SAS® 9.1 Integration Technologies
Technical Overview

Table of Contents

SAS® 9.1 Integration Technologies: Technical Overview

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 the enabling software 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.

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.
- 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 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.
- a Stored Process application interface, which enables client applications to execute SAS programs that are stored centrally on a server.
- Web services implementations of the XML for Analysis interface for calling SAS Stored Processes.
- the SAS Web Infrastructure Kit, 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. (To incorporate SAS Metadata Server functions into your SAS programs, you can use the SAS Open Metadata Interface. For information, see ⊕ Getting Started with the SAS 9.1 Open Metadata Interface and the ⊕ SAS 9.1 Open Metadata Interface: User's Guide.

To read more about each feature of Integration Technologies, use the links at left. For information about how to use these features to integrate or extend your application environment, refer to the Integration Technologies Developer's Guide.

Support for SAS Open Metadata Architecture

SAS 9.1 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

- 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 Kit to create Web applications that use portal technology
- use SAS Foundation Services to implement Java applications that are integrated with the SAS intelligence architecture
- install and operate other SAS products that depend on the SAS intelligence architecture.

If you are installing SAS Integration Technologies for the first time, you will probably want to use the SAS Open Metadata Architecture so that you can take advantage of all of the features and benefits previously listed. Refer to the SAS Integration Technologies Administrator's Guide for detailed information about administration and configuration tasks for this implementation model.

For more information about the SAS Open Metadata Architecture, refer to the ⊕ SAS 9.1 Open Metadata Interface: Reference and the ⊕ SAS 9.1 Metadata Server: Setup Guide.

Continued Support of LDAP/LDIF

SAS 9.1 Integration Technologies continues to support storage of metadata on a Lightweight Directory Access Protocol (LDAP) directory server, or in flat configuration files in the LDAP Data Interchange Format (LDIF). If you are using a previous version of SAS Integration Technologies, you can choose to upgrade to SAS 9.1 Integration Technologies with your existing LDAP or LDIF configuration. Refer to the SAS Integration Technologies Administrator's Guide (LDAP Version) for detailed information about administrative tasks for this implementation model.

Alternatively, you might want to migrate to the SAS Open Metadata Architecture when upgrading to SAS 9.1 Integration Technologies so that you can take advantage of the features that are available only with the new architecture.

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 using any programming environment that supports bindings to the COM/DCOM or Java object model. The programming environments that are supported include

    ♦ 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 using Web–based Distributed Authoring and Versioning (WebDAV)
- Web services development based on the XML for Analysis interface
- Web application development on Web servers that are compatible with the Java 2 platform
- the Lightweight Directory Access Protocol (LDAP), including Sun ONE Directory Server, Netscape Directory Server, IBM SecureWay Directory Server, and Microsoft Active Directory implementations.

Technical Overview

Integrated Object Model

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. The most fundamental of these interfaces is the IOM workspace interface, which provides access to Base SAS software features such as the SAS language, SAS libraries, the server file system, results content, and formatting services.

Support for SAS Metadata Server

For SAS Integration Technologies 9, the information that is needed to connect to an IOM server can now be obtained from a SAS Metadata Server. SAS Integration Technologies 9 supports additional SAS software features by allowing Java and Windows clients to access various types of IOM servers including

- other SAS Metadata Servers
- SAS OLAP servers
- SAS Stored Process Servers
- SAS Workspace Servers

Note: If you are using an LDAP directory server as your metadata repository, you can only access workspace servers.

IOM Objects

SAS Integration Technologies 9 adds

- the Windows Object Manager. Windows client programs can use this helper class to make a connection to an IOM server.
- the Java Connection Factory. Java client programs can use this helper class to make a connection to an IOM server.

If you are using a SAS Metadata Server, you can access stored process objects, OLAP objects, SAS Metadata Server objects, and workspace objects. If you are using an LDAP server, you can only access workspace objects.

Workspace Objects

A SAS workspace represents a session with the SAS System, and is functionally equivalent to a SAS Display Manager session or the execution of the SAS System as a batch job. In SAS Integration Technologies 9, you can use the new Windows Object Manager and Java Connection Factory to request a workspace within an IOM server. Each workspace supports the service hierarchy shown in Workspace Object Hierarchy.

Note: SAS Integration Technologies 9 still supports the Version 8 interface for using the Java Workspace Factory and the Windows Workspace Manager. However, it is recommended that you use the Connection Factory or Windows Object Manager interface in order to take advantage of the new features.

