

## CHAPTER

## 1

## Using the Metadata LIBNAME Engine

---

<i>What Does the Metadata Engine Do?</i>	3
<i>Understanding How the Metadata Engine Works</i>	4
<i>Advantages of Using the Metadata Engine</i>	5
<i>What Is Supported?</i>	7

---

### What Does the Metadata Engine Do?

The metadata engine enables you to use metadata in order to process and augment data that is identified by the metadata. The metadata engine retrieves information about a target SAS data library from metadata objects in a specified SAS Metadata Repository.

The *metadata* is information about the structure and content of data and the applications that process and manipulate that data. The metadata contains details such as the location of the data and the SAS engine that is used to process the data.

The metadata engine provides a consistent method for accessing many data sources. For example, because SAS provides different engines, with different options, behavior, and tuning requirements, it can be difficult to keep track of how each SAS engine works. By taking advantage of metadata, the necessary information required to access data can be created in one central location so that applications can simply use the metadata engine in order to access different sources of data, without having to understand the differences and details of each SAS engine.

---

## Understanding How the Metadata Engine Works

The metadata engine works much like other SAS engines. That is, you execute a LIBNAME statement in order to assign a libref and to specify an engine. You then use that libref throughout the SAS session where a libref is valid.

However, instead of the libref being associated with the physical location of a SAS data library, the metadata libref is associated with a set of metadata objects. The metadata objects identify the SAS engine that provides access to the data and options that are necessary to process the SAS data library and its members.

Here is an example of a LIBNAME statement for the metadata engine and a description of what happens when you execute the statement:

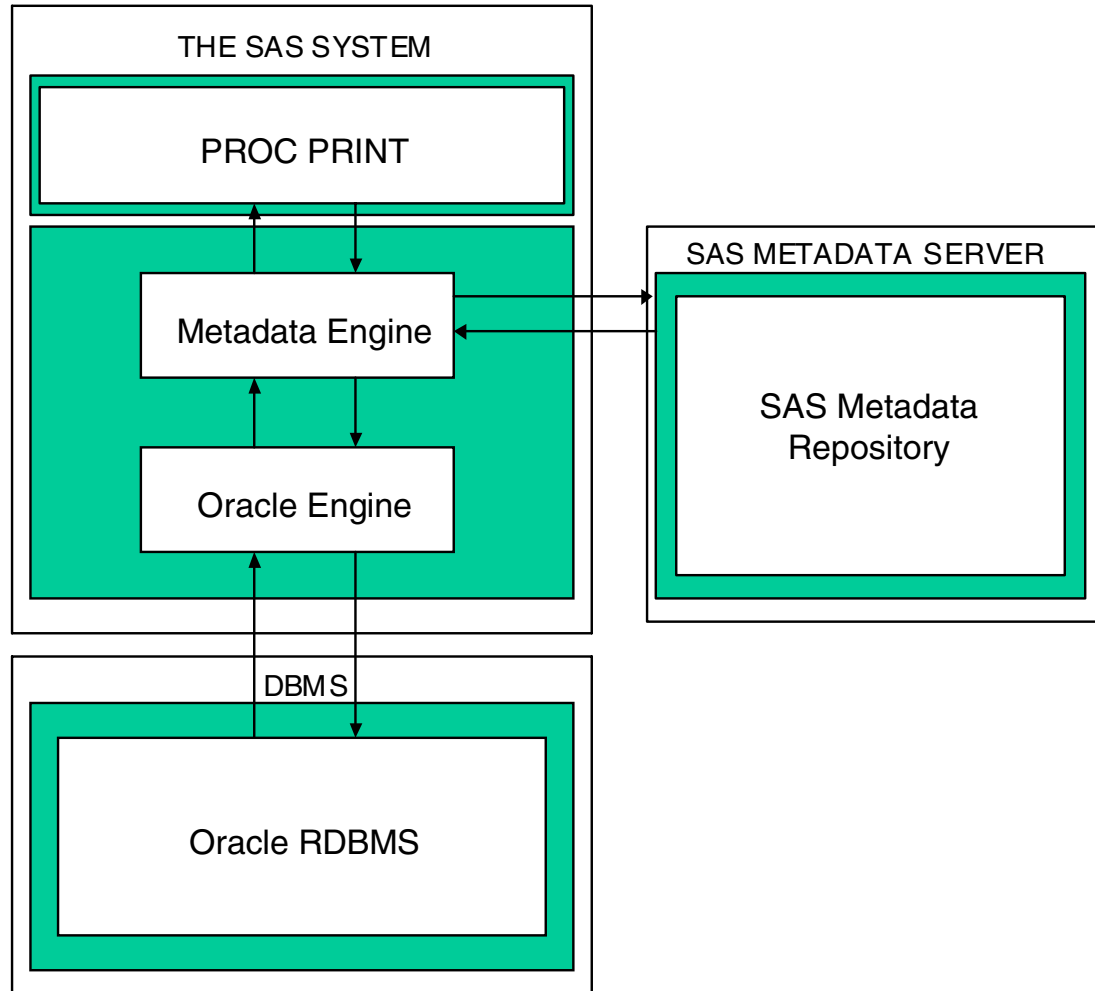
```
libname oralib meta libid=A8000001 repid=AWPKT800
    userid=metaid pw=metapw
    ipaddr=myip.us.org.com port=6401
    protocol=bridge liboptset=myopts;
```

