SAS® Viya™ 3.2: Platform Overview

Architecture of SAS Viya

Introduction

This section provides a concise summary for new administrators.

Here are related topics:

- To get started with SAS Viya administration, see *SAS Viya Administration: Orientation*.
- To learn about benefits of SAS Viya, see SAS Viya on the SAS website.

Key Components

Here are software components that might be of particular interest to administrators.

The analytics engine to SAS Viya	*SAS Cloud Analytic Services: Fundamentals*
A modular set of supporting services	*SAS Viya Administration: General Servers and Services*
A web application for basic administration	*SAS Viya Administration: Using CAS Server Monitor*
A web application for enterprise administration	*SAS Viya Administration: Using SAS Environment Manager*
A web application for writing and submitting code	*Getting Started with Programming in SAS Studio*
A web application for visual reporting, exploration, and modeling	*SAS Visual Analytics: Overview*
Multiple application programming interfaces	*Getting Started with CASL* *Getting Started with SAS Viya for Lua* *Getting Started with SAS Viya for Java* *Getting Started with SAS Viya for Python*

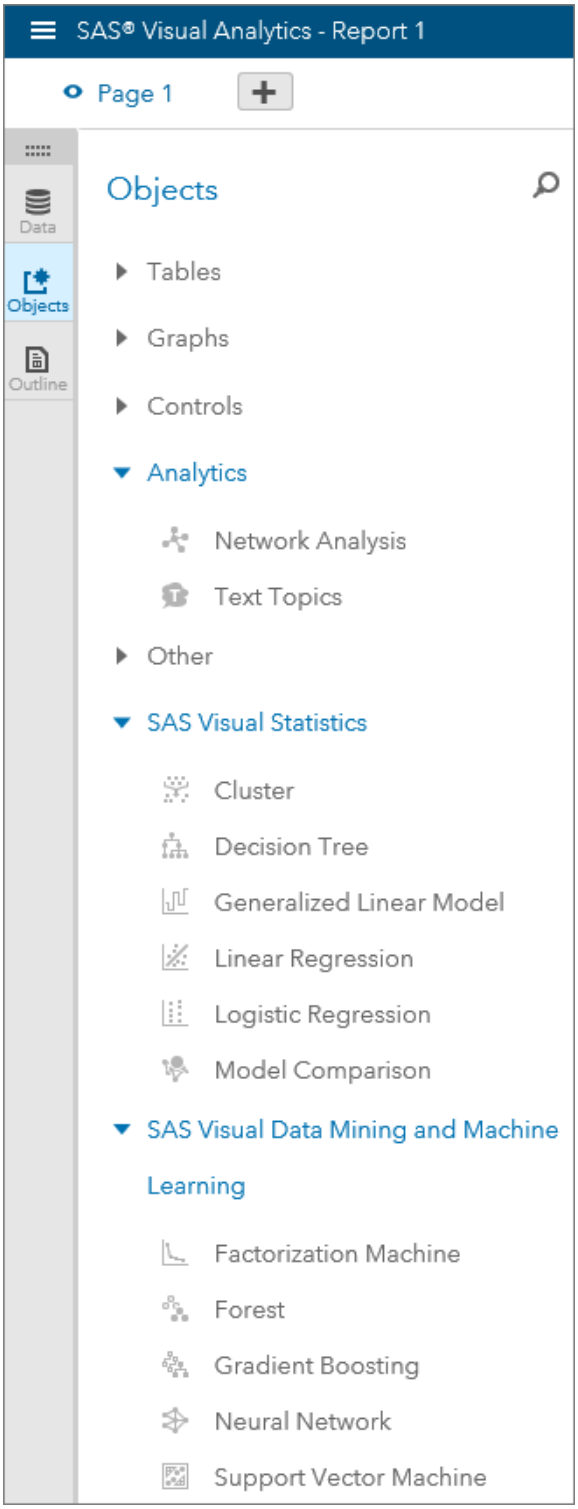
TIP For information about other components, search the SAS Viya Administration documentation.

Cumulative Functionality

Among the core products on SAS Viya, available functionality is cumulative.

- SAS Visual Analytics provides baseline functionality, including reporting and basic analytics.
- SAS Visual Statistics provides an additional set of advanced analytic functions.
- SAS Visual Data Mining and Machine Learning provides a second additional set of advanced analytic functions.

For example, if you have SAS Visual Data Mining and Machine Learning, the objects that are available in the SAS Visual Analytics web application are as follows:



Note: All three of the core products offer both programming and visual interfaces.

