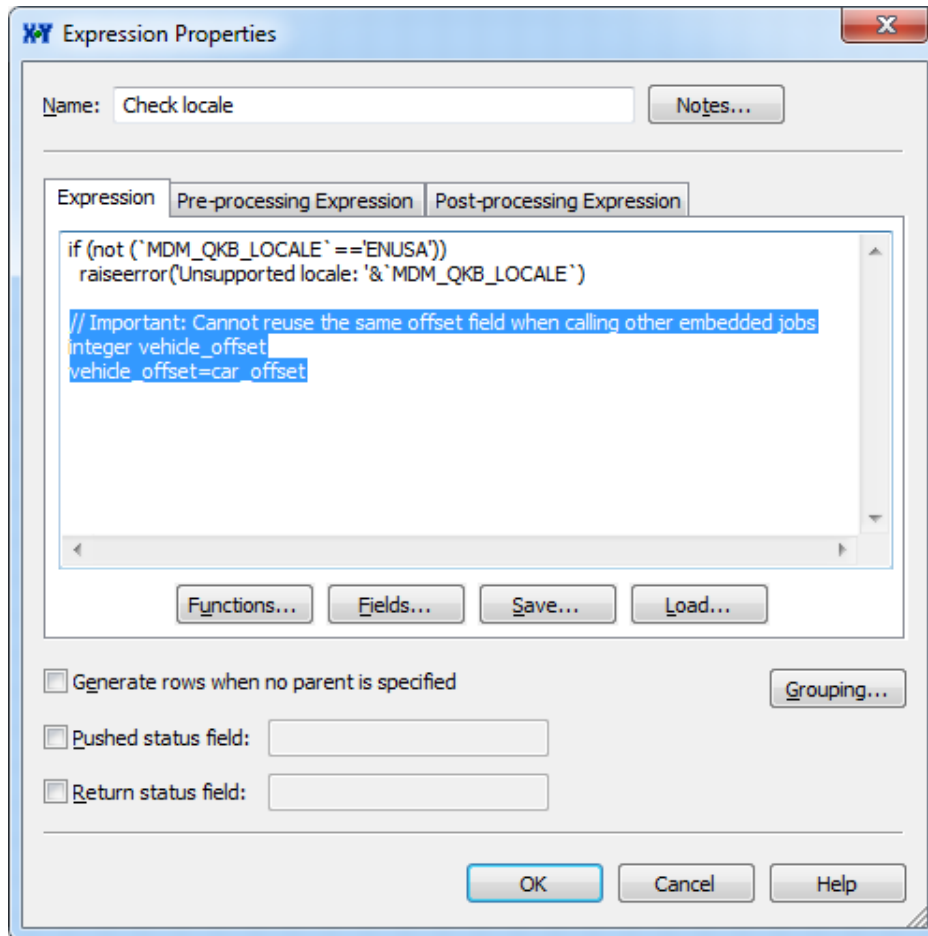
- 3 Click **OK** to generate the jobs. The job generation process uses the metadata that has been saved in the hub and publishes the jobs to your DataFlux Data Management Server.

Add Match Code Generation to the Standardization Job

The steps in this section modify the `mdm_stnd_car.ddf` standardization job to reference the `mdm_stnd_vehicle.ddf` standardization job so that the CAR entity type uses the standardization defined by the Vehicle entity type.

