



CHAPTER

1

Concepts

<i>Developing Java Clients</i>	1
<i>Java Client Installation and JRE Requirements</i>	2
<i>Client Installation</i>	2
<i>JRE Requirements</i>	2
<i>Java Client Security</i>	3
<i>Using the IOM Server</i>	3
<i>Overview of Using the IOM Server</i>	3
<i>Using a Metadata Server with the Connection Service</i>	4
<i>Connecting a Java Client to an IOM Server</i>	4

Developing Java Clients

The application programming interfaces that are provided with SAS Integration Technologies enable you to develop Java-based distributed applications that are integrated with the SAS platform. SAS Integration Technologies includes the following features for developing Java clients:

- The Connection Factory interface, which enables Java programs to communicate with Integrated Object Model (IOM) servers through an IOM Bridge connection. The connection factory allows you to obtain server attributes from a SAS Metadata Server, from the Information Service (which is part of SAS Foundation Services), or directly from an application program. When it is used with the SAS Open Metadata Architecture, the Java Connection Factory interface provides the following:
 - connections to new types of IOM servers (SAS Metadata Servers, SAS Stored Process Servers, and SAS OLAP Servers) in addition to SAS Workspace Servers
 - the ability to use load balancing for workspace and stored process servers and spawners
- SAS Foundation Services, which extends Java application development beyond access to IOM servers. The following core infrastructure services are provided:
 - client connections to application servers (including the Java Connection Factory interface previously mentioned)
 - dynamic service discovery
 - user authentication and profile management
 - session context management
 - metadata and content repository access

- activity logging

SAS Foundation Services also includes extension services for event management, information publishing, and stored process execution.

- The SAS Foundation Services Facade, which includes convenience services that Web application developers can use to easily access the most commonly used SAS Foundation Services methods and objects.

Use of this software requires some knowledge of distributed programming. However, the software rigorously conforms to Java distributed programming standards such as CORBA and JDBC. Whether you are developing an applet, a stand-alone application, a servlet, or an enterprise JavaBean, you can focus your attention on exploiting the features of the SAS platform rather than determining how to communicate with it.

SAS Integration Technologies supports any Java integrated development environment (IDE), including Eclipse, JBuilder, and SAS AppDev Studio.

Java Client Installation and JRE Requirements

Client Installation

To install the Java client software, you must install SAS Foundation Services.

JRE Requirements

The current release of the Java client software requires the Java Runtime Environment (JRE), Version 5.

To compile and run the code examples included in the Java client development documentation, you must include Java archives in your classpath as specified in the following table.

Table 1.1 Requirements for Code Samples

Samples	Java Archives Needed
All samples	<code>sas.core.jar</code>
	<code>sas.svc.connection.jar</code>
SAS Metadata Server samples	<code>sas.oma.joma.jar</code>
	<code>sas.oma.joma.rmt.jar</code>
	<code>sas.oma.omi.jar</code>
	<code>sas.oma.util.jar</code>
	<code>sas.svc.connection.platform.jar</code>

To run the IDL-to-Java compiler or run the binder utility, you must include `sas.iom.tools.jar` in your classpath.

For help setting your classpath, see the documentation for your Java Runtime Environment.

Java Client Security

For an overview and understanding of security for the SAS Open Metadata Architecture, see the *SAS Intelligence Platform: Security Administration Guide*.

The IOM Bridge for Java has the ability to encrypt all messages exchanged with the IOM server by using a two-tiered security solution. The first tier is a SAS proprietary encryption algorithm. The second tier contains standards-based RC2, RC4, DES, and Triple DES encryption algorithms.

The SAS proprietary encryption algorithm (SASPROPRIETARY) is appropriate to use when you want to prevent accidental exposure of information while it is being transmitted over a network between an IOM Bridge for Java and an IOM server. Access to this encryption algorithm is included with your Base SAS license, and the Java implementation is integrated into the IOM Bridge for Java.

The second-tier encryption algorithms are appropriate to use when you want to prevent exposure of secret information. In other words, using these algorithms makes it extremely difficult to discover the content of messages exchanged between an IOM Bridge for Java and an IOM server. To use these algorithms you must license SAS/SECURE software.

In addition to encryption, SAS/SECURE software also supports message authentication codes (MAC). A MAC is a few bytes of information that is appended to a message to allow the receiver to confirm that the message has not been altered in transit.

Instructions for the security features of the IOM Bridge for Java are included with the documentation for the `com.sas.services.connection` class. Those instructions contain some tips on how to configure the IOM server, but more complete information is available in the documentation for Base SAS software. Installation instructions and usage information for second-tier encryption algorithms is provided in the documentation for SAS/SECURE software.

Using the IOM Server

Overview of Using the IOM Server

This section introduces the steps necessary to construct and execute a Java application that uses the IOM server. As you become more familiar with Java client programming for the IOM server, you can build on these steps to exploit the more sophisticated features of the IOM server.

- a SAS Metadata Server.
- server parameters supplied directly in the source code. (You can supply a `ManualConnectionFactoryConfiguration` object directly in the source code. For details, see “Connecting with Directly Supplied Server Attributes” on page 8).

The Connection Service can connect to SAS Workspace Servers, other metadata servers, SAS OLAP Servers, and SAS Stored Process Servers.

Using a Metadata Server with the Connection Service

If you are using a metadata server, the first step in developing and running a client program is to make sure you have access to a properly configured server. You can access a server by reading the connection information from a SAS Metadata Server.

As is the case in client development, you can start with a basic server configuration and then move into more a sophisticated configuration over time.

After the IOM server has been configured, you can begin developing a Java client for the IOM server.

Connecting a Java Client to an IOM Server

Java clients can use the Java Connection Factory interface to access an IOM server by performing the following steps:

- 1 From the Java Connection Factory, obtain a connection to an IOM server. Then, obtain the remote object reference connected to that IOM server and narrow it to the appropriate remote interface.
- 2 Use Java CORBA stubs for IOM objects and JDBC connection objects to exploit the power of SAS in the IOM server.
- 3 Return the connection to the Java Connection Factory for disconnection or reuse.

To get started, you can put together a simple client application by composing the examples given for each step. Then you can continue to read the additional documentation and learn about Java client programming for the IOM server in greater detail.