SAS® 9.2
Integration Technologies
Overview

Contents

What's New

Overview

New features for SAS Integration Technologies in SAS 9.2 include a number of general product enhancements and documentation enhancements.

General Product Enhancements

The following enhancements have been added to SAS Integration Technologies:

□ SAS Stored Processes provides enhancements to the SAS Stored Process Web Application and to stored process metadata, as well as several general enhancements.

□ SAS 9.2 BI Web Services provide generated Web services, as well as general enhancements.

□ The Publishing Framework provides enhancements to security and to the support of the HTTPS protocol and IPV6 addresses.

□ The following software components are obsolete:

□ Integration Technologies Administrator (IT Administrator)

□ SAS Publisher

□ SAS Subscription Manager

□ The Lightweight Directory Access Portal (LDAP) is no longer supported as a metadata store for SAS applications. For example, using LDAP to configure publication channels, Integrated Object Model (IOM) servers, and SAS Information Delivery Portal is no longer supported. You must use the SAS Metadata Server. The SAS Metadata Server can still use LDAP as an authentication provider, and LDAP connection from SAS is still supported for data loading and extraction.

Documentation Enhancements

The following enhancements have been made to the SAS Integration Technologies documentation:

☐ The *SAS Integration Technologies: Developer's Guide* has been divided into the following books, which are available in the SAS online documentation, as well as in PDF form:

  ☐ *SAS Stored Processes: Developer's Guide*

  ☐ *SAS BI Web Services: Developer's Guide*

  ☐ *SAS Publishing Framework: Developer's Guide*

  ☐ *Application Messaging with SAS*

  ☐ *SAS Integration Technologies: Java Client Developer's Guide*

  ☐ *SAS Integration Technologies: Windows Client Developer's Guide*

  ☐ *SAS Integration Technologies: Directory Services Reference*

☐ The *SAS Integration Technologies: Administrator's Guide* has been divided. SAS Stored Process and Publishing Framework topics have been incorporated into the *SAS Stored Processes: Developer's Guide* and *SAS Publishing Framework: Developer's Guide*, respectively. SAS Foundation Services topics are covered in the *SAS Foundation Services: Administrator's Guide*.

☐ Information from the *SAS Integration Technologies: Server Administrator's Guide* has been incorporated into the *SAS Intelligence Platform: Application Server Administration Guide*.

☐ The *SAS Integration Technologies: Administrator's Guide (LDAP Version)* has been discontinued.

C H A P T E R

1
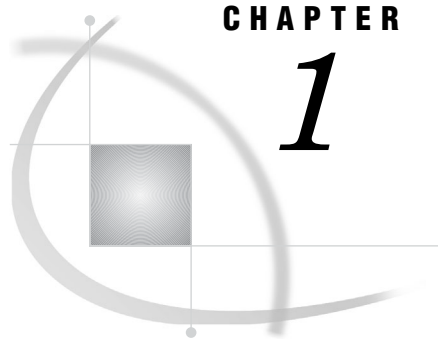
Overview of SAS Integration Technologies

What is SAS Integration Technologies?

SAS Integration Technologies, in combination with other SAS software and solutions, enables you to make information delivery and decision support a part of the information technology architecture for your enterprise.

SAS Integration Technologies provides you with software that enables you to build a secure client/server infrastructure on which to implement SAS distributed processing solutions. With SAS Integration Technologies, you can integrate SAS with other applications in your enterprise; provide proactive delivery of information from SAS throughout the enterprise; extend the capabilities of SAS to meet your organization's specific needs; and develop your own distributed applications that leverage the analytic and reporting powers of SAS.

SAS Integration Technologies is part of the SAS Intelligence Platform. For an overview of the SAS Intelligence Platform and the products it contains, see the *SAS Intelligence Platform: Overview*.

Accessibility Features of SAS Integration Technologies

This product has not been tested for compliance with U.S. Section 508 standards. If you have specific questions about the accessibility of SAS products, send them to **accessibility@sas.com** or call SAS Technical Support.

What SAS Integration Technologies Includes

SAS Integration Technologies includes the following integration and system development tools, which are based on a combination of industry-standard technologies and technology developed by SAS:

☐ the Integrated Object Model (IOM), which provides distributed object interfaces to SAS software features. IOM enables you to use industry-standard languages,

programming tools, and communication protocols to develop client programs that access these services on IOM servers. The IOM Bridge communications protocol enables diverse clients to connect transparently to IOM servers on multiple platforms.

□ two types of IOM servers: the SAS Workspace Server, which surfaces the SAS programming environment to calling clients through an application programming interface (API); and the SAS Stored Process Server, which enables clients to execute parameterized SAS programs without having to know the SAS language.

   Note: Other types of IOM servers include the SAS Metadata Server, which is provided with Foundation SAS, and the SAS OLAP Server, which is provided with SAS Intelligent Storage products. △

□ SAS Foundation Services, which is a set of core infrastructure services that Java programmers can use to write distributed applications that are integrated with the SAS platform. The services provide client connections to IOM servers, dynamic service discovery, user authentication, profile management, session context management, metadata and content repository access, activity logging, event management, information publishing, and stored process execution.

□ the Publishing Framework, which consists of SAS CALL routines and graphical user interfaces that enable you to publish information and events proactively by using a subscription channel model.

□ the Application Messaging interface, which provides three sets of CALL routines that enable you to incorporate messaging services into your SAS programs.

□ SAS Stored Processes, which enable client applications to execute SAS programs that are stored centrally on a server.

□ SAS BI Web Services, which enable client applications to send XML requests and parameters in a SOAP envelope. Web services can discover and execute SAS Stored Processes.

□ the SAS Web Infrastructure Platform, which enables you to develop Web applications and components using portal technology.

□ the Directory Services interface, which enables you to incorporate LDAP directory services functions into your SAS programs.

Support for SAS Open Metadata Architecture

SAS Integration Technologies supports the SAS Open Metadata Architecture, which is a metadata management facility that provides common metadata services to SAS applications. This architecture is required in order to do the following tasks:

□ use the SAS Metadata Server to store configuration information for SAS Integration Technologies and other SAS products

□ use SAS Management Console as a central interface to administer configuration and security information for SAS Integration Technologies and other SAS products

□ use new security features for user registration, authentication, and authorization (access control)

□ configure new types of IOM servers (SAS Metadata Servers, SAS Stored Process Servers, and SAS OLAP Servers) in addition to SAS Workspace Servers

□ use load balancing for workspace servers and spawners, as well as stored process servers and spawners

□ operate stored processes on a stored process server to produce streaming output for use in Web applications

□ use the SAS Web Infrastructure Platform to create Web applications that use portal technology

□ use SAS Foundation Services to implement Java applications that are integrated with the SAS Intelligence Platform

□ install and operate other SAS products that depend on the SAS Intelligence Platform

For more information about the SAS Open Metadata Architecture, see the *SAS Intelligence Platform: System Administration Guide*.

Support for Industry Standards

SAS Integration Technologies supports the following industry standard technologies, allowing you to leverage your existing infrastructure investments and skill sets to provide application interoperability:

□ client development by using any programming environment that supports bindings to the COM and DCOM or Java object model. The programming environments that are supported include the following:

□ Microsoft Visual Basic for Applications, VBScript, Visual C++ .NET, Visual Basic .NET, and Visual C# .NET

□ any Java integrated development environment (IDE), including Eclipse, Sun ONE Studio, IBM VisualAge, Borland JBuilder, and SAS webAF (which is part of SAS AppDev Studio)

□ proprietary Windows programming environments such as Borland Delphi, Sybase PowerBuilder, and others

□ asynchronous message queuing through the use of IBM WebSphere MQ (formerly called MQSeries), Microsoft MSMQ, and TIBCO Rendezvous.

□ file management that uses Web-based Distributed Authoring and Versioning (WebDAV). The publish and subscribe features of SAS Integration Technologies can access any WebDAV server. Two specific types of WebDAV servers are supported for use by SAS business intelligence Web applications:

□ Xythos Software's WebFile Server (WFS) can be used to store content for the SAS Web Infrastructure Kit and the SAS Information Delivery Portal. It can also be used to store reports that have been created with SAS Web Report Studio. The SAS User Management Customization enables a Xythos WebFile Server to interact with the SAS Metadata Server for authorization and authentication.

□ Apache HTTP Server (with its WebDAV module enabled) can also be used to store reports that have been created with SAS Web Report Studio.

□ Web services development based on the XML for Analysis interface.

□ Web application development on Web servers that are compatible with the Java 2 platform.
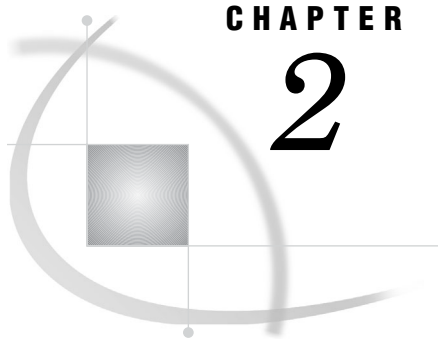
CHAPTER

2

Components of SAS Integration Technologies

IOM Servers

Overview of IOM Servers

The Integrated Object Model (IOM) in SAS Integration Technologies provides distributed object interfaces to SAS software features. To call these interfaces, clients can use industry-standard languages, programming tools, and communication protocols. The interfaces are built into SAS and are available to clients whenever SAS is executed as an object server.