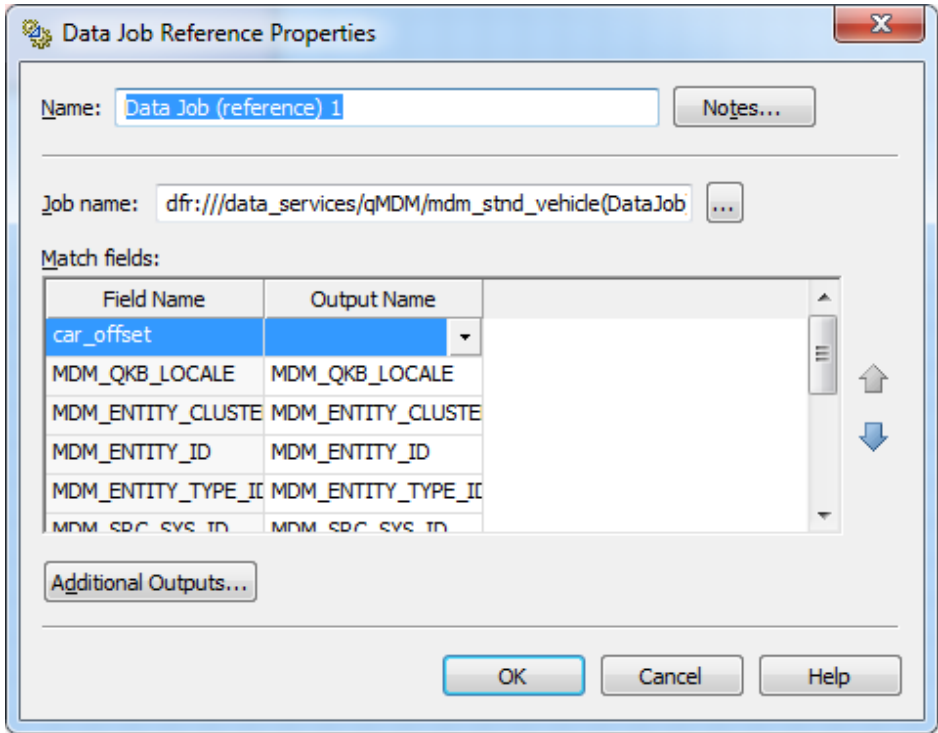
- 1 Select and log on to the DataFlux Data Management Server from the **Data Management Servers** riser in DataFlux Data Management Studio. Expand the **Real-Time Data Services** folder and the **SAS MDM** folder, and locate the `mdm_stnd_car.ddf` job.
- 2 Right-click the `mdm_stnd_car.ddf` job and select **Export**. From the Export To Folder dialog box, export the job to your SAS MDM repository folder.
- 3 Locate and open the `mdm_stnd_car.ddf` job in the SAS MDM repository folder on the **Folders** riser in DataFlux Data Management Studio.
- 4 Double-click the **Check Locale** node to display its properties dialog box. On the **Expression** tab, declare an integer variable named 'vehicle_offset', and set the variable to 'car_offset'.

Figure 12.18 Check Locale Node



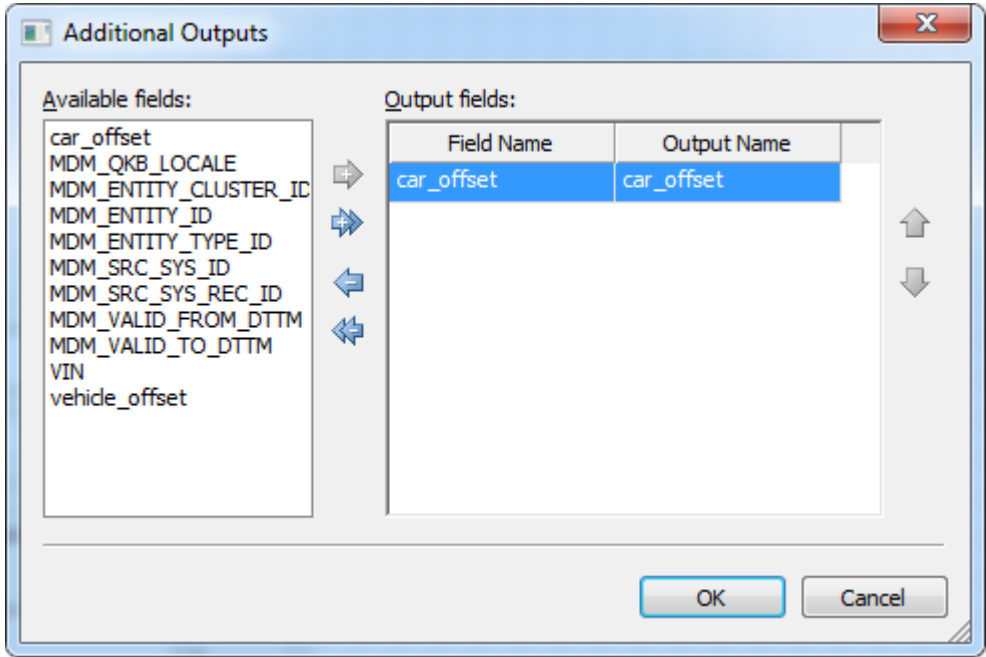
- 5 Click **OK** to save the changes.
- 6 Add a **Data Job Reference** node between the **Check Locale** node and the **Field Layout 1** node. The **Data Job Reference** node uses the `mdm_stnd_vehicle.ddf` job to compute the VIN match code.
- 7 Double-click the **Data Job Reference** node to display its Properties dialog box. In the Properties dialog box, match each field with its corresponding parameter in the `mdm_stnd_vehicle.ddf` standardization job. Note that the **car_offset** field does not have a corresponding parameter.