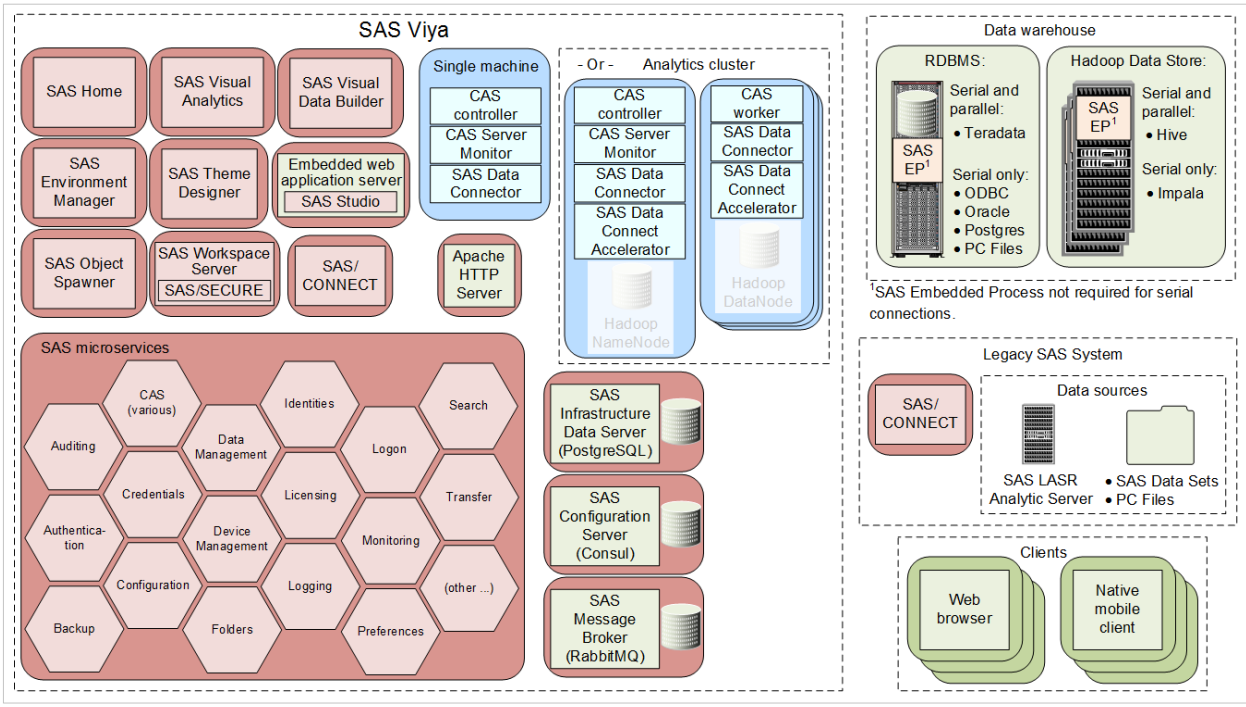
Selective Deployment (Optional)

By default, all of your software is deployed. As a convenience for special circumstances, it is possible to deploy only a subset of components.

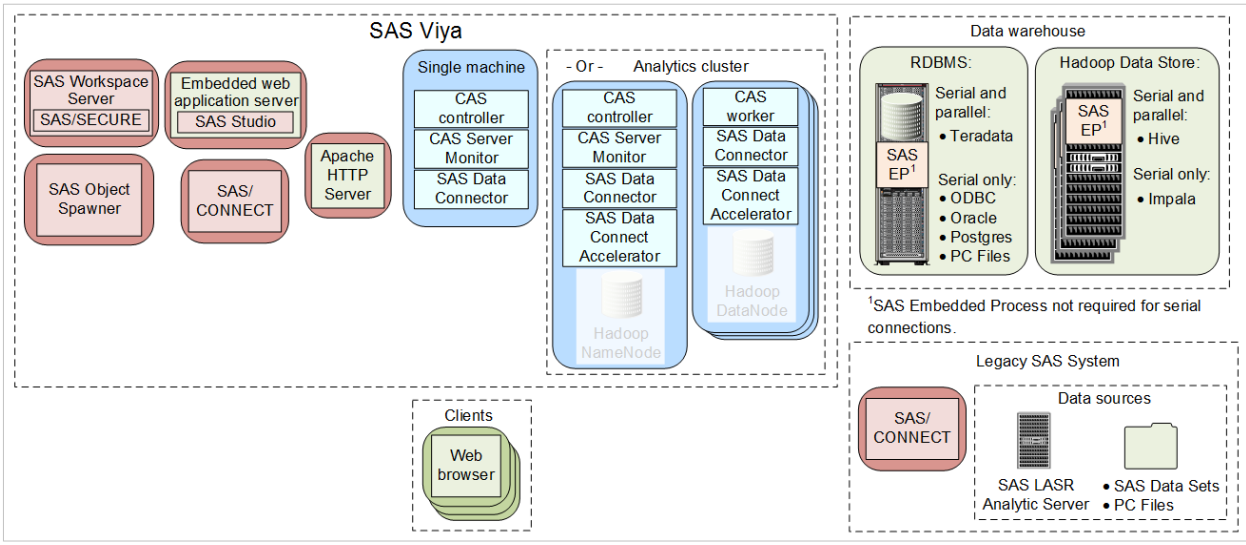
- A programming-only deployment excludes general services and visual interfaces. For example, a programming-only deployment of SAS Visual Analytics does not include the SAS Visual Analytics web application.
- A visual-only deployment excludes SAS Studio and most SAS programming functionality.