Stored Process, Open Metadata, and OLAP Objects

You can use the new Java Connection Factory and Windows Object Manager to request stored process, SAS Metadata Server, and OLAP objects within an IOM server. For more information on the SAS Metadata Server, refer to the

⊕ SAS 9.1 Open Metadata Architecture: Reference and the ⊕ SAS 9.1 Metadata Server: Setup Guide.

IOM Scenario

One of the chief benefits of building component−based applications using 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 illustration below. 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.
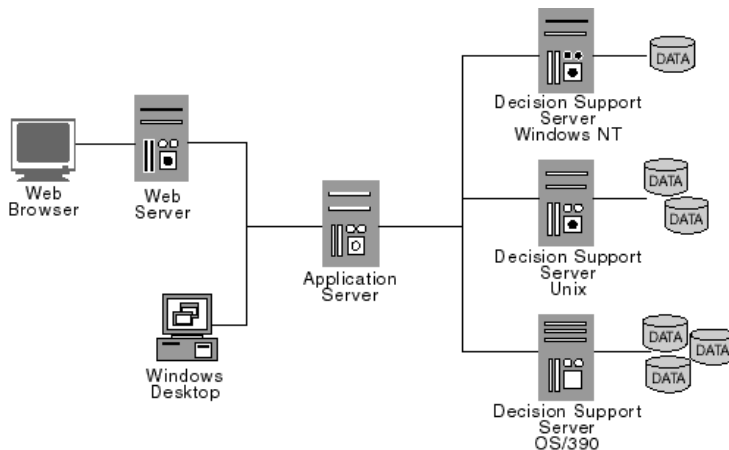
Within the Web browser context, you can use client−side scripting such as VBScript, local components such as ActiveX controls, and Java applets to implement access to server components.

Similarly, you can assess server components from desktop application environments including custom applications written in Visual Basic, Java, C++, Delphi, and PowerBuilder. Further, any desktop application container that supports 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 Microsoft's 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:

- 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 illustration depicts the object hierarchy for these interfaces.
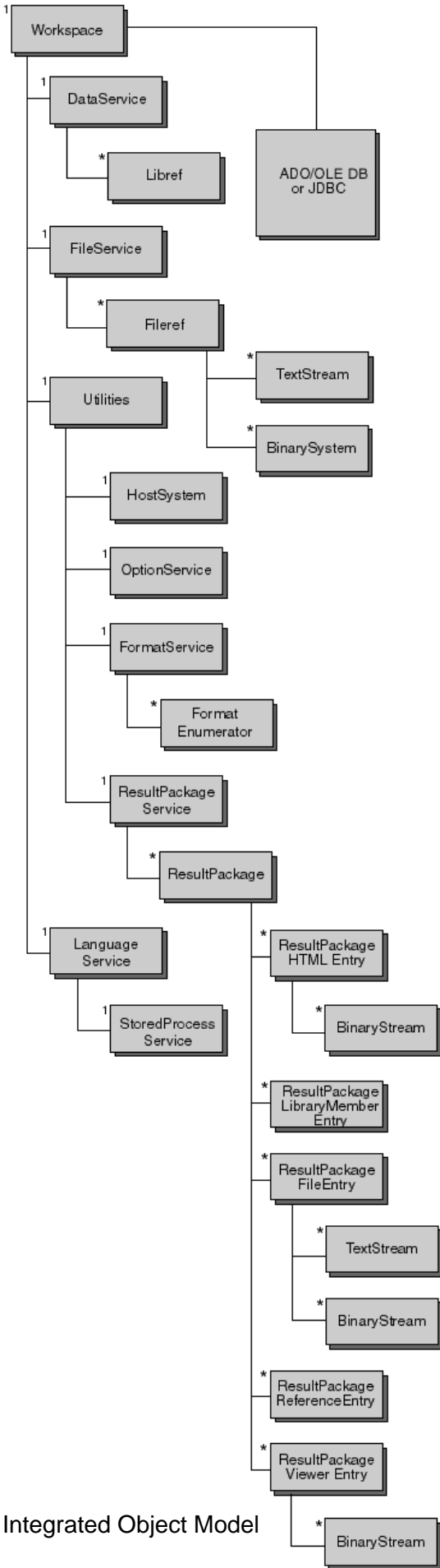
In the illustration, the number 1 indicates that that part of the hierarchy has exactly one child per parent. An asterisk (*) indicates that that part of the hierarchy has zero or greater children per parent.

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 using the SAS Output Delivery System can utilize 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 utilizes standard interface mechanisms within the Microsoft COM and Java application environments to expose the object hierarchy. This enables you to use the 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. And 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.

