CHAPTER 1

Understanding the SAS Java Metadata Interface

About This Guide 1
Interface Overview 1
Software Installation and JRE Requirements 2
Understanding the Interface 2
Important Terms 4

About This Guide

This guide provides usage information about the SAS 9.1.3 Java Metadata Interface. Reference information about the interface is provided as class documentation that is shipped with the product. You can view a web-enabled version of the reference documentation at support.sas.com/rnd/gendoc/bi/api/.

Interface Overview

The SAS Java Metadata Interface provides a Java object interface to the SAS Metadata Server. The interface provides a way to access SAS metadata repositories through the use of client Java objects that represent server metadata. The interface consists of classes for

- connecting to the metadata server
- instantiating an object factory that creates Java objects to represent the SAS Metadata Model
- creating, reading, and writing Java metadata object instances on the client and propagating additions and changes to the SAS Metadata Server.

There are two implementations of the SAS Java Metadata Interface:

- a static version for visual applications that use only one Java Virtual Machine. Example applications include SAS ETL Studio and SAS Management Console plugins.
- a remote version for applications that have multiple tiers and use more than one Java Virtual Machine. Example applications include: SAS Report Studio, SAS Information Map Studio, and any application that uses SAS Foundation Services implementations.

The SAS Java Metadata Interface includes the following Java packages:

com.sas.metadata
provides the static Java object interface to the SAS Metadata Server.

com.sas.metadata.impl
provides the implementation of the static interface to the SAS Metadata Server.

com.sas.metadata.remote
provides the remote Java object interface to the SAS Metadata Server.

com.sas.metadata.remote.impl
provides the implementation of the remote interface to the SAS Metadata Server.

The com.sas.metadata.remote packages are typically used in conjunction with com.sas.services.information package included with SAS Foundation Services software. The com.sas.services.information package provides a generic interface for interacting with heterogeneous data repositories, including SAS Metadata Repositories, Lightweight Directory Access Protocol (LDAP) repositories, and WebDAV repositories, from client applications. Using Information Service methods, a client can submit a single query that searches all available repository sources and returns the results in a "smart object" that provides a uniform interface to common data elements. The com.sas.services.information package is described in the SAS Foundation Services class documentation. SAS Foundation Services is a component of SAS Integration Technologies. Both the software and the class documentation are available from the SAS Installation Kit CD-ROM software media.

Software Installation and JRE Requirements

SAS Java Metadata Interface software is supported in UNIX and Windows host environments. You can install the software and class documentation from the SAS Software Installation Kit CD-ROM software media that is shipped with SAS 9.1.

The current release of the Java client software requires Java 2 Standard Edition Version 1.4 (JDK 1.4). The Java Runtime Environment can be obtained from the Third Party Software Components CD included in your SAS Software Installation Kit.

The javadoc can be viewed with Microsoft ® Internet Explorer Web Browser or Netscape ® Web Browser versions that support frames.

Understanding the Interface

The SAS Java Metadata Interface consists of

□ an *object factory* for creating and controlling the lifecycle of objects in the client