Diagrams by Deployment Type

Full Deployment (Native Operating Systems)

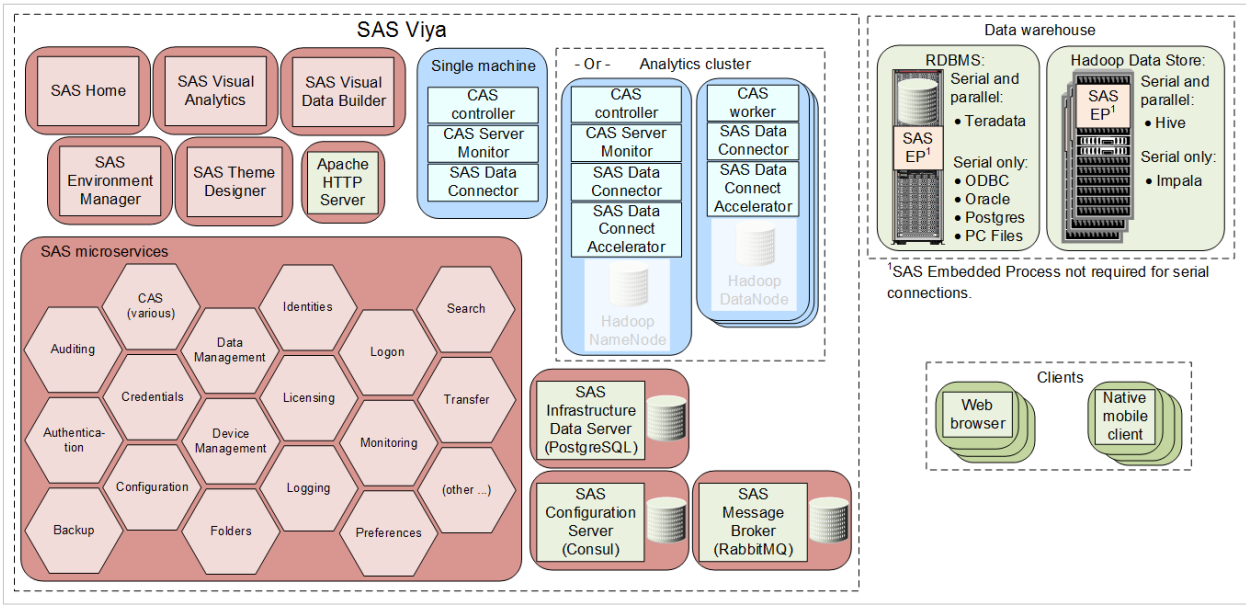


Programming-Only Deployment (Native Operating Systems)



Visual-Only Deployment

Note: Visual-only deployments are available only on native operating systems.



Security in SAS Viya

Authentication

Authentication is the aspect of security that verifies the identity of a user or service account.

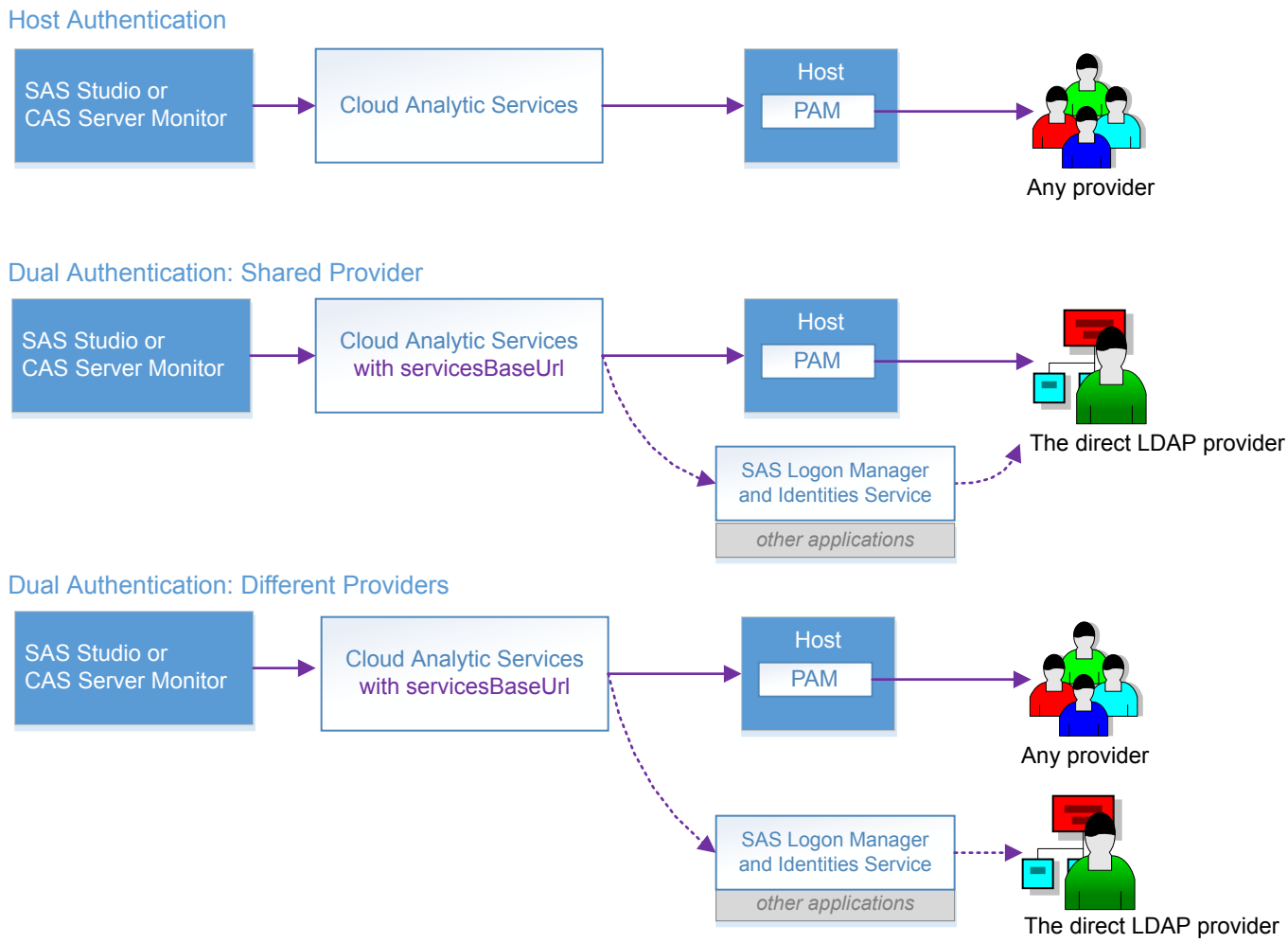
When you sign in, one of the following authentication patterns is used:

Pattern	Description	Usage
Host authentication	Requests are sent to the appropriate host and processed by any authentication mechanism supported by that host. Programming-only deployments use this pattern exclusively. Other deployments use dual authentication for sign in to CAS Server Monitor and access to CAS from SAS Studio. **Note:** You can configure the host to use pluggable authentication modules (PAM). SAS provides starter PAM configuration files for CAS and SAS Studio. You can create an Authinfo file for use with PAM in command-line access to CAS.	When you sign in to SAS Studio, the associated object spawner asks its host (which is also the host of the SAS Studio web application) to validate your credentials. That validation enables the object spawner to launch a workspace server for you. When you access CAS from SAS Studio, you must authenticate to the host of the target CAS server. When you sign in to CAS Server Monitor, you must authenticate to the host of the target CAS server.

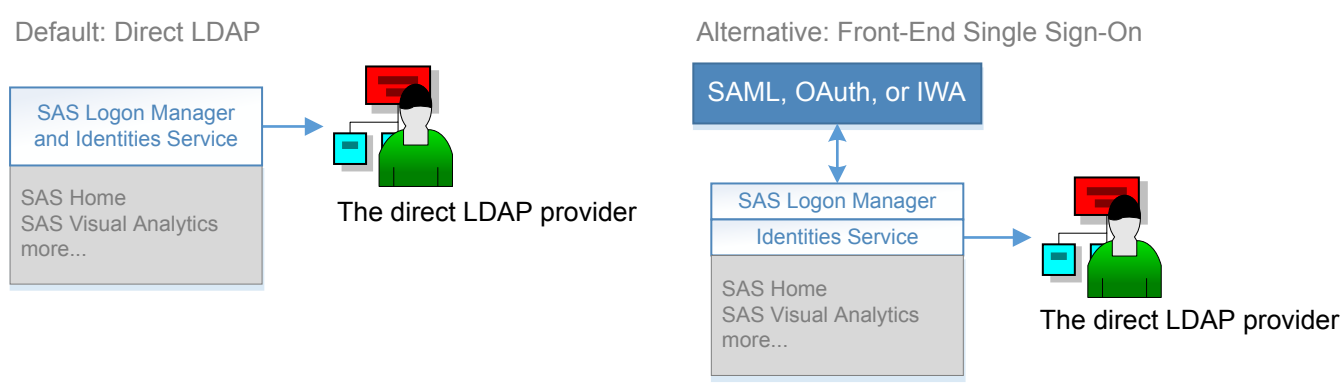
Pattern	Description	Usage
Direct LDAP authentication	Requests are sent to and processed by your designated direct LDAP provider, unless you configure front-end single sign-on using Integrated Windows Authentication (IWA), Open Authorization (OAuth), or Security Assertion Markup Language (SAML). User and group information is always obtained from your designated direct LDAP provider. **Note:** IWA, OAuth, and SAML are alternate mechanisms for identity verification by the logon service, not alternate sources of user and group information for the identities service.	When you sign in to a web application that uses the logon service (for example, SAS Visual Analytics or SAS Environment Manager), you must authenticate using this pattern. Before you can submit a command-line request to a general service (for example, the backup service or the transfer service), you must authenticate using this pattern.
Dual authentication	Requests are authenticated using both host authentication and direct LDAP authentication. If the servicesBaseUrl option is specified, CAS requires dual authentication. To facilitate this pattern, use one of these approaches: ■ Ensure that all requests are ultimately processed by the same authentication provider. For example, configure the SAS Studio and CAS hosts to use the same LDAP provider that is designated for direct LDAP authentication requests in your deployment. ■ Ensure that each affected user has a single set of credentials that are valid for all applicable authentication providers.	In a full deployment, dual authentication occurs for sign in to CAS Server Monitor and access to CAS from SAS Studio. In a visual-only deployment, dual authentication occurs for sign in to CAS Server Monitor. **Note:** When you access CAS from a web application such as SAS Visual Analytics or SAS Environment Manager, your OAuth token is validated.