Figure 12.19 Data Job Reference Properties



8 Click **Additional Outputs** to display the Additional Outputs dialog box.

Figure 12.20 Additional Outputs



- 9 In the Additional Outputs dialog box, select the **car_offset** field as an additional field to be returned from the mdm_stnd_car.ddf standardization job.
- 10 Click **OK** to save the changes.

- 11 Click **OK** to close the Properties dialog box for the **Data Job Reference** node.
- 12 Save and close the mdm_std_car.ddf job.
- 13 On the **Data Management Servers** riser, right-click the **Real-Time Data Services** folder and select **Import**. Import the modified mdm_std_car.ddf job from your SAS MDM repository folder to the DataFlux Data Management Server.

Publish the Entity Type Definition

The final step is to publish the CAR entity type definition. Publishing the entity type definition makes the definition active. This includes generating the tables used to store the data in your DBMS, as well as updating the metadata that Master Data Management references when determining which entity types are presented in the user interface.

To publish the definition:

- 1 Click  on the toolbar of the entity type editor. The Publish dialog box appears.
- 2 Click **OK** in the confirmation dialog box to publish the definition.

Because CAR has been defined as a concrete entity type, new instances of this entity type can be created.

Example 3: Creating Entity Relationships

Overview of Creating Entity Relationships

The example presented in this section demonstrates the following:

- creating an instance of the Car entity type created in [“Example 2: Creating a Concrete Entity Type” on page 164](#)
- creating Owned By and Driven By entity relationships for the entity
- exploring the Owned By and Driven By entity relationships graphically

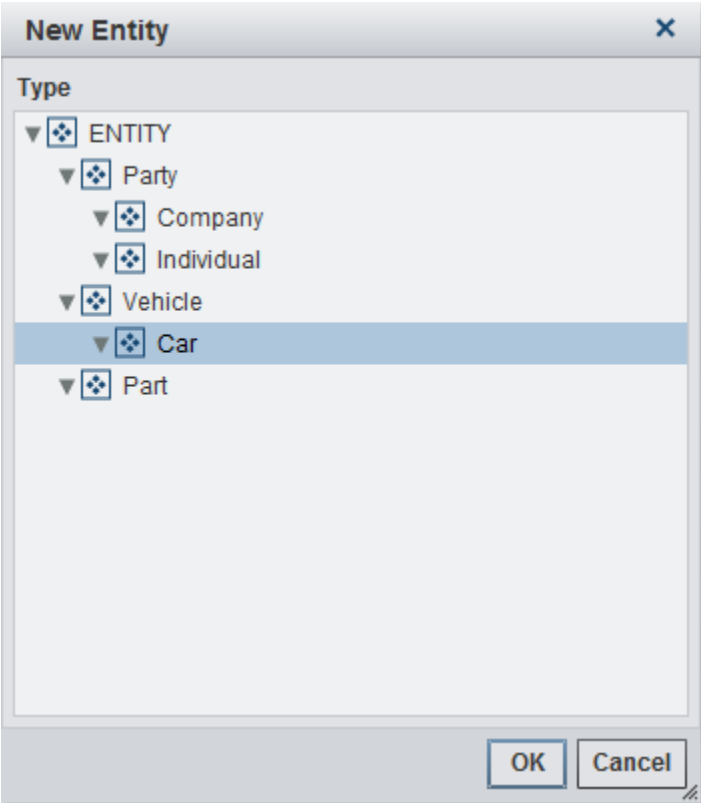
The example also illustrates the inheritance of presentation metadata specified for the Car entity type and for the Vehicle entity type created in [“Example 1: Creating an Abstract Entity Type” on page 147](#).

Create Entity

To create an entity:

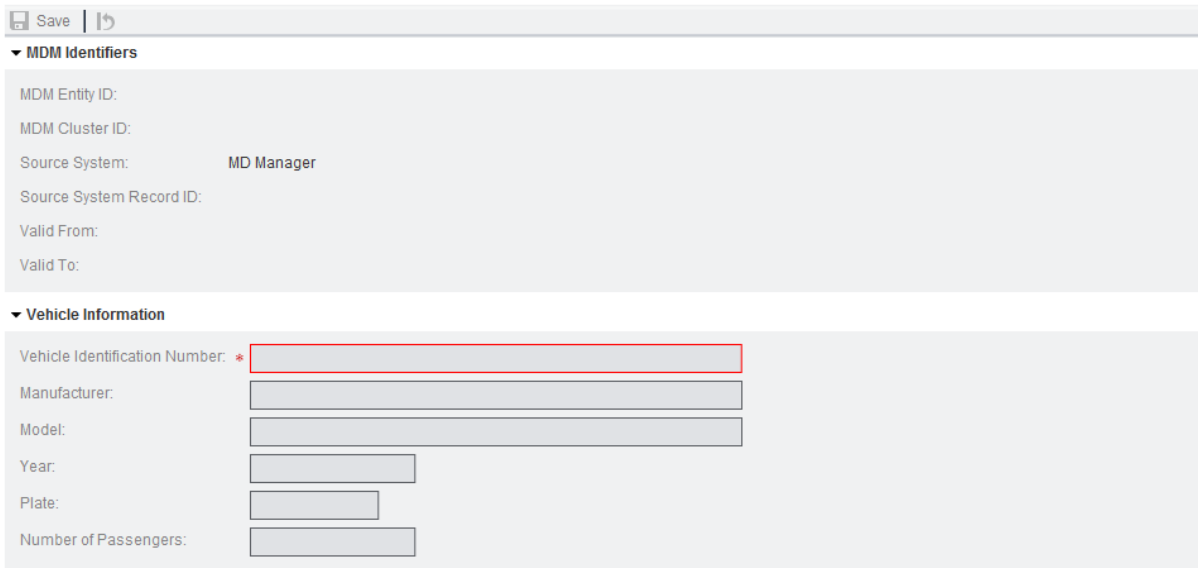
- 1 Click **Master Data Management** on the SAS Data Management Console page. The **Master Data Management** tab opens.
- 2 Click **New Entity** on the toolbar to open the New Entity dialog box:

Figure 12.21 New Entity Dialog Box



- 3 Select the Car entity type in the New Entity dialog box.
- 4 Click **OK** to open the entity editor.

Figure 12.22 New Entity Editor



The entity editor contains a **Vehicle Information** pane. This name of this pane is determined by the **Group** values of the five attributes created for the VEHICLE entity type. The **Vehicle Information** pane contains Vehicle Identification Number, Manufacturer, Model, Year, and Plate fields. These field labels are based on the **Label** values of the five attributes created for

the VEHICLE entity type. These attributes are inherited by the CAR entity type. The **Vehicle Information** pane also contains a **Number of Passengers** field. This field label is based on the **Label** value of the Capacity attribute created for the CAR entity type.

- 5 In the entity editor, enter values from the following table for the fields in the **Vehicle Information** pane.

Table 12.6 Vehicle Information Values

Field	Value
Vehicle Identification Number	A94875
Manufacturer	Acme Car Company
Model	FastRide GT
Year	2012
Plate	ABC-1234
Number of Passengers	4

Figure 12.23 New Entity Editor

The screenshot shows the 'New Entity Editor' interface. At the top, there is a toolbar with a 'Save' button and a 'Help' icon. Below the toolbar, the 'MDM Identifiers' section is expanded, showing fields for 'MDM Entity ID:', 'MDM Cluster ID:', 'Source System:' (set to 'MD Manager'), 'Source System Record ID:', 'Valid From:', and 'Valid To:'. The 'Vehicle Information' section is also expanded, showing input fields for 'Vehicle Identification Number:' (with a red asterisk and the value 'A94875'), 'Manufacturer:' (with the value 'Acme Car Company'), 'Model:' (with the value 'FastRide GT'), 'Year:' (with the value '2012'), 'Plate:' (with the value 'ABC-1234'), and 'Number of Passengers:' (with the value '4').

- 6 Click **Save** on the toolbar to save the changes. Two new panes, **Driven By** and **Owned By**, appear below the **Vehicle Information** pane:

Figure 12.24 New Entity Editor with Relationship Panes

Save

Matrix Edit

▼ MDM Identifiers

MDM Entity ID:

21787

MDM Cluster ID:

6383

Source System:

Best Record

Source System Record ID:

21787

Valid From:

07/31/13 11:32:49 AM

Valid To:

▼ Vehicle Information

Vehicle Identification Number: *

A94875

Manufacturer:

Acme Car Company

Model:

FastRide GT

Year:

2012

Plate:

ABC-1234

Number of Passengers:

4

▶ Driven By

▶ Owned By

Create Entity Relationships

- The **Driven By** and **Owned By** panes correspond to the Driver and Owner relationship types created for the VEHICLE entity type. You can search for and select entities among which you want to create relationships.
- To create entity relationships:
- 1 Click  to expand the **Driven By** pane.
 - 2 Click **New Relationship Type** on the toolbar to display the New Relationship dialog box:

Figure 12.26 New Relationship Dialog Box

Save

Matrix Edit

MDM Identifiers

Vehicle Information

Driven By

New Relationship

MDM Entity Type	MDM Cluster ID	Full Name	Do not Call	Last Contact	Address Line 1 (Standardized)
Individual	282	ABBEY CATA	No	01/14/11 08:58:...	206 Odonnell Dr

New Relationship

MDM Entity Type	MDM Cluster ID	Full Name	Address Line 1 (Standardized)	City
Company	6023	COLBURN BANKING	4009 Summitview Ave	MERION

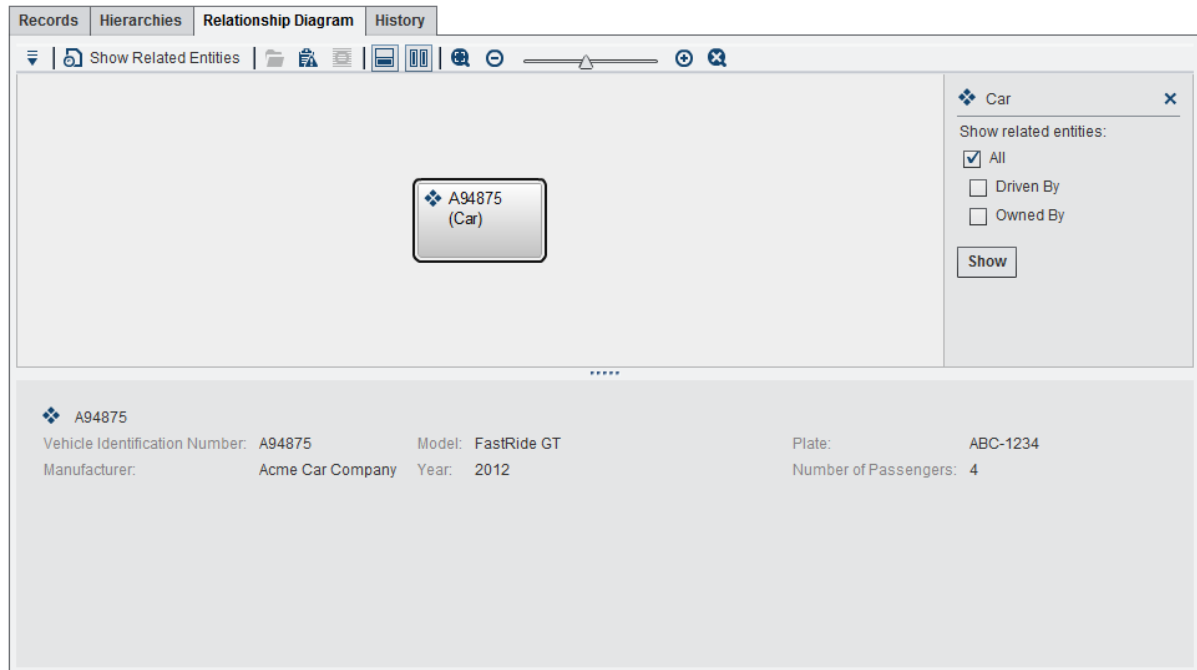
Explore Entity Relationships

- 1

Click the **Relationship Diagram** tab in the entity editor to explore the previously created entity relationships graphically.
- 2

Select the A94875 entity node in the entity graph on the **Relationship Diagram** tab. Leave the **All** check box selected.

