

CHAPTER

1

Concepts

| | |
|---|---|
| <i>Overview of SAS Foundation Services</i> | 1 |
| <i>Understanding Service Deployments</i> | 4 |
| <i>Overview of Service Deployments</i> | 4 |
| <i>Service Dependencies</i> | 6 |
| <i>Understanding Service Deployment Configuration</i> | 6 |

Overview of SAS Foundation Services

SAS Foundation Services includes tools to enable application development and service administration for the SAS Foundation Services. Depending on the components that you choose to install, SAS Foundation Services includes one or more of the following components:

- *SAS Management Console plug-ins*, which enable you to administer configuration metadata in a metadata repository. The following plug-ins can be installed with the SAS Foundation Services:
 - *Application Monitor*, which enables administrators to monitor the performance and activities of a foundation service-enabled application.
 - *Foundation Services Manager*, which enables administrators to define and manage service deployments and service configurations.
 - *Publishing Framework Manager*, which enables administrators to set up metadata for users and applications to do the following:
 - publish SAS files to a variety of destinations
 - receive and process published information

For more information about SAS Foundation Services administration, see the online Help for the appropriate administrative plug-in.

- *SAS Foundation Services*, which is a set of platform infrastructure and extension services for programmers who want to write applications that are integrated with the SAS platform. For information about coding applications that use the SAS Foundation Services, see "Using SAS Foundation Services" in the *SAS Integration Technologies: Java Client Developer's Guide*.

The following table presents the function and related documentation for each of the SAS Foundation Services:

Table 1.1 SAS Foundation Services

| Service | Java Class | Function | Related Documentation |
|----------------------|--------------------------------------|--|---|
| Connection Service | com.sas.services.connection.platform | IOM connection management | For details about administering the SAS servers that you connect to with the Connection Service, see the <i>SAS Intelligence Platform: Application Server Administration Guide</i> . For development information and coding examples, see the "Using the Java Connection Factory" chapter in the <i>SAS Integration Technologies: Java Client Developer's Guide</i> . |
| Discovery Service | com.sas.services.discovery | locating and binding to deployed services | For details about how applications use the Discovery Service, see "Understanding How Applications Locate Foundation Services" on page 20. |
| Event Broker Service | com.sas.services.events.broker | asynchronous event notification and request management to support dynamic, event-driven processes | For details about editing the Event Broker Service configuration, see "Modifying the Session and User Service Configurations" on page 44. For information about using the Publishing Framework to generate and publish events, see <i>SAS Publishing Framework: Developer's Guide</i> . |
| Information Service | com.sas.services.information | repository federation, searching repositories, a common entity interface, and creating personal repositories | For details about editing the Information Service configuration, see "Modifying the Information Service Configuration" on page 36. |
| Logging Service | com.sas.services.logging | runtime execution tracing, response metric and resource utilization reporting, and error tracking | For details about editing the logging service configuration, see "Modifying the Logging Service Configuration" on page 40 . |

| Service | Java Class | Function | Related Documentation |
|------------------------|--------------------------------|--|---|
| Publish Service | com.sas.services.publish | access to the publication framework | For details about the Publishing Framework, see the <i>SAS Publishing Framework: Developer's Guide</i> . |
| Security Service | com.sas.services.security | user authentication, propagation of user identity context across distributed security domains, and protected-resource access policy administration and enforcement | For detailed information about implementing security in your environment, see the <i>SAS Intelligence Platform: Security Administration Guide</i> . |
| Session Service | com.sas.services.session | context management, resource management, and context passing | For details about editing the session service configuration, see "Modifying the Session and User Service Configurations" on page 44. |
| Stored Process Service | com.sas.services.storedprocess | access to stored process execution and package navigation | For details about stored processes, see the <i>SAS Stored Processes: Developer's Guide</i> . |
| User Service | com.sas.services.user | access to authenticated user context, access to global, solution-wide, and application-specific profiles, and access to personal objects | For details about editing the User Service configuration, see "Modifying the Session and User Service Configurations" on page 44. |

In addition, you use the deployment utilities (com.sas.services.deployment) to deploy the services.