The following high-level conceptual drawings illustrate key points from the preceding table:

Authentication from SAS Studio or CAS Server Monitor to CAS



Authentication from Other Applications



After you sign in, you have seamless access to SAS Viya and, in some contexts, to external data sources.

For more information, see the following documents:

SAS Viya Administration: Authentication

SAS Viya Administration: Identity Management

SAS Viya Administration: External Credentials

Authorization

Authorization is the aspect of security that determines which resources are available to which users. The SAS Viya authorization layer consists of two authorization systems:

- CAS authorization system
- general authorization system

Each system uses a distinct model to protect a distinct class of resources. The general authorization system is not applicable in a programming-only deployment.

Initial and default access are restrictive:

- Any access that is not granted is implicitly disallowed.
- Predefined objects are protected by predefined rules or access controls.
- Only members of special groups or roles have access to privileged administrative functionality.
- Access to objects that users add is managed by inheritance, other influencing rules, and any direct settings.
- Regular users have limited Write access. They can write to their personal folder, the shared Public folder, and the shared Public caslib. They can create top-level folders.

  Note: To enable only administrators to create top-level folders, see "Restrict Creation of Top-Level Folders" in SAS Viya Administration: General Authorization.

For more information, see the following documents:

SAS Viya Administration: Orientation to Authorization

SAS Viya Administration: Cloud Analytic Services Authorization

SAS Viya Administration: General Authorization

SAS Viya Administration: Identity Management

Encryption

Encryption is the aspect of security that protects data by converting it into an unintelligible form in transmission or in storage.

For data in motion in a new deployment, TLS security is provided and follows the highest standards. At installation SAS Viya provides self-signed certificates to provide HTTP and HTTPS access to SASHome out of the box. You can increase the encryption strength and coverage by completing additional configuration.

For data at rest in a new deployment, encryption is not automatically enabled. You can configure encryption of data that is added to PATH, HDFS, and DNFS caslibs.

For more information, see the following documents:

Encryption in SAS Viya: Data in Motion

Encryption in SAS Viya: Data at Rest

SAS 9 and SAS Viya

Coexistence of SAS 9 and SAS Viya

In most cases, SAS Viya customers who also have SAS 9 run both platforms in parallel. This approach provides the following advantages:

- Users can continue to benefit from their investment in SAS 9 as they begin to make use of SAS Viya functionality and features.
- From within familiar SAS 9 interfaces, projects, and code, users can access the performance enhancements that SAS Viya provides.

Integration of SAS 9 and SAS Viya

Here are preliminary considerations for accessing SAS 9 data from the SAS Viya platform:

- SAS Viya operates with UTF-8 encoded data. If your SAS 9 encoding is not UTF-8, you might need to re-create your data sets. See *Migrating Data to UTF-8 for SAS Viya*.
- Any user-defined formats must be available to your CAS session. See *SAS Cloud Analytic Services: User-Defined Formats*.

Here are approaches to accessing SAS 9 data from the SAS Viya platform:

- If you have the SAS Visual Analytics 8.1 web application, you can use self-service import. See *SAS Viya: Self-Service Import*.
- If you have SAS Environment Manager 3.1, you can use that web application to add caslibs and load data. See *SAS Viya Administration: Data*.
- If you have the SAS Viya 3.2 programming interfaces, you can write code to load data. See *SAS Cloud Analytic Services: Accessing and Manipulating Data*.
- If you have SAS/CONNECT for SAS 9 and SAS Viya, you can move and share data between SAS 9 and SAS Viya environments. See Sharing Data Between SAS 9 and SAS Viya Using SAS/CONNECT on page 10.
- • If you have SAS Enterprise Guide or SAS Add-In for Microsoft Office 7.13 or later, you can use the Upload to CAS task to move data from SAS 9 to CAS. For more information about how to configure the Upload to CAS task for SAS Enterprise Guide and SAS Add-In for Microsoft Office, see Administering SAS Enterprise Guide or Administrating the SAS Add-In for Microsoft Office in the *SAS 9.4 Intelligence Platform: Desktop Application Administration Guide*.

Related Documents

SAS Cloud Analytic Services: Fundamentals

Differences in the SAS 9 and SAS Viya Platforms

An Introduction to SAS Viya Programming

Sharing Data Between SAS 9 and SAS Viya using SAS/CONNECT

When a new release of SAS software becomes available, users typically either continue running their current version of SAS software or they choose to upgrade and migrate to the new release. That changes with SAS Viya. With SAS Viya,customers are expected to continue running their current version of SAS software along with SAS Viya. This allows users to continue to benefit from all of the features, functionality, and investment in their current SAS deployment as well as to take advantage of the SAS Data Mining and Machine Learning algorithms and other new features of SAS Viya.

SAS/CONNECT as a Bridge between SAS 9.4 and SAS Viya

How can SAS 9.4 users integrate the SAS Viya platform with their existing SAS environments? They can use SAS/CONNECT to create a virtual bridge between the two environments.