An IOM server is an object server that is launched in order to fulfill client requests for IOM services. There are four types of IOM servers:

SAS Workspace Server	is provided with SAS Integration Technologies and is accessed through the IOM workspace interface. This interface provides access to Foundation SAS software features such as the SAS language, SAS libraries, the server file system, results content, and formatting services. A SAS workspace represents a session with the SAS System, and it is functionally equivalent to a SAS Display Manager session or the execution of the SAS System as a batch job.
SAS Stored Process Server	is a multi-user server that is provided with SAS Integration Technologies and is used to execute SAS Stored Processes and deliver the results. A SAS Stored Process is a SAS program that is stored on a server and can be called by client applications. SAS Stored Processes enable clients to execute parameterized SAS programs without having to know the SAS language.
SAS OLAP Server	delivers pre-summarized cubes of data to OLAP clients like SAS Enterprise Guide using OLE DB for OLAP. *Cubes* are logical sets of data that are organized and structured in a hierarchical, multidimensional arrangement. You can query cubes by using the multidimensional expression (MDX) language. The SAS OLAP Server, which is provided with SAS Intelligent Storage products, is designed to reduce the load on traditional back-end storage systems by delivering summarized views of data to business intelligence applications, irrespective of the amount of data underlying these summaries.
SAS Metadata Server	is a multi-user server, provided with SAS Foundation, that enables users to read metadata from or write metadata to one or more SAS Metadata Repositories. SAS Metadata Repositories contain metadata that represents items such as SAS application servers (including SAS Workspace Servers, SAS OLAP Servers, and SAS Stored Process Servers), users in the metadata environment, libraries, tables, stored processes, and cubes.

Configuring and Using IOM Servers

You can use the SAS Deployment Wizard to plan, install, and define the configurations for your IOM servers. Detailed documentation is provided to help you perform these tasks either with or without the wizard. Documentation is also provided to help you manage and administer your server configurations after installation, and to assist you in developing application programs that use the IOM servers.

The following table provides information about the documentation that is available for IOM servers:

Table 2.1   Documentation for IOM Servers

Server Type	Type of Information	Documentation
all IOM server types	using SAS Deployment Wizard to plan and configure	*SAS Intelligence Platform: Installation and Configuration Guide*
	securing	*SAS Intelligence Platform: Security Administration Guide*
	starting	*SAS Intelligence Platform: System Administration Guide*
	developing applications	"IOM Scenario" on page 10 in this overview
	connecting a client application	"Connecting Clients to IOM Servers" on page 8 in this overview *SAS Integration Technologies: Java Client Developer's Guide* *SAS Integration Technologies: Windows Client Developer's Guide*
	implementing distributed security for clients	*SAS Integration Technologies: Java Client Developer's Guide* *SAS Integration Technologies: Windows Client Developer's Guide*
SAS Workspace Server	planning and configuring without SAS Deployment Wizard	*SAS Intelligence Platform: Application Server Administration Guide*
	modifying and administering the configuration after installation	*SAS Intelligence Platform: Application Server Administration Guide*
	modifying the configuration to improve the performance of SAS business intelligence applications	*SAS Intelligence Platform: Application Server Administration Guide*
	developing clients	"Workspace Object Hierarchy" on page 11 in this overview *SAS Integration Technologies: Java Client Developer's Guide* *SAS Integration Technologies: Windows Client Developer's Guide*

Server Type	Type of Information	Documentation
SAS Stored Process Server	planning and configuring without SAS Deployment Wizard	*SAS Intelligence Platform: Application Server Administration Guide*
	modifying and administering the configuration after installation	*SAS Intelligence Platform: Application Server Administration Guide*
	modifying the configuration to improve the performance of SAS business intelligence applications	*SAS Intelligence Platform: Application Server Administration Guide*
	developing clients	*SAS Stored Processes: Developer's Guide* *SAS Integration Technologies: Java Client Developer's Guide*
SAS OLAP Server	modifying and administering the configuration after installation	*SAS Intelligence Platform: Application Server Administration Guide*
SAS Metadata Server	modifying and administering the configuration after installation	*SAS Intelligence Platform: System Administration Guide*
	developing clients	*SAS Open Metadata Interface: Reference* *SAS Java Metadata Interface: User's Guide*

Connecting Clients to IOM Servers

Overview of Communication between Clients and IOM Servers

The information that is needed to connect to an IOM server is obtained from the SAS Metadata Server. SAS Integration Technologies supports additional SAS software features by enabling Java and Windows clients to connect to and request objects from various types of IOM servers including other SAS Metadata Servers, SAS OLAP Servers, SAS Stored Process Servers, and SAS Workspace Servers.

SAS Integration Technologies facilitates client/server communication across multiple vendor architectures by supporting multiple client/server interoperability standards. The following figure shows the different ways that clients and IOM servers can communicate.

Figure 2.1 Communication between Clients and IOM Servers

SAS as IOM Bridge client

SAS batch
SAS -dms
SAS -objectserver
etc.

Java client

Java

COM client
(Windows)

C#
VB.NET
Visual Basic 6
VBA
VBScript
Visual C++
etc.

SAS -objectserver   protocol=bridge
(all supported platforms)

SAS -objectserver   protocol=com
(Windows)

SAS as COM client
(Windows)

SAS batch
SAS -dms
SAS -objectserver
etc.

Connection Factory   IOM Bridge for Java   IOM Bridge for COM   Object Manager

— — — — SAS/IOMP   ——————— COM/DCOM

Windows Clients

As shown in the figure, Windows clients that are written in languages such as Visual
Basic or Visual C++ access IOM servers that are running in the Windows operating
environment by using the Microsoft Component Object Model (COM). Windows client
interfaces provided in SAS Integration Technologies support two leading industry
standards: ActiveX Data Objects (ADO) and Object Linked Embedding for Databases
(OLE DB).

Windows clients accessing IOM servers on other server platforms use the IOM
Bridge for COM. This bridge allows you to develop native COM and DCOM applications
that access server data, for example, on UNIX and mainframe platforms. This
transparency is a key feature of SAS Integration Technologies. It enables application
developers to have full access to the architectural elements that are available in the
Windows environment, even when their clients communicate with servers in other

operating environments. The bridge makes other operating environments appear to be extensions of the client's native operating environment.

With SAS Integration Technologies, Windows clients can use the Windows Object Manager to connect to IOM servers.

Note: In SAS®9, SAS Integration Technologies still supports the SAS 8 interface for using the Windows Workspace Manager. However, it is recommended that you use the Windows Object Manager interface in order to take advantage of the new features. △

Java Clients

Java clients that access IOM servers use the Common Object Request Broker Architecture (CORBA). CORBA is an architecture for an open software bus on which objects can interoperate across networks and operating systems.

The Object Request Broker (ORB) is the key element of CORBA. It provides the infrastructure for distributed object computing by enabling software components (objects) on one machine to locate and communicate with components on other machines. This infrastructure enables you, as an application developer, to focus on implementing your business logic instead of worrying about all the underlying technology.

SAS Integration Technologies provides an ORB called the IOM bridge for Java. This ORB implements the standard CORBA ORB interface defined by the Object Management Group (OMG). It uses a proprietary inter-orb protocol (called the IOM Bridge protocol) to communicate with the IOM server.

With SAS Integration Technologies, Java clients can use the Java Connection Factory to connect to IOM servers.

Note: In SAS®9, SAS Integration Technologies still supports the SAS 8 interface for using the Java Workspace Factory. However, it is recommended that you use the Connection Factory interface in order to take advantage of the new features. △

Using IOM to Develop Applications

IOM Scenario

One of the chief benefits of building component-based applications by using the Integrated Object Model (IOM) is that it enables you to de-couple your business application logic from your presentation methods. This enables you to reuse core application functionality in multiple presentation personalities and, as a result, it extends your application's deployment options.

For example, consider the following figure. Using SAS Integration Technologies software, you can develop applications in which Web browsers, desktop GUIs, and middle-tier Web and application server presentation platforms can all become clients of SAS software server components.

Figure 2.2   Clients of SAS Software Server Components
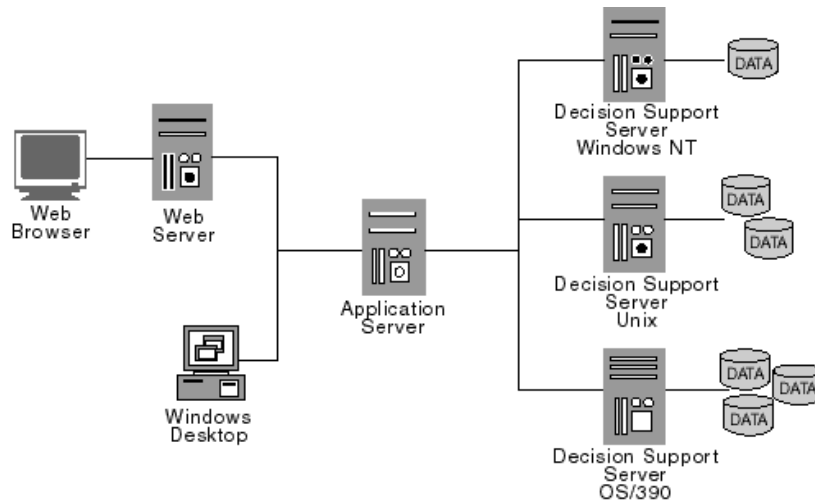
Within the Web browser context, you can use client-side scripting such as VBScript, local components such as ActiveX controls, and Java applets to implement access to server components.