This document covers the following SAS Foundation Services topics:

service deployments

In order to use the foundation services in your applications, you must deploy the services. To deploy the services, you must configure a service deployment. To understand service deployments and service deployment configuration, see "Understanding Service Deployments" on page 4 and "Understanding Service Deployment Configuration" on page 6.

service deployment definitions

To define and manage service deployments, see the Managing Service Deployments topics.

installing and running SAS Foundation Services as a Windows service

The Java Service Wrapper from Tanuki Software is provided with SAS Foundation Services. You can use this software to install and run SAS Foundation Services as a Windows service for use with any application that uses SAS Foundation Services. For details, see “Overview of Installing and Running Foundation Services as a Windows Service” on page 15.

applications that are enabled for SAS Foundation Services

To understand how applications deploy, locate, and share services, see the following topics:

“Understanding How Applications Deploy Foundation Services” on page 19

“Understanding How Applications Locate Foundation Services” on page 20

“Understanding How Applications Share SAS Foundation Services” on page 26

service configurations

To understand how to modify the configurations of certain foundation services, see “Overview of Modifying Service Configurations” on page 27.

Understanding Service Deployments

Overview of Service Deployments

A service deployment is a configuration of a collection of SAS Foundation Services that specifies the data that is necessary to instantiate the services, as well as dependencies on other services. You create service deployments for applications that deploy or access the services. You can store the service deployment configuration in the following locations:

SAS Metadata Repository

You can use the Foundation Services Manager plug-in (of SAS Management Console) to administer service deployment metadata that is stored in a SAS Metadata Repository. The SAS Metadata Server controls access to the metadata.

XML file

You can export service deployment metadata from the SAS Metadata Server to an XML file. You can then use the XML file to import service deployment metadata into another SAS Metadata Repository. If you use an XML file to store service deployment metadata, then there is no administration or access control for the metadata in the XML file.

container deployment file:

You can export service deployment metadata from the SAS Metadata Repository to a container deployment file. You can then deploy the SAS Foundation Services configuration in a Spring Framework container.

Note: It is recommended that you store the service deployment metadata on a SAS Metadata Repository. Storing the service deployment metadata in a SAS Metadata Repository enables it to be updated and queried from one centralized location. \triangle

To enable your application to deploy and access the foundation services, you can create local or remote service deployments:

local service deployment

supports exclusive access to a set of services deployed within a single Java Virtual Machine (JVM). Use a local service deployment when you want your application to have its own exclusive set of foundation services.

remote service deployment

supports shared access to a set of services that are deployed within a single JVM, but are available to other JVM processes. Use a remote service deployment when you want to share a foundation service deployment among multiple applications. When you create services for remote service deployments, you must specify that the services will be accessed remotely. In order to allow remote access to the services, you must also create a service registry and associate named services with the named components for the remote services.

A service deployment contains:

service deployment groups

When you create service deployments (local or remote), you can also create groups within the service deployment in order to organize services within a deployment hierarchy.

services and service initialization data

Within each service deployment group, you must define the services for that group. Service definitions contain the following information:

service types (interfaces)

designate which service interfaces are implemented by the service. The Discovery Service locates services according to their service interfaces. For example, if you want to locate a service that implements a Logging Service interface, have the Discovery Service search for a service that implements **com.sas.services.logging.LoggingServiceInterface**.

Note: All SAS Foundation Services (including local services) implement the Remote Service interface. Δ

service configuration

specifies the Java class that is used to create the service, the service's optional configuration data, and the service's configuration user interface. The service configuration user interface defines the Java class used by the Foundation Services Manager to configure the service's configuration details.

service dependencies

specify other services on which the service depends. When they are deployed, foundation services might depend on the availability of one or more other foundation services. When you define a service, you must specify the other services on which that service depends. For example, the Authentication Service uses the Logging Service. Therefore, when you define the Authentication Service in a service deployment, you must specify the Logging Service as a dependency.

service names (for remote access only)

specify named services for remote access. If a foundation service is to be made available for remote clients, you must enable the service for remote access and define named services (service names) that specify the service's name bindings to one or more service registries.

consumers (application configurations)

group the resources that are associated with a specific application. If several applications share a single service configuration, then you can group the resources in your service configurations by defining consumers. Consumers enable you to install and uninstall the specific parts of a service configuration that are associated with a specific application. For example, if your Logging Service configuration contains a renderer that is specific to SAS Web Report Studio and you uninstall SAS Web Report Studio, then the consumer

definition can be used to remove that specific renderer while leaving the rest of the Logging Service configuration intact.

authorization permissions

enable you to specify which user or group identities can perform which actions on a particular resource.