SAS/CONNECT software is a SAS client/server toolset that provides the ability to manage, access, and process data in distributed and parallel SAS environments. As a client/server application, SAS/CONNECT links a SAS client session to a SAS server session. The terms *client* and *server* depict the relationship between two SAS sessions. The client session is the initial SAS session that creates and manages one or more server sessions. The server session can run either on the same computer as the client or on separate hardware, such as on a remote computer across a network. You can use SAS/CONNECT statements to create a software connection, or bridge, to facilitate interoperability between SAS 9 and SAS Viya environments. The connection enables features such as:

- transferring disk copies of data
- directly processing remote data sources and getting results back locally
- using local graphical user interfaces to process data sources remotely
- running multiple independent processes asynchronously
- combining resources from multiple computers to work in parallel

SAS/CONNECT becomes the bridge that enables you to move and share data and computing resources across environments. You can move data seamlessly and leverage the robust capabilities of SAS Viya from existing SAS 9 environments. You can continue to use your SAS 9 projects and custom code while accessing SAS Viya and its new analytic algorithms.

While working in the SAS Viya programming environment, you can access data in your SAS 9 environment and transfer it directly into memory in SAS Viya. SAS/CONNECT supports all SAS releases, so you can move and share data and computing resources between any SAS deployment and SAS Viya.

SAS/CONNECT Requirements

To use SAS/CONNECT with your SAS 9 and SAS Viya environments, it must be licensed in both environments. SAS 9.4 customers should check with their SAS software administrator: Many current SAS 9.4 customers already have licenses for SAS/CONNECT in their existing environments. SAS Visual Data Mining and Machine Learning in its default configuration on SAS Viya does not include licensing for the SAS/CONNECT utility. For such installations, SAS/CONNECT for SAS Viya needs to be licensed separately.

Encoding Compatibility between SAS/CONNECT Client and Server Sessions

To successfully use SAS/CONNECT programming services, the encodings of the client and server sessions must be compatible. In the SAS Viya environment, the default session encoding is UTF-8. In the SAS/CONNECT 9.4 server environments, the default session encoding is LATIN1. Transport data has an encoding family dependency, so the encodings of the client and server sessions should be compatible to ensure that the data is not corrupted during transmission. Compatible encodings share a common character set. For example, client and server sessions that each use the UTF-8 encoding should be compatible with each other.

If one session's encoding is not compatible with the other session's encoding, then SAS issues a note stating that data might not have been transmitted correctly. In this example, the SAS Viya client is signing on to a SAS 9.4 system:

```
signon host.9650 user=&user pwd=&pwd
NOTE: Remote signon to HOST.9650 commencing (SAS Release V.03.00M0P050516).
NOTE: FIPS validated AES encryption is being used to protect network traffic.
NOTE: The client session encoding utf-8 does not match the server session
encoding latin1. This may produce errors when moving some character data.
Search "SAS/CONNECT Encoding Compatibility" for details.
NOTE: Unable to open SASUSER.PROFILE. WORK.PROFILE will be opened instead.
NOTE: All profile changes will be lost at the end of the session.
NOTE: Copyright(c) xxxx SAS Institute Inc., Cary, NC 27513-2414, U.S.A.
NOTE: SAS (r) Proprietary Software Version 7 (TSmm.xxx)
Licensed to SAS Institute Inc. Host Testing, Site 00000001.
NOTE: Remote signon to HOST.9650 complete.
```

If one session is using UTF-8 and the other session has an unknown, or unsupported, encoding, an error occurs and the connection is not made.

Leveraging SAS 9.4 Solutions and SAS Viya

SAS Viya is designed to co-exist with SAS 9 and SAS 9.4 solutions. This design enables you to access the performance features of the SAS Viya environment from within familiar SAS 9.4 solution interfaces, projects, and SAS code.

Following are several examples of SAS 9.4 solutions that contain user interface features that you can use to issue SAS/CONNECT statements to bridge the SAS 9.4 and SAS Viya environments:

SAS Enterprise Miner
: From your SAS Enterprise Miner process flow diagram running on SAS 9.4, you can remotely submit SAS Data Mining and Machine Learning procedures to run in SAS Viya and then return the results to SAS 9.4 and integrate them into the SAS Enterprise Miner output.

SAS Model Manager
: You can run SAS Data Mining and Machine Learning procedures and train models in your SAS Viya environment, and then upload the models and the models' metadata to SAS 9.4. From SAS 9.4, you can remote submit code to register the SAS Viya created model and metadata in SAS Model Manager. You can use SAS Model Manager to manage both your SAS 9.4 models and your SAS Viya models. For example, you can use SAS Model Manager to deploy score code derived from both SAS 9.4 and SAS Viya models to a database.

SAS Data Integration Studio
: You can use SAS Data Integration Studio on SAS 9.4 to perform tasks such as fetching data from multiple sources and preparing the data for mining. Then, you can use the Data Transfer node to upload the data table to SAS Viya and load it directly into memory.