Similarly, you can assess server components from desktop application environments including custom applications that are written in Visual Basic, Java, C++, Delphi, and PowerBuilder. Further, any desktop application container that supports Visual Basic for Applications (VBA), including those in the Microsoft Office suite and many third-party offerings, can also participate.

Finally, you can deploy middle-tier Web and application servers, such as the Microsoft Internet Information Server (IIS), Microsoft Transaction Server (MTS), and various .NET servers to host logic that accesses SAS software server components.

This level of deployment flexibility enables you to distribute your logic components across the enterprise in order to realize the performance advantages that are associated with data source co-location. This component distribution can be accomplished in a manner that is transparent to the client.

Workspace Object Hierarchy

The principal IOM interfaces of the SAS Workspace Server include the following:

- □ Workspace (a SAS session)
- □ LanguageService (submit, list/log, stored processes)
- □ DataService (librefs, plus access through ADO, OLE-DB or JDBC)
- □ FileService (filerefs and directories)
- □ Utilities (result packages, formats, host info)

The following figure depicts the object hierarchy for these interfaces.

Figure 2.3 Workspace Object Hierarchy
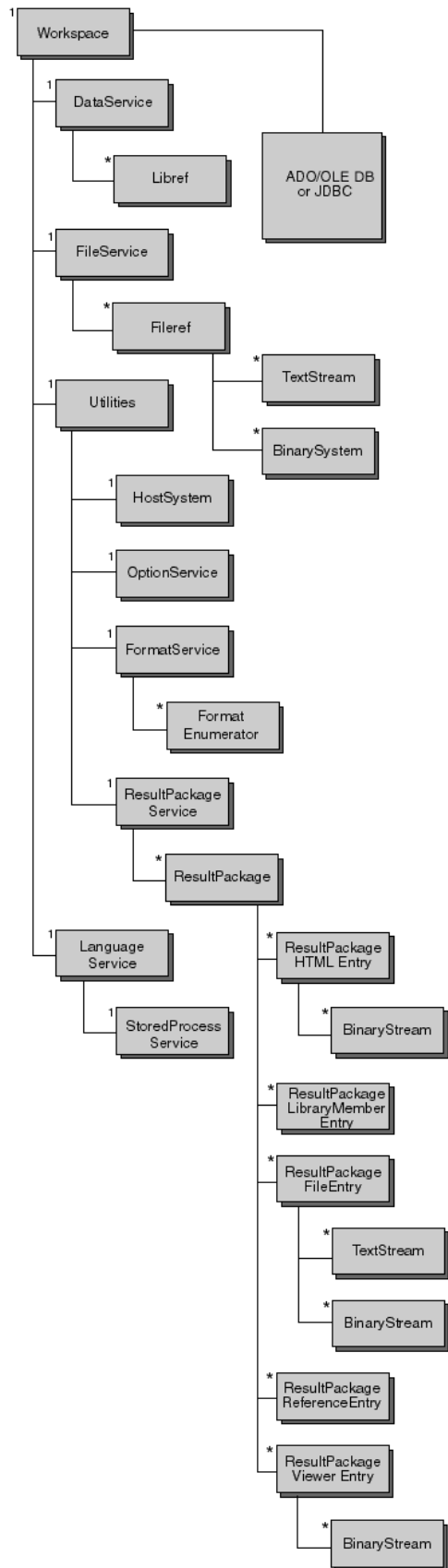
In the figure, the number 1 indicates that that part of the hierarchy has exactly one child per parent. An asterisk (*) indicates that that part of the hierarchy has zero or greater children per parent.

The LanguageService defines methods for submitting SAS procedural scripting language statements (to execute SAS DATA and procedure steps) and for retrieving the associated LOG and LIST outputs.

Applications that produce sophisticated results such as HTML and graphs generated by using the SAS Output Delivery System can use the ResultPackageService to retrieve collected elements. Program execution progress events such as step begin, step end, and error conditions can also be monitored with a LanguageService event interface.

The FileService defines methods for managing SAS file references (FILEREFs) and reading or writing files on the server's host file system. A client can exploit SAS external file access methods and host-specific fileref assignment options to access the wide variety of specialized features in SAS external file I/O on the various server host platforms. The DataService surfaces similar control over SAS library references (LIBREFs).

IOM uses standard interface mechanisms within the Microsoft COM and Java application environments to expose the object hierarchy. This enables you to use the ActiveX Data Objects (ADO) and OLE DB access protocols in the Windows COM environment. In Java environments, the JDBC 2.0 access protocol is supported. These standard mechanisms provide semantic richness, including read, write, update, and query services. All components that support the same standards can easily interoperate.

Application hierarchies are also available through specialized SAS product offerings and custom applications developed with the SAS/AF software.

SAS Integration Technologies provides the middleware necessary to connect clients and servers across multiple vender architectures, as described in "Connecting Clients to IOM Servers" on page 8.

Foundation Services

The SAS Foundation Services are a set of infrastructure and extension services which support the development of integrated, scalable, and secure Java applications. The SAS Foundation Services are based on the following design principles:

- □ implementation modularity
- □ location transparency
- □ robust and adaptive resource management
- □ runtime monitoring
- □ consistent deployment methodology
- □ client neutrality

The design model of the SAS Foundation Services supports both local and remote resource deployment and promotes resource sharing among applications. Sharing can occur for a specific session, for a specific user, or globally, as appropriate. At the same time, the model controls access to protected resources based on privileged-user status and group membership.

The SAS Foundation Services are as follows:

- □ Connection Service
- □ Discovery Service

□ Event Broker Service

□ Information Service

□ Logging Service

□ Publish Service

□ Security Service

□ Session Service

□ Stored Process Service

□ User Service

The Connection, Publish, and Stored Process services are extensions of similar services that are part of the Integrated Object Model (IOM) Services published with SAS Integration Technologies 8.2. The IOM Services continue to be supported. However, the SAS Foundation Services provide enhanced features, including support for the use of a SAS Metadata Server for storing configuration information, and support for the use of shared remote service deployments.

For information about configuring and administering SAS Foundation Services, see the *SAS Foundation Services: Administrator's Guide*. For information about using SAS Foundation Services to develop applications, see the *SAS Integration Technologies: Java Client Developer's Guide*.

Publishing Framework

Overview of the Publishing Framework

The Publishing Framework feature of SAS Integration Technologies provides a complete and robust publishing environment for enterprise-wide information delivery. The Publishing Framework consists of SAS CALL routines, application programming interfaces (APIs), and graphical user interfaces that enable both users and applications to publish SAS files (including data sets, catalogs, and database views), other digital content, and system-generated events to a variety of destinations:

□ e-mail addresses

□ message queues

□ publication channels and subscribers

□ WebDAV-compliant servers

□ archive locations

The Publishing Framework also provides tools that enable both users and applications to receive and process published information. For example, users can receive packages with content, such as charts and graphs, that is ready for viewing. In addition, SAS programs can receive packages with SAS data sets that might in turn trigger additional analyses on that data.

The functions of the Publishing Framework include channel definition, subscription management, package publishing, package retrieval, package viewing, and event publishing.

For information about how to perform publishing tasks, or how to incorporate publishing tasks into your SAS programs or applications, see the *SAS Publishing Framework: Developer's Guide*. For information about how to administer publication

channels and subscriber information, see the product help for the Publishing Framework plug-in in SAS Management Console.

Channel Definition

The Publishing Framework enables you to define SAS publication channels, which are conduits for publishing particular categories of information. You can set up a channel for a particular topic, organizational group, user audience, or any other category.

To define a channel, use SAS Management Console to define a name for the channel, a description, a subject, the path in which the channel's archived packages are to be stored, and one or more key words. The channel definition, or metadata, is stored on the SAS Metadata Server. SAS Management Console can also be used to specify which groups of users are authorized to subscribe to the channel.

Subscription Management

Once publication channels have been defined, authorized users can subscribe to them and automatically receive information whenever it is published to those channels.

First, you must define the users who are eligible subscribers. To do so, use SAS Management Console to enter user authentication information.

If your organization has installed the SAS Information Delivery Portal, users can manage their subscriptions from within the portal. The portal enables users to select channels to subscribe to channels, specify the desired delivery transport (such as an e-mail address or message queue), and specify filters that indicate which information is to be included or excluded.

Administrators can also use the Publishing Framework plug-in for SAS Management Console to manage subscriptions. Using this application, administrators can create groups of subscribers, subscribe individual users or groups to a channel, and specify delivery transport and filtering options.

Package Publishing

The Publishing Framework enables you to create packages that contain one or more information entities, including SAS data sets, SAS catalogs, SAS databases, and almost any other type of digital content. You can also define viewers that make the information entities easier to display.

After creating a package, you can publish the package and its associated viewers to one or more channels. Publishing the package causes the information to be delivered to each user who has subscribed to those channels if the package and its contents meet the subscriber's filtering criteria. In addition to channels, you can publish packages directly to one or more e-mail addresses, message queues, WebDAV-compliant servers, and archive locations.

To create and publish packages, you can use any of the following methods:

☐ use the publish CALL routines to create packages and publish them from within a SAS program

☐ use the APIs that are provided with SAS Integration Technologies to create packages and publish them from within a third-party application

You can also use SAS Enterprise Guide or the SAS Information Delivery Portal to create and publish packages via the Publishing Framework.