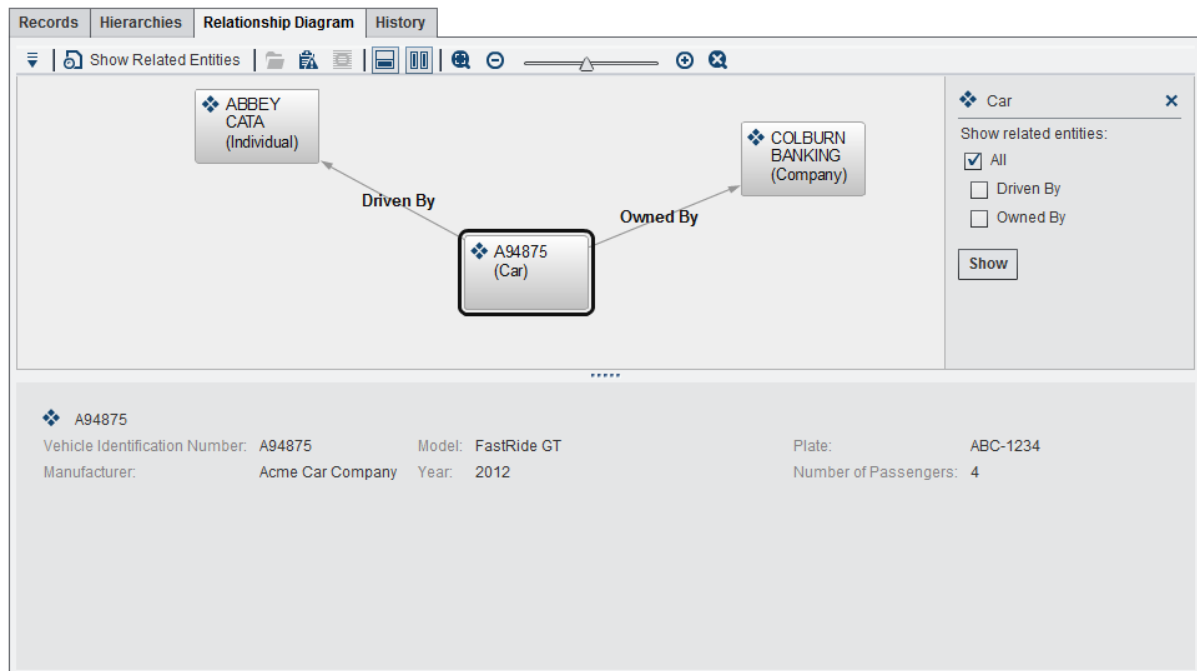
Figure 12.27 Initial Relationship Tab



3 Click **Show**.

4 The entity graph pane now displays nodes for the Abbey Cata and Colburn Banking entity records. The arrows between the nodes indicate the Driven By and Owned By relationships with the A94875 entity. Clicking each node displays additional details for the entity in the details pane.

Figure 12.28 Populated Relationship Tab



Note: You can click and drag the nodes to rearrange them on the entity graph. To move the entire graph, press Shift while clicking and dragging the graph.

- 5 Select the Driven By relationship on the graph.
- 6 Edit its attributes to set the primary driver.
- 7 Click **Save**.
- 8 Select the new attribute to show its details in the Details panel.

Recommended Reading

- *DataFlux Data Management Studio Installation and Configuration Guide*
- *DataFlux Data Management Studio User's Guide*
- *DataFlux Data Management Server Administrator's Guide*
- *DataFlux Data Management Server Users's Guide*
- *SAS Workflow Studio User's Guide*
- *SAS Intelligence Platform Middle-Tier Administration Guide*
- *SAS Management Console User's Guide*
- *SAS MDM Administrator's Guide*
- *SAS Task Manager User's Guide*
- *SAS Data Remediation User's Guide*

For a complete list of SAS books, go to support.sas.com/bookstore. If you have questions about which titles you need, please contact a SAS Book Sales Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-3228
Fax: 1-919-677-8166
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore

Glossary

authorization

the process of determining the permissions that particular users have for particular resources. Authorization either permits or denies a specific action on a specific resource, based on the user's identity and on group memberships.

business rule

a statement of business logic that specifies conditions to be evaluated and actions to be taken if those conditions are satisfied.

database

an organized collection of related data. A database usually contains named files, named objects, or other named entities such as tables, views, and indexes.

database management system

a software application that enables you to create and manipulate data that is stored in the form of databases. Short form: DBMS.

DBMS

See database management system

ETL

See extract, transform, load

extract, transform, load

a data warehousing process in which data is extracted from outside sources, transformed according to operational and quality needs, and loaded into a target database. Short form: ETL.

locale

a setting that reflects the language, local conventions, and culture for a geographic region. Local conventions can include specific formatting rules for paper sizes, dates, times, and numbers, and a currency symbol for the country or region. Some examples of locale values are French_Canada, Portuguese_Brazil, and Chinese_Singapore.

match

a set of values that produce identical match codes or identical match code components. Identical match codes are assigned to clusters.

match code

an encoded representation of an input character variable that is used to cluster and compare data.

match definition

a part of a locale that is referenced during the creation of match codes. Each match definition is specific to a category of data content. In the ENUSA locale, for example, match definitions are provided for names, e-mail addresses, and street addresses, among others.

metadata

descriptive data about data that is stored and managed in a database, in order to facilitate access to captured and archived data for further use.

repository

a storage location for data, metadata, or programs.

rule

See business rule

Simple Object Access Protocol

See SOAP

SOAP

a standard, web-based protocol that enables applications written in various programming languages to communicate. Short form: SOAP.