- 1 The metadata engine retrieves information about the target SAS data library from the metadata.
- 2 The metadata engine uses the retrieved information to construct a LIBNAME statement for the engine that is specified in the metadata (referred to as the underlying engine) and assigns it the appropriate options.
- 3 Then, when the metadata engine needs to access data, the metadata engine uses the underlying engine to process the data.

For example, if you have an Oracle data library defined in metadata and you reference the library with the LIBNAME statement for the metadata engine, the metadata engine constructs a LIBNAME statement that assigns the Oracle SAS/ACCESS engine. When the metadata engine needs to process a member of the library, such as with the PRINT procedure, the metadata engine issues a request to the metadata repository for the metadata that is associated with that member and uses the Oracle engine. The metadata includes PhysicalTable, Column, and Property objects.

The following diagram illustrates the metadata engine process.

Figure 1.1 Metadata Engine Process



## Advantages of Using the Metadata Engine

Using the metadata engine provides the following advantages:

- The metadata engine provides a single point of access to many heterogeneous data sources. You do not have to be aware of any of the engine-specific details.
- The metadata engine, in conjunction with the SAS Open Metadata Architecture Authorization facility, enables an administrator to control what data users are allowed to access in three ways:
  - The administrator can associate authorization metadata to any metadata resource in a repository using the SAS Open Metadata Architecture Authorization Facility. The metadata engine enforces the decision returned by the authorization facility for libraries and tables.
  - The administrator can include only those libraries, tables, and columns that users are allowed to access in the metadata. In this way, only those libraries, tables, and columns can be accessed by the metadata engine. For example, if a data source has five tables but only two are defined in the repository, the metadata engine provides access to only those two tables. In another situation, if a table has 20 columns but only five columns are defined in the

repository, the metadata engine provides access to only those five columns. In this context, the metadata engine acts as a filter, providing member- and column-level security.

- An administrator can associate authorization metadata to a metadata resource that prevents the user from accessing the resource in the data source. For example, if a library has 20 tables and all 20 tables are defined in the repository, then the administrator can associate authorization metadata with five tables, which prevents the user from accessing those tables. Therefore, the user will have access to only 15 tables. Again, the metadata engine acts as a filter, providing member-level security.

An advantage of this behavior is that an administrator can use the metadata engine as a means to provide library and table security. See Chapter 7, “Metadata Requirements for Using the SAS Open Metadata Architecture Authorization Facility to Control Data Access,” on page 41.

- The SAS Open Metadata Architecture, which is a general-purpose metadata management facility that provides common metadata services to SAS applications, provides a permanent storage location to maintain column formats, informats, labels, and other information that cannot currently be saved with some data sources.
- Changing the type of data source you use does not require changing SAS code or other applications. SAS Open Metadata Architecture provides a centralized location for storing information about your data environment. Because the metadata engine uses this environment to access your data, you are insulated from these changes. It does not matter if your data changes from Oracle to DB2 to SAS; the SAS code remains the same.

---

## What Is Supported?

The metadata engine supports the following:

- The metadata engine accesses metadata in a SAS Metadata Repository in order to process data that the metadata identifies. Using the metadata engine and SAS code, you can have full read and write access (input, update, output) to tables in the data source.

*Note:* The metadata engine supports only tables (SAS data sets and DBMS tables). The metadata engine does not support other SAS files such as catalogs or views. △

See Chapter 3, “Examples of Using the Metadata Engine,” on page 11.

- The metadata engine in conjunction with the SAS Open Metadata Architecture Authorization facility enables an administrator to control what data users are allowed to access.
- The metadata engine can augment your data. For example, if you execute the DATASETS procedure in order to get a listing of all the members in a library, the metadata engine sends a request to the metadata repository for this information. The metadata engine does not use the underlying engine. You get a listing of only those members that have been populated in the repository. When you execute the CONTENTS procedure, the table and column attributes that are returned are from the repository, not a result of the underlying engine processing the data. Any formats, informats, or labels that are stored in the metadata are applied to the underlying data. Any columns that are not populated in metadata are not available to the user.
- The metadata engine supports passing data set options directly to the underlying engine. LIBNAME statement options and data set options can also be placed in the SAS Metadata Repository to be used by the engine.
- The metadata engine supports SQL implicit pass-through.
- The metadata must be consistent with the identified data source. For example, it is invalid to have metadata for a SAS data set where a variable is defined as numeric data, but the actual data source defines it as character data.