Package Retrieval and Viewing

The Publishing Framework provides SAS Package Retriever, which is a graphical user interface that enables users to extract and save information from packages that have been published through the Publishing Framework. The SAS Package Reader user interface enables users to display the contents of packages. If the SAS Information Delivery Portal has been installed, users can view published information from the portal.

In addition, you can use CALL routines to extract and process published information from within SAS programs. APIs are provided to enable third-party programs to extract and process published information.

Event Publishing

SAS Integration Technologies includes the Publish Event Interface. This interface consists of CALL routines that enable you to write SAS programs, including stored processes, that create and publish events. Events can be generated and published explicitly, or they can be generated implicitly as part of the publication of a package. Implicit event generation occurs when packages are published to a channel that has event subscribers defined.

Events are published as well-formed XML documents. They can be published to HTTP servers, to message queues, or to publication channels and their associated subscribers. You can develop custom applications that receive and process events that are generated by the Publishing Framework.

Application Messaging

Overview of Application Messaging

The Application Messaging Interface includes three sets of CALL routines that enable you to incorporate messaging services into your SAS programs. The interfaces support asynchronous message queuing through the use of the following messaging software: IBM WebSphere MQ (formerly called MQSeries); Microsoft Message Queuing Services (MSMQ), which are part of the Windows NT® Server product; and TIBCO TIB/Rendezvous.

Application messaging enables two or more applications to communicate with one another indirectly and asynchronously using message queues. The applications do not have to be running at the same time or even in the same operating environment. An application can communicate with another application by sending a message to a queue. The receiving application retrieves the message when it is ready.

SAS Integration Technologies provides three messaging interfaces: the WebSphere MQ Functional Interface, the MSMQ Functional Interface, and the Common Messaging Interface.

For complete documentation of the SAS application messaging interfaces, including CALL routine syntax and program samples, see the *Application Messaging with SAS*.

WebSphere MQ Functional Interface

With the SAS interface to IBM WebSphere MQ (formerly called MQSeries), SAS programs can create new WebSphere MQ message queues or take advantage of existing ones that are available throughout the enterprise. This interface was designed to be as similar as possible to the WebSphere message queuing interface (MQI).

MSMQ Functional Interface

With the SAS interface to Microsoft MSMQ, SAS programs can create new MSMQ message queues or take advantage of existing ones that are available throughout the enterprise. The CALL routines in this interface interact directly with the MSMQ application programming interface (API).

Common Messaging Interface

The SAS Common Messaging Interface provides a seamless environment for writing SAS programs that access message queues of the IBM WebSphere MQ, Microsoft MSMQ, and TIBCO TIB/Rendezvous transports. The CALL routines in this interface enable programs to interact in a consistent manner that is independent of the application messaging transport being used.

The Common Messaging Interface also includes CALL routines that enable your SAS program to use the local SAS registry or a distributed LDAP repository to store and retrieve messaging information.

Stored Processes

A stored process is a SAS program that is stored centrally on a server. A client application can then execute the program and receive and process the results. Stored processes enable you to centrally maintain and manage code, give you better control over changes, enhance security and application integrity, and ensure that every client executes the latest version of code that is available.

Stored processes are like other SAS programs except they have an additional feature that enables customization of the program's execution. This feature enables the invoking application to supply parameters at the time that the stored process is invoked. For example, if you have a stored process that analyzes monthly sales data, you could create a MONTH variable in the stored process. At execution time, the user could supply the parameter MONTH=MAY to analyze May sales data.

You can use stored processes for Web reporting, analytics, building Web applications, delivering packages to clients or the middle tier, and publishing results to channels or repositories. Stored processes can access any SAS data source or external file and can create new data sets, files, or other data targets supported by the SAS System.

You can use the following clients to invoke stored processes:

□ SAS Add-In for Microsoft Office
□ SAS BI Web Services
□ SAS Data Integration Studio
□ SAS Enterprise Guide
□ SAS Information Delivery Portal

□ SAS Information Map Studio

□ SAS Stored Process Web Application

□ SAS Web Report Studio

□ Stored Process Service application programming interface (API)

Stored processes can generate output files in a variety of formats. They can also produce streaming output for Web applications, as well as package output for publishing. Stored processes can operate either on a SAS Workspace (IOM) Server or a SAS Stored Process Server.

For information about how to create a stored process and invoke it in a client application, see the *SAS Stored Processes: Developer's Guide.* For information about how to create the metadata to define a stored process, see the product help for the New Stored Process Wizard or the Stored Process Properties dialog box in SAS Management Console.

SAS BI Web Services

SAS BI Web Services enable client applications to obtain metadata about and execute SAS Stored Processes by using the XML for Analysis (XMLA) Version 1.1 interface.

A Web service is an interface that enables communication between distributed applications. Web services provide cross-platform integration by enabling applications written in various programming languages to communicate by using a standard Web-based protocol. This functionality makes it possible for businesses to bridge the gaps between different applications and systems.

SAS Integration Technologies includes two implementations of SAS BI Web Services: a Java implementation, which requires a servlet container, and an implementation that uses the .NET framework. These implementations use the Simple Object Access Protocol (SOAP) communication protocol.

Before SAS 9.2, you could write only XMLA Web services. Starting with SAS 9.2, you can use SAS Management Console to deploy a set of stored processes as a generated Web service.

With XMLA Web Services, two methods, Discover() and Execute(), are provided. The Discover method calls the SAS Metadata Server to obtain the requested metadata, and the Execute method calls the SAS Stored Process Server to invoke a stored process. Stored processes that are called by these methods must be registered on the SAS Metadata Server with **XMLA Web Service** as a keyword and **Streaming** as the output type. The output from a stored process that is called by a SAS BI Web Service is always an XML stream.

To create generated Web services, you can select a set of stored processes in SAS Management Console and deploy them to the Web Service Maker. The Web Service Maker generates a new Web service that contains one operation for each stored process that you selected. A new Web Service Description Language (WSDL) document is generated when you deploy the Web service. The generated WSDL document contains all the information that would be obtained if you were using the Discover method.

For detailed information about using SAS BI Web Services, including instructions for creating stored processes that Web services can access, see the *SAS BI Web Services: Developer's Guide.*

Directory Services

If your organization uses an enterprise directory that conforms to the Lightweight Directory Access Protocol (LDAP), you can use the directory services interfaces of SAS Integration Technologies to incorporate enterprise directory services into your SAS applications. Through these interfaces, distributed SAS applications can share a common enterprise directory with components that might be executing in other run-time environments across the enterprise.

SAS Integration Technologies provides two interfaces that you can use to interact with an LDAP server from a SAS program:

- □ The LDAP CALL Routine Interface, which is set of CALL routines that you can use to add, delete, modify, and search entries in the LDAP server.

- □ The LDAPSERVICES class for the SAS Component Language (SCL). This class provides methods that add, delete, modify, and search entries in the LDAP server.

For documentation about the syntax and use of these interfaces, refer to the *SAS Integration Technologies: Directory Services Reference*.

SAS Web Infrastructure Platform

The SAS Web Infrastructure Platform is a collection of services and applications that provide common infrastructure and integration features to be used by SAS Web applications. These services and applications provide the following benefits:

- □ consistency in installation, configuration, and administration tasks for Web applications

- □ greater consistency in users' interactions with Web applications

- □ integration among Web applications as a result of the ability to share common resources

The following services and applications are included in the SAS Web Infrastructure Platform:

SAS BI Web Services for Java
> can be used to enable your custom applications to invoke and obtain metadata about SAS Stored Processes. Web services enable distributed applications that are written in different programming languages and that run on different operating systems to communicate using standard Web-based protocols. The most common protocol is the Simple Object Access Protocol (SOAP).
>
> The SAS BI Web Services for Java interface is based on the XML for Analysis (XMLA) Version 1.1 specification.

SAS Shared Web Assets
> contains graph applet JARs that are shared across SAS Web applications. They display graphs in stored processes and in the SAS Stored Process Web application.

SAS Web Infrastructure Platform Services
> provide a common infrastructure for SAS Web applications. The infrastructure supports activities such as auditing, authentication, configuration, status and monitoring, e-mail, theme management, and data sharing across SAS Web applications.

SAS Logon Manager
>provides a common user authentication mechanism for SAS Web applications. It displays a dialog box for user ID and password entry, authenticates the user, and launches the requested application. SAS Logon Manager supports a single sign-on authentication model. When this model is enabled, it provides access to a variety of computing resources (including servers and Web pages) during the application session without repeatedly prompting the user for credentials.
>
>You can configure SAS Logon Manager to display custom messages and to specify whether a logon dialog box is displayed when users log off.
>
>In addition, you can use third-party products in conjunction with SAS Logon Manager to enable users to access multiple Web applications within the same browser session.

SAS Preferences Manager
>provides a common mechanism for managing preferences for SAS Web applications. The feature enables administrators to set default preferences for locale, theme, alert notification, and time, date, and currency display. Within each Web application, users can view the default settings and update their individual preferences.

SAS Stored Process Web Application
>executes stored processes on behalf of a Web client and return results to a Web browser. The SAS Stored Process Web application is similar to the SAS/IntrNet Application Broker, and has the same general syntax and debug options.

SAS Web Administration Console
>provides features for monitoring and administering middle-tier components. This browser-based interface enables administrators to do the following:
>
>□ view a list of users who are logged on to SAS Web applications, and send e-mail to active users
>
>□ create, delete, and manage permissions for folders on the SAS Content Server
>
>□ view configuration information for each middle-tier component

SAS Content Server
>stores digital content (such as documents, reports, and images) that is created and used by SAS Web applications. For more information, see the SAS content server topic in *SAS Intelligence Platform: Overview*.