SQL

See Structured Query Language

Structured Query Language

a standardized, high-level query language that is used in relational database management systems to create and manipulate objects in a database management system. SAS implements SQL through the SQL procedure. Short form: SQL.

Index

Special Characters

.NET applications, Web service calls from 10

A

abstract entity types
 defining 102
 example of creating 147, 172
 generating jobs for 32, 99
 overview 27
 access
 control 43
 managing 47
 applications, integrating with SAS
 MDM 9
 architecture 4
 archive files, generating 39
 attribute properties 103
 attributes
 constraints 105
 creating 103
 data types 28, 105
 display order 29
 encrypting 30, 106
 flux 17, 104
 grouping 105
 inherited 26
 label attribute 35, 102, 153
 length 105
 modifying 103
 read-only 29, 105
 regular expression constraints 105
 required 105
 retiring 104
 retiring required 30, 104

B

batch jobs 7
 Boolean data type 28
 business tier 5

C

cleansing, data 16
 cluster conditions 30
 creating 107
 modifying 107
 clustering 17
 clustering editor 107
 clustering tab, entity type editor 107
 clusters
 specifying conditions 107
 command line interface 5
 concrete entity types
 example of creating 164
 constraints, attribute 105
 country code, in language definitions 137

D

data cleansing 16
 data clustering 17
 data encryption 30, 106
 data model 93
 data profiling 12, 16, 27
 data quality 5, 16
 data standardization 5, 16
 data tier 5
 data types, supported 28, 105
 data, trimming 39
 DataFlux Data Management Server 7, 10
 licensing 17
 permissions 47
 DataFlux Data Management Studio 6, 21
 licensing 17
 Match Codes node 21
 date data type 28
 deep linking 65
 default attributes
 grouping 105
 definition editor 100
 definition tab
 entity type editor 100
 definitions

- language 135
- source system 129
- deploying SAS MDM 25
- derived entity types 27
- dfwproc 35
- display order, attribute 29

E

- editing entities 68
- encrypting data 30, 106
- end-user applications, integrating with SAS MDM 9
- enterprise applications, integrating with SAS MDM 9
- Enterprise Resource Planning 12
- entities
 - construction of 14
 - overview 13
 - relationships 14
- ENTITY (top-level entity type) 26, 102, 108
- entity relationships
 - example of creating 172
- entity type editor 96
 - attributes properties 103
 - clustering tab 107
 - definition tab 100
 - relationship types tab 112
 - tools tab 108
- entity types 96
 - abstract 27, 102
 - cluster conditions 107
 - concrete 102
 - definition 100
 - derived 27
 - example of creating 147, 172
 - parent 102
 - publishing 97, 98
 - relationship types 112, 113, 121
 - retiring 96
 - sample 31
- entity types editor
 - relationship types tab 113, 121
- ERP 12
- ETL 9, 13, 39
- examples
 - creating a concrete entity type 164
 - creating an abstract entity type 147, 172
 - creating entity relationships 172
- Extraction, Transformation, and Loading 9, 13, 39

F

- flux attributes 17
 - creating 104
 - grouping 105

G

- generate jobs dialog box 97, 98
- global preferences 54
- groups, defining 46

H

- history tab 78
- home page, SAS Data Management Console 52

I

- integer data type 28
- integrating other applications with SAS MDM 9
- interface, SAS MDM 56
- IS_MAX_LOAD 35

J

- Java applications, Web service calls from 10
- jobs
 - designing 47
 - generating for entity types 32

L

- label attribute 35, 102, 153
- language definitions 135
- licensing, QKB 17, 135
- list data type 28, 105
- load problems, recovering from 39
- locales 135
 - adding 17
 - licensing 17
- log on, SAS Data Management Console 51

M

master data management [61](#), [127](#)
 Master Data Management Foundation
 [41](#)
 Master Data Management user
 interface [7](#)
 match codes [5](#), [17](#), [20](#)
 Match Codes node, DataFlux Data
 Management Studio [21](#)
 match conditions, for entity
 relationship types [119](#), [123](#)
 match definitions [21](#)
 match sensitivity [22](#)
 matching records [17](#), [20](#)
 metadata
 adding [25](#)
 attributes [27](#)
 entity type definitions [26](#)
 entity types [96](#)
 inherited [26](#)
 overview [13](#)
 presentation [29](#)
 sample [31](#)
 uses in hub [13](#)

N

new entities, creating [78](#)