Note: If a service depends on other services, then you must define those services before defining the service that depends on them. For details about service dependencies and order of definition, see “Service Dependencies” on page 6. △

service registries and associated named services (remote service deployments only)

To enable services for remote access, you must define a service registry to use in locating remote services. (A service registry is a searchable registry of service descriptions that is used to register named service bindings). You must then register the services with the service registry by creating or associating named services that define how each service is to be used within the context of the Discovery Service.

To understand where service deployments are defined, see “Understanding Service Deployment Configuration” on page 6.

Service Dependencies

If a service has a dependency on another service, you must first create the service on which it depends. The following table shows the service dependencies and the relative order in which you must define the services.

Table 1.2 Service Dependencies Table

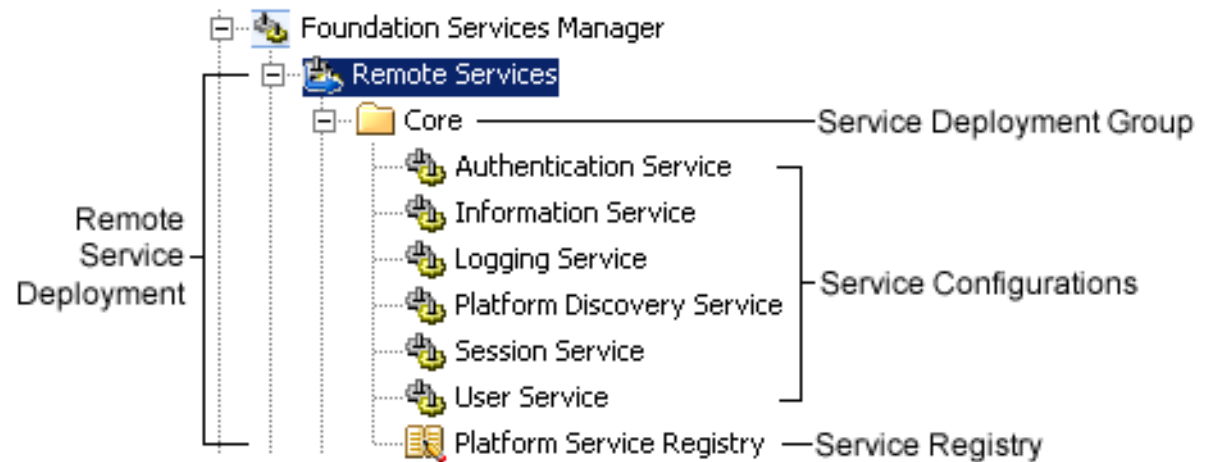
| Service | Service Dependencies |
|------------------------|---|
| Logging Service | |
| Authentication Service | Logging Service |
| Information Service | Logging Service |
| User Service | Logging Service Authentication Service Information Service |
| Session Service | Logging Service |
| Discovery Service | Logging Service |
| Event Broker Service | Logging Service Authentication Service Information Service User Service Session Service |
| Stored Process Service | Logging Service |

Understanding Service Deployment Configuration

The following figure shows the SAS Management Console Foundation Services Manager connected to a SAS Metadata Repository that contains a service deployment

named Remote Services. The figure also points to the service deployment group, services, and service registry defined within the Remote Services service deployment.

Figure 1.1 SAS Foundation Services Deployment



You can define a service deployment in a SAS Metadata Repository in one of the following ways:

- Use the Foundation Services Manager Plug-in of SAS Management Console to create a service deployment. For details, see “Defining Service Deployments” on page 9.
- Import an XML file that contains the service deployment. If your application’s service deployment configuration is contained in an XML file, you can import it into a SAS Metadata Repository. For details, see “Importing Service Deployments” on page 12.

After you import or create a service deployment, you can do the following:

- Export the service deployment to an XML file. If an application does not have access to a SAS Metadata Repository in its runtime environment, you can export the service deployment configuration to an XML file that the application can access for service deployment configuration information. For details, see “Exporting Service Deployments” on page 12.
- Export the service deployment to a container deployment file. You can export the service deployment configuration to a container deployment file that can be deployed in a Spring Framework container. For details, see “Exporting Service Deployments” on page 12.
- Duplicate the service deployment. If you need to use a service deployment that is similar to an existing service deployment, you can duplicate an existing service deployment configuration. For details, see “Duplicating Service Deployments” on page 13.

In addition, you might need to update the prototypes that define the foundation services. (To update prototypes, select the Foundation Services Manager and select **Actions ► Update Prototypes**). For further information about using the Foundation Services Manager to create service deployments, see the Foundation Services Manager Help.