The following documentation is provided for the SAS Web Infrastructure Platform:

□ the *SAS Information Delivery Portal: Introduction*, which provides a high-level description of the features of the SAS Information Delivery Portal

□ the *SAS Intelligence Platform: Web Application Administration Guide*, which describes the configuration and administration tasks that are necessary to implement Web applications (including the SAS Information Delivery Portal) that were developed using the SAS Web Infrastructure Platform

□ *Developing Portlets for the SAS Information Delivery Portal*, which describes how to use the platform to develop a portal-like Web application or to customize or extend the functionality of the SAS Information Delivery Portal

□ help files, accessible through the SAS Web Infrastructure Platform and SAS Information Delivery Portal's user interface, which provide instructions for page navigation, logging on and logging off, metadata searching, bookmarking, personalization, and content administration

Glossary

access control entry
a set of identities and permissions that are directly associated with a particular resource. Each access control entry is directly associated with only one resource. More than one ACE can be associated with each resource. Short form: ACE.

access control template
a reusable named authorization pattern that you can apply to multiple resources. An access control template consists of a list of users and groups and indicates, for each user or group, whether permissions are granted or denied. Short form: ACT.

ACE
See access control entry.

ACT
See access control template.

administered object
a pre-configured Java Message Service (JMS) object that is created by a system administrator for use by client applications. Administered objects handle the complexities of using JMS in conjunction with proprietary messaging systems so that client applications can focus on business logic. See also Java Message Service and Event Broker Service.

alert
an automatic notification of an electronic event that is of interest to the recipient.

archive
in the Publishing Framework, a package that is compressed and saved to a directory. The archive contains the contents of a package, plus metadata that is necessary for extracting the contents.

archive path
in the Publishing Framework, the complete path of the location to which a user or program can publish an archive package.

asynchronous logging
a process that enables log messages to be queued within and delivered from a thread other than the main application thread. See also Logging Service.

authentication
the process of verifying the identity of a person or process within the guidelines of a specific authorization policy.

authentication domain
a SAS internal category that pairs logins with the servers for which they are valid. For example, an Oracle server and the SAS copies of Oracle credentials might all be classified as belonging to an OracleAuth authentication domain.

authentication provider
a software component that is used for identifying and authenticating users. For example, an LDAP server or the host operating system can provide authentication.

authorization
the process of determining which users have which permissions for which resources. The outcome of the authorization process is an authorization decision that either permits or denies a specific action on a specific resource, based on the requesting user's identity and group memberships.

background
a mode of computer processing that does not require user interaction and which allows users to perform multiple tasks on the computer concurrently. In the SAS Information Delivery Portal, some stored processes run in the background so that you can perform other portal tasks during processing.

base
the distinguished name (DN) to use as a starting place for searches of an LDAP directory. Searches can begin at any point in a directory tree. Beginning the search lower in the tree can dramatically reduce both the search time and spurious results. For example, the base might be specified in SAS system options as BaseDN or ldapBase.

bind
to create an association among two or more entities for a particular scope of time and place. For example, an association could be created between two or more programming objects, between a variable name and an object, between a symbolic address and a real machine address, or between a client and a server.

bookmark
a placeholder that stores either the address and name of a document or the address of a location within a document so that you can quickly and easily return to that document or location later.

chaining
a process by which a log message is passed to the current logging context and to each context in the hierarchy of the current logging context. If chaining is not specified at a particular level in the hierarchy, then the log message is not passed beyond that level. See also logging context and Logging Service.

channel
a virtual communication path for distributing information. In SAS, a channel is identified with a particular topic (just as a television channel is identified with a particular radio frequency). Using the features of the Publishing Framework, authorized users or applications can publish digital content to the channel, and authorized users and applications can subscribe to the channel in order to receive the content. See also publish and subscribe.

client
a computer or application that requests services, data, or other resources from a server. See also server.

cluster
a group of machines that participate in load balancing. Each machine in the cluster runs an object spawner that handles client requests for connections.

COM
See Component Object Model.

Common Object Request Broker Architecture
a standard API for distributed object communication. CORBA was created by the Object Management Group. It is the most widely used distributed object standard for connecting operating system platforms from multiple vendors. Short form: CORBA.

component
a self-contained, reusable programming object that provides some type of service to other components in an object-oriented programming environment.

Component Object Model
an object-oriented programming model that defines how software components interact within a single process or between processes. For example, COM includes standard rules of communication that enable a user-interface object to be dragged and dropped from one application window to another. Short form: COM.

Connection Factory
an application interface that enables Java programs to make IOM bridge connections to IOM servers. The interface provides the scalability features of pooling and server failover as well as support for load-balancing spawners. The interface is part of the SAS Foundation Services Connection Service. See also SAS Foundation Services and Connection Service.

Connection Service
one of the SAS Foundation Services. This service enables applications to (1) connect to IOM servers that use the IOM Bridge protocol, (2) use the Connection Factory to access existing connection objects and to create new connection objects for various server configurations, and (3) use advanced connection management features, such as connection pooling, failover, and load balancing, which are available through the Connection Factory. See also SAS Foundation Services and Connection Factory.

container
a software application environment in which a component such as an applet, an application client, a servlet, a JavaServer page, or an Enterprise Java Bean is executed. Containers provide specific services that can be exploited by the types of components that they support.

content type
a value that tells a client's Web browser how to interpret and display a transferred object such as an image file, a sound file, or a video file. For example, GIF, JPG, TIFF, MIDI, and WAV are content types. See also Multipurpose Internet Mail Extensions.

context
the set of facts or circumstances that surround a situation or event. In Java applications, context generally refers to a collection of settings and attributes that describe a container or service that is currently executing.

CORBA
See Common Object Request Broker Architecture.

DCOM
See Distributed Component Object Model.

delivery transport
in the Publishing Framework, the method of delivering a package to the consumer. Supported transports include e-mail, message queue, and WebDAV. Although not a true transport, a channel also functions as a delivery mechanism. See also e-mail, message queue, WebDAV (Web Distributed Authoring and Versioning), and channel.

directory information tree
the entries that an LDAP directory server contains. These entries are stored in a hierarchical, tree-like structure. Short form: DIT.

directory schema
the set of defined attributes and object classes that defines the content of acceptable entries within an LDAP directory server.

Discovery Service
one of the SAS Foundation Services. This service enables applications to (1) find implementations of the SAS Foundation Services based on specified service capabilities and, optionally, service attributes, and (2) rediscover a previously discovered service by using the service's Discovery Service ID. The Discovery Service can find service implementations that have been deployed locally for the application's exclusive use, as well as service implementations that have been deployed remotely for the use of multiple applications. See also SAS Foundation Services.

distinguished name
a unique identifier of an entry in an LDAP network directory. In effect, a distinguished name is the path to the object in the directory information tree. Short form: DN.

Distributed Component Object Model
an extension to the Component Object Model (COM) that enables components to request services from components that are on other computers in a network. Short form: DCOM. See also component and Component Object Model.

DIT
See directory information tree.

DN
See distinguished name.

DNS name
a name that is meaningful to people and that corresponds to the numeric TCP/IP address of a computer on the Internet. For example, www.alphaliteairways.com might be the DNS name for an Alphalite Airways Web server whose TCP/IP address is 192.168.145.6.

domain
a database of users that has been set up by an administrator by using a specific authentication provider such as LDAP or the host operating system. The domain name should be unique within your enterprise. For example, you should not have a Windows domain and a UNIX domain that are both named "SALES". See also authentication domain.

e-mail
a system for transmitting messages electronically, usually between two computers.

encryption
the act or process of converting data to a form that only the intended recipient can read or use.

entry
   a set of name/value pairs that describe a resource that is defined on an LDAP directory server. See also name/value pair.

entry filter
   See package entry type filter, MIME type filter.

event
   a type of notification that triggers an Event Broker Service to take a particular action. An event is specified as well-formed XML that contains the name of the event, associated headers and properties, and the message body. See also Event Broker Service and XML (Extensible Markup Language).

Event Broker Service
   one of the SAS Foundation Services. This service enables applications to send and deliver events to the appropriate handling agents for processing. The handling agent can be (1) a statically defined process flow that runs in its own thread within the Event Broker Service to process the event or (2) an application that has registered itself with the Event Broker Service at run time so that it can receive events. See also SAS Foundation Services and event.

event message specification
   a well-formed XML representation of an event. The event message specification contains attributes of the event such as name, version, identity, credentials, sender, send time, and response type. See also Event Broker Service.

Extensible Markup Language
   a markup language that structures information by tagging it for content, meaning, or use. Structured information contains both content (for example, words or numbers) and an indication of what role the content plays. For example, content in a section heading has a different meaning from content in a database table. Short form: XML.

factory
   an object that controls the creation of other objects, access to other objects, or both. See also Connection Factory.

filter
   in the Publishing Framework, a named set of criteria that are used for selecting either a package or specific package entries for a subsequent operation. Filters are useful in the following contexts: channel subscription properties, viewer file processing, stored processes, and LDAP directory searches. For example, a channel subscriber can define a name/value filter that controls the types of packages that he or she receives. See also name/value pair, package entry type filter, and MIME type filter.

foundation services
   See SAS Foundation Services.

group
   in the Publishing Framework, a collection of subscribers. When a group is subscribed to a channel, each member of the group who has a defined subscriber entry in the Publishing Framework will receive the information that is published to that channel. Only metadata administrators have the authority to add members or to remove members from a group.