Technical Overview

Connecting Clients to IOM Servers

SAS Integration Technologies facilitates client/server communication across multiple vendor architectures by supporting multiple client/server interoperability standards. SAS Integration Technologies 9 supports the ability to connect to and request objects from many different IOM Servers, including SAS Metadata, OLAP, Workspace, and Stored Process servers. The illustration below depicts the different ways that clients and IOM servers can communicate.
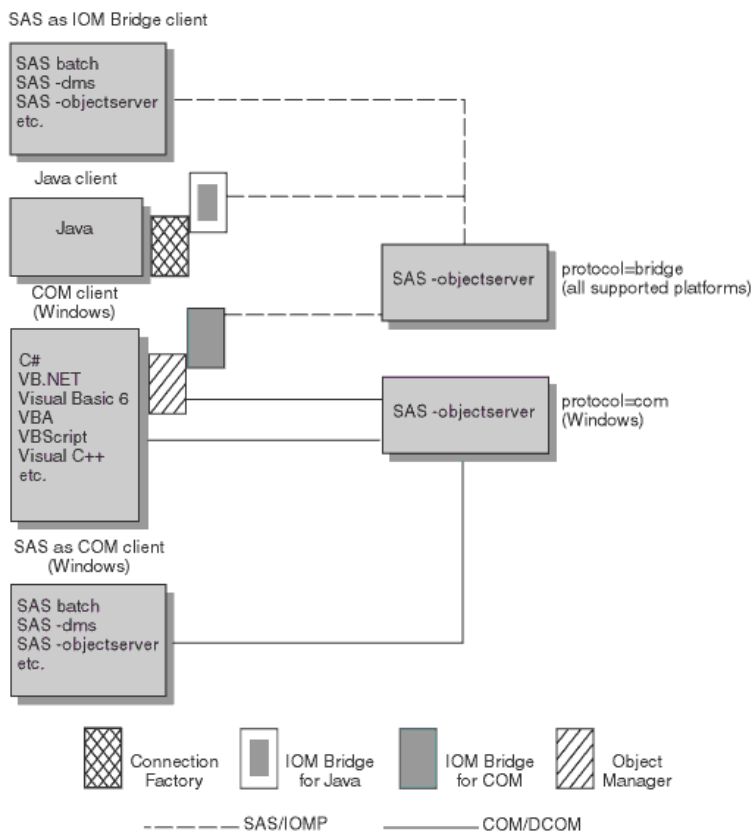
Windows Clients

As shown in the illustration, Windows clients written in languages such as Visual Basic or Visual C++ access IOM servers running in the Windows operating environment using Microsoft's Component Object Model (COM/DCOM). 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 non−Windows server platforms use the IOM Bridge for COM. This bridge allows you to develop native COM/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 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 9, Windows clients can use the Windows Object Manager to connect to IOM servers.

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 9, Java clients can use the Java Connection Factory to connect to IOM servers.

Technical Overview

Foundation Services

The SAS Foundation Services are a set of infrastructure and extension services which support the development of integrated, scalable, and secure Java–based 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.

Foundation Services Documentation

For information about configuring and administering SAS Foundation Services, refer to the Foundation Services chapter of the *SAS Integration Technologies Administrator's Guide*.

For information about using SAS Foundation Services to develop applications, refer to the Foundation Services chapter of the *SAS Integration Technologies Developer's Guide*.

For detailed class documentation, refer to com.sas.services in the *SAS Integration Technologies Developer's Guide*.

Technical Overview

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 including

- 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; and SAS programs can receive packages with SAS data sets that might in turn trigger additional analyses on that data.

Functions of the Publishing Framework

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

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 the 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. The SAS Management Console can also be used to specify which groups of users are authorized to subscribe to the channel.

If you are using an LDAP directory server to store your metadata, you can define a channel using the SAS Integration Technologies Administrator application.

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 the SAS Management Console (or the SAS Integration Technologies Administrator application, if you are using LDAP) 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, 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.

If you are using an LDAP directory server as a metadata repository, users can use SAS Subscription Manager, which is provided with SAS Integration Technologies, to manage their subscriptions. Alternatively, administrators can use the SAS Integration Technologies Administrator application to manage LDAP–based subscription information for subscribers. 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 containing one or more information entities, including SAS data sets, SAS catalogs, SAS databases, and almost any other type of digital content. If you wish, you can also define viewers that will 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. This 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 SAS Publisher user interface, which is provided as part of Base SAS
- 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 the SAS Package Retriever, which is a graphical user interface to enable 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; and APIs are provided to enable third–party programs to extract and process published information.