SAS Enterprise Guide / SAS Add-In for Microsoft Office

You can use the custom code nodes in either of these products to upload or download data between the two environments. For example, you could remote submit code to SAS Viya for processing, and then return the computational results back to SAS Enterprise Guide on SAS 9.4.

You can also use the Upload to CAS task in these products to upload data to SAS Viya. For more information about how to configure the Upload to CAS task for SAS Enterprise Guide and SAS Add-In for Microsoft Office, see Administering SAS Enterprise Guide or Administrating the SAS Add-In for Microsoft Office in the *SAS 9.4 Intelligence Platform: Desktop Application Administration Guide*.

SAS Studio

You can use the SAS Studio web interface to perform many programming actions in SAS Viya. For example, you could enter code in SAS Studio to access data in your SAS Viya environment, remotely submit SAS Viya procedures to run on the data, and then transfer the results data between the two environments.

SAS Stored Processes

You can save SAS/CONNECT SIGNON, RSUBMIT, UPLOAD, and DOWNLOAD statements as SAS Stored Processes, and then use the SAS Stored Processes as another way to bridge your SAS 9.4 and SAS Viya environments.

SAS Display Manager

You can use the SAS Display Manager (or SAS batch jobs launched using SAS Display Manager) as an interface to submit SAS/CONNECT SIGNON, RSUBMIT, UPLOAD, and DOWNLOAD statements to establish a bridge between your SAS 9.4 and SAS Viya environments.

Example Data Processing Tasks Enabled by the SAS/CONNECT Bridge

Here are some examples of typical data processing tasks that leverage performance by using SAS/CONNECT to utilize both SAS 9.4 and SAS Viya environments:

- You can connect from a SAS 9 environment to SAS Viya. For example, a SAS Enterprise Miner 14.1 user running on SAS 9.4 might create a process flow diagram that performs data preparation, visualization, and partitioning, and then uses a SAS Code node to transfer the partitioned input data into the SAS Viya environment for further processing in a massively parallel environment.
- You can use SAS Viya to complete processing instructions issued from within a SAS 9 solution. For example, a SAS Enterprise Guide process flow diagram might use a SAS Code node and SAS/CONNECT to transfer prepared input data sources to the SAS Viya environment. Data in the SAS Viya environment is processed according to the instructions submitted by the SAS 9 solution, while exploiting massive parallel processing when possible.
- You can connect from a SAS Viya environment to a SAS 9 environment. For example, the results and metadata from running a massively parallel modeling algorithm on a table in SAS Viya can be seamlessly loaded back into a SAS Enterprise Miner session running on SAS 9.4. The SAS Enterprise Miner process flow diagram transparently integrates the uploaded data from SAS Viya for results visualizations and further data mining operations in the SAS 9.4 environment.
- You can run multiple parallel SAS processes on a table within a single SAS Viya environment. By using the SASCMD sign-on feature of the SAS/CONNECT bridge, you can run multiple procedures or multiple DATA steps in parallel on a table that is loaded into SAS Viya memory.

Starting and Using SAS/CONNECT with SIGNON and RSUBMIT Statements

SAS/CONNECT is a toolset that can connect different SAS environments, enabling access to all of the SAS computing resources on your network. You can use any SAS 9.4 solution's code interface to run SAS/CONNECT, including SAS solution program editors, SAS solution start-up (autoexec.sas) files, or built-in

solution interfaces such as SAS Enterprise Miner's SAS Code node or the User Submitted Code feature in SAS Data Integration Studio.

The following examples use SAS Studio to submit SAS/CONNECT statements to make a connection between SAS 9.4 and SAS Viya. When SAS Studio starts, it requests a SAS workspace server through the object spawner. A SAS/CONNECT spawner must also be running in both environments. When a SAS/CONNECT client uses a SIGNON statement, the SAS/CONNECT client contacts the SAS/CONNECT spawner. The SAS/CONNECT spawner then initiates a SAS/CONNECT server session in the remote environment. Once the server session is running in the remote environment, code can be submitted to the remote environment using RSUBMIT statements.