HTTP
See HyperText Transfer Protocol.

HTTP server
a server that handles an HTTP request from a client such as a Web browser. Usually the client's HTTP request indicates that the client wants to retrieve information that is pointed to by a URL. An example of a popular HTTP server is the Apache HTTP Server from the Apache Software Foundation. See also Web server.

HyperText Transfer Protocol
a protocol for transferring data to the Internet. HTTP provides a way for servers and Web clients to communicate. It is based on the TCP/IP protocol. Short form: HTTP.

information map
a collection of data items and filters that provides a user-friendly view of a data source. When you use an information map to query data for business needs, you do not have to understand the structure of the underlying data source or know how to program in a query language.

Information Service
one of the SAS Foundation Services. This service enables applications to perform a federated search of any repositories that a user has a connection to. The term federated means connected and treated as one. The classes in the Information Service package enable applications to create a single filter that can search disparate repositories (for example, SAS Metadata Repositories and LDAP repositories). See also SAS Foundation Services.

input stream
a connection that enables a client application to stream data into a stored process. The input stream is visible to the stored process as a SAS fileref.

Integrated Object Model
a set of object-based interfaces to features or services that are provided by Base SAS software. IOM enables application developers to use industry-standard programming languages, programming tools, and communication protocols to develop client programs that access these services on IOM servers. Short form: IOM.

Integrated Object Model server
a SAS object server that is launched in order to fulfill client requests for IOM services. Short form: IOM server.

IOM
See Integrated Object Model.

IOM bridge
a software component of SAS Integration Technologies that enables Java clients and Windows clients to access an IOM server.

IOM bridge for COM
the software component of SAS Integration Technologies that is used (transparently) when a Windows client accesses an IOM server on a platform other than Windows. The bridge enables applications that conform to the Microsoft COM or DCOM specifications to access SAS servers in operating environments such as z/OS and UNIX that do not support the COM or DCOM specifications. See also Integrated Object Model, Component Object Model, and Distributed Component Object Model.

IOM bridge for Java
a software component of SAS Integration Technologies that enables Java clients to access IOM servers.

IOM server
See Integrated Object Model server.

Java Development Kit
a software development environment that is available from Sun Microsystems, Inc. The JDK includes a Java Runtime Environment (JRE), a compiler, a debugger, and other tools for developing Java applets and applications. Short form: JDK.

Java Message Service
an application programming interface (API) that enables client applications to access the facilities of a message service. This interface provides a standard way for Java programs to create, send, receive, and read messages. Short form: JMS.

Java Naming and Directory Interface
a standard extension to the Java platform that enables developers to create applications that can interact with a number of different naming services and directory services, such as the Domain Name System (DNS) and the Lightweight Directory Access Protocol (LDAP). Short form: JNDI.

Java RMI
See remote method invocation.

Java Virtual Machine
a program that interprets Java programming code so that the code can be executed by the operating system on a computer. The JVM can run on either the client or the server. The JVM is the main software component that makes Java programs portable across platforms. A JVM is included with JDKs and JREs from Sun Microsystems, as well as with most Web browsers. Short form: JVM.

JavaServer page
a type of servlet that enables users to create Java classes through HTML. Short form: JSP.

JDK
See Java Development Kit.

JMS
See Java Message Service.

JNDI
See Java Naming and Directory Interface.

JSP
See JavaServer page.

JVM
See Java Virtual Machine.

keyword
in BI Manager, a descriptive word or phrase that can be stored in the metadata for one or more BI objects and which can subsequently be used in a filtered search. For example, 'XMLA Web Service' is a reserved keyword that is used to identify and locate stored processes that can be used as Web services.

layout pattern
a feature of the Logging Service that enables you to associate a layout with a particular logging output destination. The layout specifies how the output is to be formatted before it is sent to the output destination. The layout is specified as a

pattern string similar to the pattern strings that are used in the C language <code>printf</code> statement. See also Logging Service.

LDAP
See Lightweight Directory Access Protocol.

LDAP Data Interchange Format
a file format that is widely used for building a database of directories that are stored on multiple servers, for adding large numbers of directories to a directory database, and for synchronizing the contents of those directories. The servers on which the directories are stored must support the LDAP protocol. Short form: LDIF.

LDAP directory
a repository that contains data about an enterprise's users and resources, as well as related security information, and that stores this data and information in a format that clients on a network can access by using the Lightweight Directory Access Protocol (LDAP).

LDAP directory server
a server that provides access to one or more LDAP directories.

LDAP server
See LDAP directory server.

LDIF
See LDAP Data Interchange Format.

Lightweight Directory Access Protocol
a protocol that is used for accessing directories or folders. LDAP is based on the X.500 standard, but it is simpler and, unlike X.500, it supports TCP/IP. Short form: LDAP.

link
(1) a portal content item that can be accessed using a URL; (2) a character string in a portal that you can click to initiate an action.

load balancing
for IOM bridge connections, a program that runs in the object spawner and that uses an algorithm to distribute work across object server processes on the same or separate machines in a cluster.

local service deployment
a service deployment that supports an application's exclusive access to a set of SAS Foundation Services that are deployed within a single Java Virtual Machine (JVM). See also service deployment.

localhost
a keyword that is used to specify the machine on which a program is executing. If a client specifies localhost as the server address, the client connects to a server that runs on the same machine.

logging context
a collection of attributes and settings that define a particular way in which the Logging Service is to be used. The logging context specifies where and in what format logging calls will be written. See also Logging Service.

Logging Service
one of the SAS Foundation Services. This service enables applications to (1) send run-time messages to one or more output destinations, including consoles, files, and socket connections; (2) configure and control the format of information that is sent to a particular destination, either by using static configuration files or by invoking run-time methods that control logging output; and (3) perform remote logging, which

involves sending log messages that are generated in one Java Virtual Machine (JVM) to another JVM. See also SAS Foundation Services.

logical name
a unique name that can be assigned to related resources in order to manage the resources as a group.

logical server
in the SAS Metadata Server, the second-level object in the metadata for SAS servers. A logical server specifies one or more of a particular type of server component, such as one or more SAS Workspace Servers.

login
a SAS copy of information about an external account. Each login includes a user ID and belongs to one SAS user or group. Most logins do not include a password.

message node
in a process flow diagram, a node that encapsulates inputs and outputs to one or more process nodes. See also process node and process flow.

message queue
in application messaging, a place where one program can send messages that will be retrieved by another program. The two programs communicate asynchronously. Neither program needs to know the location of the other program nor whether the other program is running. See also delivery transport.

message queue polling
a feature of the object spawner that monitors message queues and distributes message queue tasks efficiently among SAS processes.

metadata
a description or definition of data or information.

metadata repository
a collection of related metadata objects, such as the metadata for a set of tables and columns that are maintained by an application. A SAS Metadata Repository is an example.

metadata server
a server that provides metadata management services to one or more client applications. A SAS Metadata Server is an example.

middle tier
in a SAS business intelligence system, the tier in which J2EE Web applications and J2EE enterprise applications execute.

MIME
See Multipurpose Internet Mail Extensions.

MIME type filter
in the Publishing Framework, a filter that specifies which file entries of a specific MIME type to include in packages that are published to a subscriber. See also Multipurpose Internet Mail Extensions.

Multipurpose Internet Mail Extensions
a method of registering content types and their associated actions with mechanisms for actions such as retrieval and display. Short form: MIME. See also content type.

name/value filter
in the Publishing Framework, a subscription property that determines which packages are published to that subscriber. In order to receive a particular package, a subscriber's name/value filter must contain the name/value pair that the publisher specifies in the package. See also name/value pair and filter.

name/value pair
in the Publishing Framework, an attribute and value that the publisher uses to describe the contents of a package or a package entry. For example, a 'type=production' name/value pair might be used to specify that a package contains production data. One or more site-specific name/value pairs can be assigned to an entire package or to a selected package entry. Subscribers can use name/value pairs to define name/value filters, which control the types of packages that are delivered to them. See also filter.

named service
a SAS Foundation Service that is bound to the service registry by name. Named services are defined by named service bindings. Named service bindings enable applications to locate a service based on the name that is bound to the registry. See also bind, named service binding, and service registry.

named service binding
the method that is used to bind one of the SAS Foundation Services to a service registry when the service has been deployed remotely. Named service bindings enable applications to locate a service by using the name that is bound to the registry. See also bind, named service, and service registry.

namespace
a URL that identifies an entry in a WebDAV package.

naming hierarchy
a naming system that the Event Broker Service uses to map events to an event configuration. The event names that are included in the hierarchy are separated by periods, as in this example: <code>Animals</code>, <code>Animals.Dogs</code>, <code>Animals.Dogs.Retriever</code>. See also event and Event Broker Service.

object
in object-oriented programming, an instantiation or specific representation of a class.

object class
a definition of a type of object that can be stored in an LDAP directory. Each object class consists of a set of attributes (for example, name and description) to which you can assign values in order to define a specific instance of the object. The attributes can be either mandatory or optional. The complete set of object classes for a directory defines its schema.

Object Management Group
a nonprofit industry consortium formed in April 1989 that develops standards for distributed objects. OMG standards include the Common Object Request Broker Architecture (CORBA) and the Internet Inter-ORB Protocol (IIOP). Short form: OMG.