Event Publishing

SAS Integration Technologies 9 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, message queues, or to publication channels and their associated subscribers. You can develop custom applications that receive and process events generated by the Publishing Framework.

Publishing Framework Documentation

For information about how to perform publishing tasks, or how to incorporate publishing tasks into your SAS programs or applications, refer to the Publishing Framework chapter of the SAS Integration Technologies Developer's Guide.

For information about how to administer publication channels and subscriber information, refer to Administering the Publishing Framework in the SAS Integration Technologies Administrator's Guide. If you are using an LDAP directory server as your metadata repository, refer to Administering the Publishing Framework in the SAS Integration Technologies Administrator's Guide (LDAP Version).

Technical Overview

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 wishing to communicate with another application simply sends 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.

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.

Application Messaging Interface Documentation

For complete documentation of the SAS application messaging interfaces, including CALL routine syntax and program samples, refer to the Application Messaging chapter of the SAS Integration Technologies Developer's Guide.

Technical Overview

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 that 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 result packages to clients or the mid–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:

- the Stored Process Service application programming interface (API)
- the Stored Process Web Application
- the Information Delivery Portal

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.

Note: With SAS Integration Technologies 9, stored processes can operate either on a SAS Workspace (IOM) Server to produce result packages, or on a SAS Stored Process Server to produce streaming results. IOM Direct Interface stored processes, which were introduced with SAS Integration Technologies 8, operate on a Workspace (IOM) Server and produce result packages only.

Stored Processes Documentation

For information about how to create a stored process and invoke it in a client application, refer to the Stored Processes chapter of the SAS Integration Technologies Developer's Guide.

For information about how to create the metadata to define a stored process, refer to

- Administering Stored Processes in the SAS Integration Technologies Administrator's Guide or
- Administering Stored Processes in the SAS Integration Technologies Administrator's Guide (LDAP Version).

Technical Overview

SAS BI Web Services

SAS BI Web Services enable client applications to obtain metadata about and execute SAS Stored Processes 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 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.

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.

For detailed information about using SAS BI Web Services, including instructions for creating stored processes that Web services can access, refer to the SAS BI Web Services chapter in the SAS Integration Technologies Developer's Guide.

Technical Overview

Directory Services

SAS Integration Technologies uses an enterprise directory server to host all of its product infrastructure and run–time configuration information. This includes metadata for server and transport bindings, Publishing Framework channels and subscriber profiles, result package archive repositories, and data source locators.

With SAS 9, the SAS Metadata Server is provided to serve as a central repository for metadata for the entire enterprise. To incorporate SAS Metadata Server functions into your SAS programs, you can use the SAS Open Metadata Interface. For information, see ⊕ <u>Getting Started with the SAS 9.1 Open Metadata Interface</u> and the ⊕ <u>SAS 9.1 Open Metadata Interface: User's Guide</u> on the SAS Web site).

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 <u>Directory Services</u> chapter of the SAS Integration Technologies Developer's Guide.

Technical Overview

SAS Web Infrastructure Kit

The SAS Web Infrastructure Kit, which is provided with SAS Integration Technologies, serves as the underlying infrastructure for the SAS Information Delivery Portal Web application.

The SAS Web Infrastructure Kit includes

- a Web application shell that renders content in portlets and pages and that provides log–on and log–off capabilities, metadata searching, bookmarking, and content administration features
- a SAS Stored Process application that enables running stored processes from the Web
- predefined portlets for content viewing and navigation
- a portlet development kit, which includes an API and a set of best practices, for developing custom portlets
- administrative tools for deploying services, portlets, and additional Web applications
- SAS Java components, Web infrastructure components, and a services infrastructure.

If the SAS Information Delivery Portal is installed, then the following additional capabilities are provided:

- personalization features which enable users to create and modify their own pages and portlets to meet individual needs
- the ability to subscribe to publication channels and publish content to channels or WebDAV
- support for running SAS Stored Processes in the background and receiving alert messages when processes are finished
- support for syndicated, continuously updated Web content from information providers
- (Preproduction Feature) If your organization has installed SAS Web Report Viewer and SAS Information Map Studio, then users can access SAS information maps and SAS reports using the portal.

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

- The *SAS Web Infrastructure Kit Overview*, which provides a high–level description of the features of the SAS Web Infrastructure Kit and the SAS Information Delivery Portal.
- The *SAS Web Infrastructure Kit Administrator's 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 Kit.
- The *SAS Web Infrastructure Kit Developer's Guide*, which describes how to use the kit 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 Kit 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.