P

parent entity type [102](#)
 permissions, DataFlux Data
 Management Server [47](#)
 preferences
 global [54](#)
 SAS Data Management Console [55](#)
 presentation metadata [29](#)
 presentation tier [5](#)
 process jobs [6](#)
 publishing entity type definitions [97](#),
 [98](#)

Q

QKB [5](#), [7](#)
 licensing [17](#)
 QKB, licensing [135](#)

R

read-only attributes [29](#), [105](#)
 real data type [28](#)
 real-time services [5](#), [6](#), [7](#)
 records
 matching [17](#), [20](#)
 records tab [69](#)
 regular expression constraints,
 attribute [105](#)
 relationship diagram tab [74](#)
 relationship type definitions [113](#), [120](#)
 relationship type editor [112](#), [113](#), [121](#)
 relationship types tab, entity type
 editor [112](#)
 relationship types tab, entity types
 editor [113](#), [121](#)
 relationship types, entity [31](#)
 creating [112](#), [113](#), [121](#)
 creating match conditions [119](#), [123](#)
 modifying [112](#), [113](#), [121](#)
 relationships, entity
 overview [14](#)
 remediation issue, creating [80](#)
 required attributes [105](#)
 retiring [30](#), [104](#)
 retiring
 attributes [104](#)
 entity type definitions [96](#)
 required attributes [30](#), [104](#)

S

sample metadata [31](#)
 SAP [10](#)
 SAS Data Management Console
 home page [52](#)
 log on [51](#)
 preferences [55](#)
 SAS MDM
 interface [56](#)
 SAS MDM deployment models [9](#)
 scripting [5](#)
 search results [65](#)
 searching entities [62](#)
 sensitivity, matching [22](#)
 Service Oriented Architecture [5](#)
 Servlet 2.5 [5](#)
 Siebel [10](#)
 SOA [5](#)
 SOAP [5](#)
 source system definitions [129](#)
 source system editor
 tools tab [131](#)

staging tables 39
standardization 5, 12, 16, 27, 153,
154
string data type 28
survivor records 13
survivorship 13, 22, 99

T

text data type 28
TIBCO 10
timestamp data type 28
tools editor 108, 131
tools tab

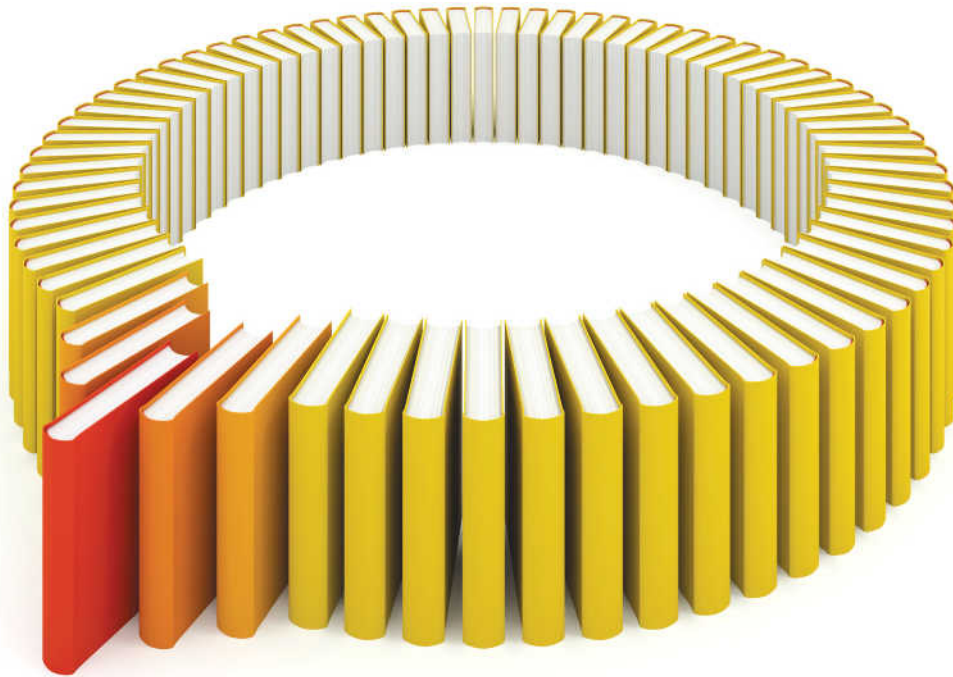
entity type editor 108
source system editor 131
trimming data 39

U

URL data type 28

W

webMethods 10



Gain Greater Insight into Your SAS[®] Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.



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. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613