Object Request Broker
in object-oriented programming, a middle-tier component that supports communication between clients and servers. When a client invokes a method that is supported by an object server class, the ORB finds an instance of the object server class, invokes the requested method, and returns the results to the requesting client. An ORB enables clients and servers to dynamically discover each other and to communicate with each other across a network. Short form: ORB.

object spawner

a program that instantiates object servers that are using an IOM bridge connection. The object spawner listens for incoming client requests for IOM services. When the spawner receives a request from a new client, it launches an instance of an IOM server to fulfill the request. Depending on which incoming TCP/IP port the request was made on, the spawner either invokes the administrator interface or processes a request for a UUID (Universal Unique Identifier).

OMG

See Object Management Group.

ORB

See Object Request Broker.

override

in the Publishing Framework, a subscription property that is assigned to a single channel and that is used in place of a default property that is otherwise assigned to channel subscriptions.

package

a container for data that has been generated or collected for delivery to consumers by the SAS Publishing Framework. Packages can contain SAS files (SAS catalogs; SAS data sets; various types of SAS databases, including cubes; and SAS SQL views), binary files (such as Excel, GIF, JPG, PDF, PowerPoint and Word files), HTML files (including ODS output), reference strings (such as URLs), text files (such as SAS programs), and viewer files (HTML templates that format SAS file items for viewing). Packages also contain metadata such as a description, an abstract, and user-specified name/value pairs.

package entry

in the Publishing Framework, an item within a package.

package entry type filter

in the Publishing Framework, a filter that specifies which types of package entries to publish to a subscriber. See also filter.

parameter

a data item that is passed to a routine.

permanent package

a container for content that was produced by a SAS program or by a third-party application, and that is written to a specific location. Permanent packages remain in existence even after the stored process completes execution and the client disconnects from the server. See also transient package.

permission

the type of access that a user or group has to a resource. The permission defines what the user or group can do with the resource. Examples of permissions are ReadMetadata and WriteMetadata.

PFD

See process flow diagram.

plug-in

a file that modifies, enhances, or extends the capabilities of an application program. The application program must be designed to accept plug-ins, and the plug-ins must meet design criteria specified by the developers of the application program. In SAS Management Console, a plug-in is a JAR file that is installed in the SAS Management Console directory to provide a specific administrative function. The plug-ins enable users to customize SAS Management Console to include only the functions that are needed.

pool
a group of server connections that can be shared and reused by multiple client applications. A client-side pool consists of one or more puddles. See also puddle.

pooling
the act or process of creating a pool. See also pool.

portal
a Web application that enables users to access Web sites, data, documents, applications, and other digital content from a single, easily accessible user interface. A portal's personalization features enable each user to configure and organize the interface to meet individual or role-based needs. See also portlet.

portlet
a Web component that is managed by a Web application and that is aggregated with other portlets to form a page within the application. Portlets can process requests from the user and generate dynamic content.

Process Editor
a utility that is used for configuring process flows for the Event Broker Service. See also Event Broker Service and process flow.

process flow
programming logic that is executed in response to an event message and that in some cases returns a result. See also Event Broker Service and event.

process flow diagram
in SAS Foundation Services, a diagram in the Process Editor that illustrates the order in which process nodes and message nodes are to be executed within a particular process flow. Short form: PFD. See also Event Broker Service, process flow, process node, and message node.

process node
in a process flow diagram, a node that represents run-time processing that is performed on messages. See also process flow diagram, process flow, and message node.

prompt
a parameter that enables a user to specify one or more values, which can be used by consumers such as applications, stored processes, and reports.

publication channel
an information repository that has been established using the SAS Publishing Framework and that can be used to publish information to users and applications. See also publish.

publish
to deliver electronic information, such as SAS files (including SAS data sets, SAS catalogs, and SAS data views), other digital content, and system-generated events to one or more destinations. These destinations can include e-mail addresses, message queues, publication channels and subscribers, WebDAV-compliant servers, and archive locations.

publisher
any person, SAS program, or application that uses the Publishing Framework to distribute information.

Publishing Framework
a component of SAS Integration Technologies that enables both users and applications to publish SAS files (including data sets, catalogs, and database views), other digital content, and system- generated events to a variety of destinations. The

Publishing Framework also provides tools that enable both users and applications to receive and process published information.

puddle
a group of servers that are started and run using the same login credentials. Each puddle can also allow a group of clients to access the servers.

queue
See message queue.

queue manager
an application that enables programs to send messages to and receive messages from message queues.

remote method invocation
a Java programming feature that provides for remote communication between programs by enabling an object that is running in one Java Virtual Machine (JVM) to invoke methods on an object that is running in another JVM, possibly on a different host. Short form: RMI. See also Java Virtual Machine.

remote service deployment
a service deployment that supports shared access to a set of SAS Foundation Services that are deployed within a single Java Virtual Machine (JVM), but which are available to other JVM processes. Applications use the remote service deployment to deploy and access remote foundation services. See also service deployment.

repository
a location in which data, metadata, or programs are stored, organized, and maintained, and which is accessible to users either directly or through a network. See also metadata repository, SAS Metadata Repository, and WebDAV repository.

resource
any object that is registered in a metadata repository. For example, a resource can be a server, a stored process, or a login.

result type
the kind of output that is produced by a stored process. Result types include none, streaming, permanent package, and transient package.

RMI
See remote method invocation.

SAS application server
a server that provides SAS services to a client. In the SAS Open Metadata Architecture, the metadata for a SAS application server specifies one or more server components that provide SAS services to a client. See also component.

SAS BI Web service
a Web service that adheres to the XML for Analysis (XMLA) specification for executing SAS Stored Processes.

SAS Foundation Services
a set of core infrastructure services that programmers can use in developing distributed applications that are integrated with the SAS platform. These services provide basic underlying functions that are common to many applications. These functions include making client connections to SAS application servers, dynamic service discovery, user authentication, profile management, session context management, metadata and content repository access, activity logging, event management, information publishing, and stored process execution. See also service.

SAS log
a file that contains a record of the SAS statements that you enter as well as messages about the execution of your program.

SAS Management Console
a Java application that provides a single user interface for performing SAS administrative tasks.

SAS Metadata Repository
one or more files that store metadata about application elements. Users connect to a SAS Metadata Server and use the SAS Open Metadata Interface to read metadata from or write metadata to one or more SAS Metadata Repositories. The metadata types in a SAS Metadata Repository are defined by the SAS Metadata Model.

SAS Metadata Server
a multi-user server that enables users to read metadata from or write metadata to one or more SAS Metadata Repositories. The SAS Metadata Server uses the Integrated Object Model (IOM), which is provided with SAS Integration Technologies, to communicate with clients and with other servers.

SAS OLAP Server
a SAS server that provides access to multidimensional data. The data is queried using the multidimensional expressions (MDX) language.

SAS Open Metadata Architecture
a general-purpose metadata management facility that provides metadata services to SAS applications. The SAS Open Metadata Architecture enables applications to exchange metadata, which makes it easier for these applications to work together.

SAS package
See package. Short form: SPK.

SAS Stored Process
a SAS program that is stored on a server and which can be executed as requested by client applications. SAS Stored Processes can be used with either a SAS Workspace Server or a SAS Stored Process Server.

SAS Stored Process Server
a SAS IOM server that is launched in order to fulfill client requests for SAS Stored Processes. See also IOM server.

SAS Stored Process Web Application
a Web application that enables you to execute stored processes and have the results returned to a Web browser.

SAS Workspace Server
a SAS IOM server that is launched in order to fulfill client requests for IOM workspaces. See also IOM server and workspace.

schema
See directory schema.

server
a computer system that provides data or services to multiple users on a network. The term 'server' sometimes refers to the computer system's hardware and software, but it often refers only to the software that provides the data or services. In a network, users might log on to a file server (to store and retrieve data files), a print server (to use centrally located printers), or a database server (to query or update databases). In a client/server implementation, a server is a program that waits for and fulfills requests from client programs for data or services. The client programs might be running on the same computer or on other computers. See also service.

service

one or more application components that an authorized user or application can call at any time to provide results that conform to a published specification. For example, network services transmit data or provide conversion of data in a network, database services provide for the storage and retrieval of data in a database, and Web services interact with each other on the World Wide Web. See also SAS Foundation Services.

service configuration

a set of values that can be customized for a particular service in SAS Foundation Services. By editing a service configuration, you can override the default configuration for the foundation service. See also SAS Foundation Services.

service deployment

a collection of SAS Foundation Services that specifies the data that is necessary in order to instantiate the services, as well as dependencies upon other services. Applications query a metadata source (a SAS Metadata Server or an XML file) to obtain the service deployment configuration in order to deploy and access foundation services. See also SAS Foundation Services.

service deployment group

a group of SAS Foundation Services within a service deployment. Service deployment groups are used to organize foundation services within a deployment hierarchy. Applications can use service deployment groups in order to deploy only specific groups of foundation services. See also service deployment.

service interface

an interface that defines the services that one of the SAS Foundation Services is providing to its users. Each of the foundation services implements one or more service interfaces. See also SAS Foundation Services.

service loader

a utility that can be invoked by client applications that use SAS Foundation Services. The service loader enables client applications to (1) instantiate services and register them with a local Discovery Service or (2) locate a remote Discovery Service that has been deployed by a server application and register it with a local Discovery Service. See also Discovery Service.

service name

(1) a SAS Foundation Service that is bound to the service registry by name. Service names (named services) are defined by named service bindings. Applications use the named service binding to locate the service based on the name that is bound to the service registry. (2) a TCP/IP port identifier. Service names are defined in the TCP/IP SERVICES file. See also bind.

service registry

a searchable registry of service descriptions that is used to register named service bindings. Applications use the service registry to locate (via named service bindings) SAS Foundation Services that have been deployed remotely. See also named service and named service binding.

servlet

a Java program that runs on a Web server. Servlets can be considered a complementary technology to applets, which run in Web browsers. Unlike applet code, servlet code does not have to be downloaded to a Web browser. Instead, servlets send HTML or other appropriate content back to a browser or to another type of Web-based client application.

servlet container

an execution environment for Java servlets that contains a Java Virtual Machine. The servlet container also provides other services for servlets and for the Web

applications that those servlets are part of. For example, the servlet container converts HTTP requests that are sent by clients to Java objects that servlets can work with, and it converts the output of servlets to HTTP responses. An example of a popular servlet container is the JBoss server.

session context
a context that serves as a control structure for maintaining state within a bound session. 'State' includes information about the latest status, condition, or content of a process or transaction. Session Services, User Services, and Logging Services use the session context to facilitate resource management and to pass information among services. See also context and bind.