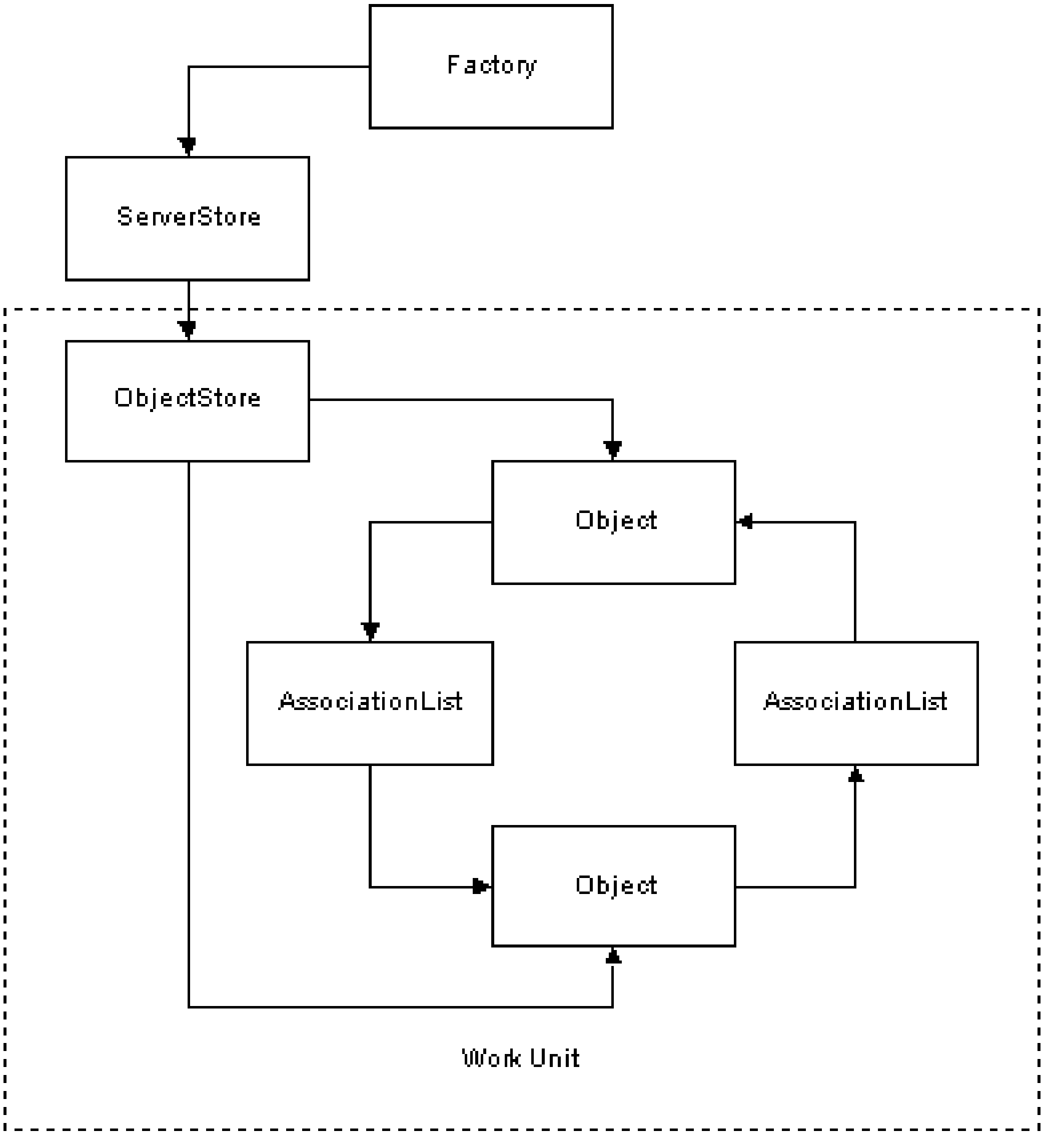
□ a *server store* for managing objects created by the object factory

□ *object stores* that serve as work-unit containers for updating object instances and for grouping object instances that need to be persisted to the server as a unit

□ Java objects for managing a metadata object's properties.

The object factory and server store represent the SAS Metadata Model in the client and provide an environment for managing Java objects that represent SAS metadata object instances.

The object store serves as a container or work unit for Java objects that users create to add metadata objects or to modify existing metadata objects on the metadata server. The following figure illustrates the relationship between the objects in an object store.



A SAS Open Metadata Interface metadata object is defined by two types of properties:

□ a set of attributes that describe the characteristics of the object instance, including its name, description, the date it was created, and any unique characteristics

□ association names that describe its relationships with other metadata objects.

Using the SAS Java Metadata Interface, you get and set a metadata object's attributes by creating a Java object representing its native type. A *native type* refers to one of the metadata types defined in the SAS namespace of the SAS Metadata Model. Information about associations is managed separately from information about attributes. You get and set associations by creating AssociationList objects. An AssociationList object stores information about the metadata objects related to a metadata object via a specific association name. To determine the associations defined for a specific metadata type, consult the "Alphabetical Listing of SAS Namespace Metadata Types" in the *SAS Open Metadata Interface: Reference*. Also see "Understanding Associations" in "Overview of the SAS Metadata Model and Model Documentation" in the reference. The alphabetical listing is available only in online versions of the reference. Look for it in *SAS Help and Documentation* or *SAS OnlineDoc*.

In the previous figure, the squares named ‘Object’ represent native objects and the squares named ’AssociationList’ describe the relationships (associations) between the native objects. Every relationship in the SAS Metadata Model is a two-way association. That is, there are two sides to each relationship and each side has a name. For example, if the native objects in the illustration represented a PhysicalTable and a Column, the PhysicalTable object would have a Columns association to the Column object and the Column object would have a Table association to the PhysicalTable object.

See “Method Classes Summary” on page 17 for an overview of the method classes used to create the factory, stores, and other objects.

See “Overview” on page 5 for instructions about how to write a SAS Java Metadata Interface client that reads and writes metadata.

For documentation describing the classes and methods, see the SAS Java Metadata Interface at support.sas.com/rnd/gendoc/bi/api/.

Important Terms

A *metadata type* is a template that models the metadata for a particular kind of object.

A *metadata object* is an instance of a metadata type, such as the metadata describing a particular table or column.

A *namespace* is a group of related metadata types and their properties. Names are used to partition metadata into different contexts.

In the current SAS release, the SAS Java Metadata Interface provides interfaces for metadata types defined in the SAS namespace of the SAS Open Metadata Interface. The SAS namespace contains metadata types describing the most commonly used application elements. These metadata types are described in “Alphabetical Listing of SAS Namespace Metadata Types.”