Session Service
one of the SAS Foundation Services. This service enables applications to (1) create a session context, (2) bind objects to a session context, (3) use the session context as a container for passing multiple contexts or for passing other services (such as User Services and Logging Services), and (4) notify bound objects when they are removed from the session context or when the session context is destroyed, so that objects can perform any necessary cleanup. See also SAS Foundation Services, session context, and bind.

Simple Object Access Protocol
a standard, Web-based protocol that enables applications written in various programming languages to communicate. Short form: SOAP.

single sign-on
an authentication model that enables users to access a variety of computing resources without being repeatedly prompted for their user IDs and passwords. For example, single sign-on can enable a user to access SAS servers that run on different platforms without interactively providing the user's ID and password for each platform. Single sign-on can also enable someone who is using one application to launch other applications based on the authentication that was performed when the user initially logged on.

smart object
an object that acts as a wrapper for metadata entries in order to hide details that pertain to a specific type of metadata repository. The Information Service uses smart objects to specify implementation details for specific types of repositories. See also Information Service.

SOAP
See Simple Object Access Protocol.

source code repository
the location on the application server that contains stored process source code.

source file
the SAS program that contains the stored process source code.

SPK
See SAS package.

stored process
a SAS program that is stored on a server and which can be executed as requested by client applications. There are two types of stored processes: IOM Direct Interface Stored Processes and SAS Stored Processes.

Stored Process Service
one of the SAS Foundation Services. This Java-based interface enables applications to synchronously or asynchronously execute a stored process. Execution can include

accessing SAS data sources or external files and creating new data sets, files, or other types of output that are supported by SAS. Clients of the service can pass input parameters and data streams to an executing stored process and can receive output streams or result packages from the stored process. See also SAS Foundation Services.

streaming result
a type of output that is generated by a stored process. In a streaming result, the content that the stored process generates is delivered to the client through an output stream. The output stream is generally accessible to the stored process as the _WEBOUT fileref. See also result type.

Struts
a public-domain framework for building Web applications. The Information Delivery Portal and the Web Infrastructure Platform (provided with SAS Integration Technologies) use this framework. Struts is based on standard technologies such as Java servlets, JavaBeans, resource bundles, and Extensible Markup Language (XML). It supports the Model 2 variation of the Model-View-Controller (MVC) design paradigm. Struts is part of the Apache Jakarta Project, which is sponsored by the Apache Software Foundation.

stub
a Java class that acts as an interface for a remote software object. Stubs are instantiated on the client. The client passes method calls to the stubs, and the stubs then repackage the method calls for delivery through an Object Request Broker (ORB) to remote software objects. In many distributed object systems, stubs are referred to as proxies.

subscribe
to sign up to receive electronic content that is published to a SAS publication channel.

subscriber
a recipient of information that is published to a SAS publication channel.

subscriber profile
a set of personal preferences for subscribing to SAS publication channels. A subscriber profile includes the method by which you want published information to be delivered and filtering criteria (in the form of name/value pairs) to limit the types of information that you receive. You can create multiple subscriber profiles if you want to subscribe to channels in different ways.

subscription
the association of a subscriber with a group or a channel.

thread
a single path of execution of a process in a single CPU, or a basic unit of program execution in a thread-enabled operating system.

transient package
a container for content that was produced by a SAS program or by a third-party application for immediate use, and that is not saved. After the client program disconnects from the server, the transient package disappears. See also permanent package.

transport
(1) in SAS Foundation Services, a communication protocol through which the Event Broker Service can listen for, receive, and respond to external events. Each transport that is specified in the service configuration runs in its own thread of execution. (2) In the Publishing Framework, this term is a short form of the term 'delivery transport.' See also Event Broker Service and delivery transport.

Unicode Transformation Format 8
a method for converting 16-bit Unicode characters to 8-bit characters. This format supports all of the world's languages, including those that use non-Latin 1 characters. Short form: UTF-8.

Uniform Resource Identifier
a string that identifies resources such as files, images, and services on the World Wide Web. A URL is a type of URI. Short form: URI. See also Uniform Resource Locator.

Uniform Resource Locator
a character string that is used by a Web browser or other software application to access or identify a resource on the Internet or on an intranet. The resource could be a Web page, an electronic image file, an audio file, a JavaServer page, or any other type of electronic object. The full form of a URL specifies which communications protocol to use for accessing the resource, as well as the directory path and filename of the resource. Short form: URL.

Universal Unique Identifier
a 128-bit identifier that is comprised of date/time information and a host's IEEE node address. The 128 bits can represent $1x10^{**}39$ unique identifiers. UUIDs are also referred to as Global Unique Identifiers (GUIDs). Short form: UUID.

URI
See Uniform Resource Identifier. See also Uniform Resource Locator.

URL
See Uniform Resource Locator.

user
a person who is registered in a SAS metadata environment.

user context
a context that contains information about the user who is associated with an active session. The user context contains information such as the user's identity, profile, and active repository connections. See also context.

User Service
one of the SAS Foundation Services. This service enables applications to (1) create, locate, maintain, and aggregate information about users of the SAS Foundation Services, (2) store and retrieve user context objects for sharing between applications, (3) manage and access user profiles, and (4) access group profiles. See also SAS Foundation Services.

UTF-8
See Unicode Transformation Format 8.

UUID
See Universal Unique Identifier.

viewer
in the Publishing Framework, a custom-written template that contains HTML tags for formatting package content for view-only delivery transports such as e-mail.

Web Distributed Authoring and Versioning
an emerging industry standard, based on extensions to HTTP 1.1, that enables users to collaborate in the development of files and collections of files on remote Web servers. Short form: WebDAV. See also delivery transport.

Web server
a server machine and software that enable organizations to share information through intranets and through the Internet.

Web Service Description Language file
a file that is obtained from a Web service. The WSDL describes the methods that are available to the client application, the endpoint (where to call the Web service), and the format of the XML that is required in order to call the Web service. Short form: WSDL file.

WebDAV
See Web Distributed Authoring and Versioning.

WebDAV content portlet
a portlet that displays the contents of an HTML fragment that is stored in the portal's WebDAV repository.

WebDAV repository
a collection of files that are stored on a Web server so that authorized users can read and edit them. See also Web Distributed Authoring and Versioning.

wizard
an interactive utility program that consists of a series of dialog boxes, windows, or pages. Users supply information in each dialog box, window, or page, and the wizard uses that information to perform a task.

workspace
in the IOM object hierarchy for a SAS Workspace Server, an object that represents a single session in SAS. See also Integrated Object Model.

WSDL file
See Web Service Description Language file.

XML
See Extensible Markup Language.

XML for Analysis
a standard specification developed by several companies for use as a Web service interface to access online analytical processing (OLAP) functions and data-mining functions. Short form: XMLA.

XMLA
See XML for Analysis.

Index

W

Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to **yourturn@sas.com**. Include the full title and page numbers (if applicable).

- If you have comments about the software, please send them to **suggest@sas.com**.

SAS® Publishing Delivers!

Whether you are new to the work force or an experienced professional, you need to distinguish yourself in this rapidly changing and competitive job market. SAS® Publishing provides you with a wide range of resources to help you set yourself apart. Visit us online at support.sas.com/bookstore.

SAS® Press

Need to learn the basics? Struggling with a programming problem? You'll find the expert answers that you need in example-rich books from SAS Press. Written by experienced SAS professionals from around the world, SAS Press books deliver real-world insights on a broad range of topics for all skill levels.

support.sas.com/saspress

SAS® Documentation

To successfully implement applications using SAS software, companies in every industry and on every continent all turn to the one source for accurate, timely, and reliable information: SAS documentation. We currently produce the following types of reference documentation to improve your work experience:

- Online help that is built into the software.
- Tutorials that are integrated into the product.
- Reference documentation delivered in HTML and PDF – **free** on the Web.
- Hard-copy books.

support.sas.com/publishing

SAS® Publishing News

Subscribe to SAS Publishing News to receive up-to-date information about all new SAS titles, author podcasts, and new Web site features via e-mail. Complete instructions on how to subscribe, as well as access to past issues, are available at our Web site.

support.sas.com/spn

§sas | THE POWER TO KNOW®