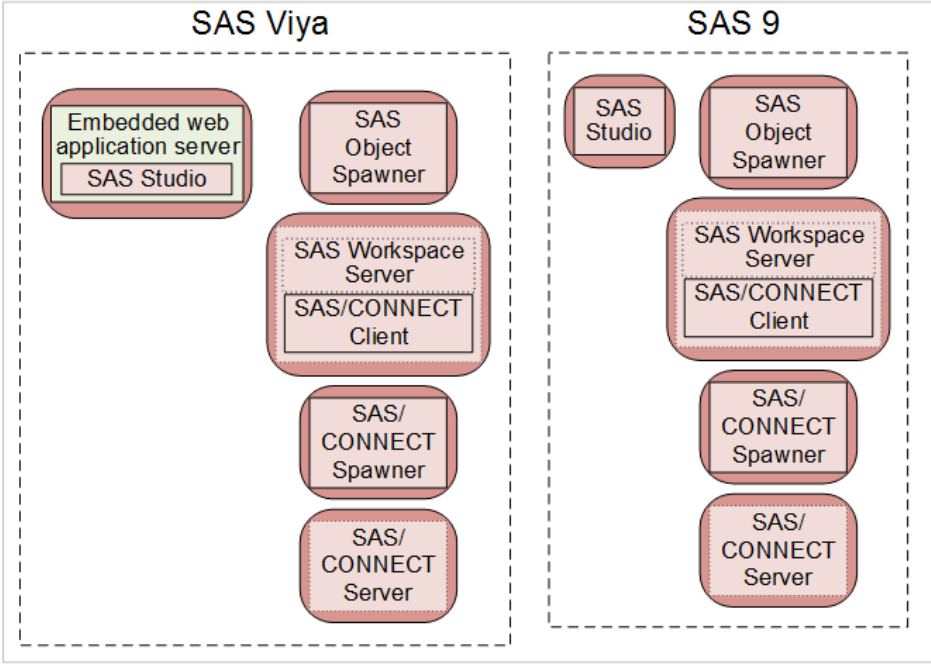
You use RSUBMIT statements to direct the execution of SAS programs to individual server sessions. A *remote submit block* is a collection of statements nested between RSUBMIT and ENDRSUBMIT commands. RSUBMIT blocks are executed in the remote environment session. When RSUBMIT block statements are executed in the remote environment, generated output and statement results are returned for display within the client session.

Verifying SAS/CONNECT between SAS 9.4 and SAS Viya

To verify connections between SAS 9.4 and SAS Viya, SAS/CONNECT must be licensed in both environments, and a SAS/CONNECT spawner must be running in any environment that will run a SAS/CONNECT server session. For example, you might have a Windows 2012 machine running SAS 9.4, with SAS Visual Analytics, SAS Visual Statistics, SAS Enterprise Miner, and SAS/CONNECT. Suppose the Windows machine is networked with a Red Hat 7.2 machine, running SAS Viya with SAS Visual Data Mining and Machine Learning and SAS/CONNECT.

The following diagram shows the key elements used for testing SAS/CONNECT between SAS 9.4 and SAS Viya. The multitude of various SAS 9.4 solution components are omitted for clarity. The SAS LASR Analytic Server is not used in testing. It is shown only to indicate its role as a functional predecessor for the SAS Cloud Analytics Services (CAS) Server.

Products Involved in Bridging SAS Viya and SAS 9



The following tests are suggested to confirm SAS/CONNECT functionality between a SAS 9 session and a SAS Viya session:

1 Test SAS/CONNECT within the SAS Viya machine

2 Test SAS/CONNECT from SAS 9.4 to SAS Viya, and upload data to the CAS server

3 Test SAS/CONNECT from SAS Viya to SAS 9.4, and download data to the CAS server.

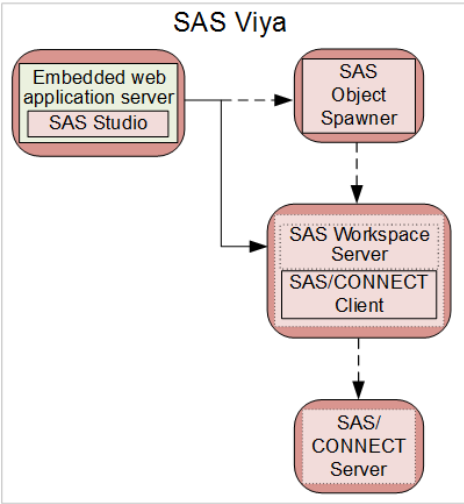
Example Code to Test SAS/CONNECT within SAS Viya

Confirm that SAS/CONNECT works within an environment before relying on the connection to other environments. This example uses SAS Studio in SAS Viya to connect to the machine in SAS Viya that is hosting the SAS Visual Data Mining and Machine Learning environment.

When SAS Studio starts, it requests a SAS workspace server using the object spawner. Once started, the test code is executed in the workspace server, which in turn executes a SIGNON statement to invoke a SAS/CONNECT server. Once the SAS/CONNECT server is running, the code for the task to be performed is sent to the SAS/CONNECT server using RSUBMIT statements.

The following diagram shows the flow:

Testing SAS/CONNECT within SAS Viya




```
/* Initial SAS/CONNECT session using SASCMD */

   signon myserver sascmd="!sascmd";
   rsubmit;

/* Display contents of CARS dataset in SAS/CONNECT */

   PROC CONTENTS data=sashelp.cars;
   run;

   data carssub ;
   set sashelp.cars ;
     if cylinders < 8 ;
    run;

/* Print subsetted results */

   PROC PRINT data=carssub;
    run;
   endrsubmit;
```

```
signoff ;
```

The code simply performs a PROC CONTENTS to display the SASHELP.CARS data table, runs a DATA step to subset it, and then uses PROC PRINT to display the output data set. If the connection is successful, you see the following message in the SAS log on the SAS Viya machine.

```
NOTE: Remote signon to MYSERVER commencing (SAS Release V.03.00M0P050516).
NOTE: Copyright (c) 2016 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software V.03.00 (TS M0 MBCS3280)
      Licensed to 16w20 SAS Viya DMML - LAX, Site 70068118.
NOTE: This session is executing on the Linux 3.10.0-327.10.1.el7.x86_64 (LIN X64) platform.
```

Sample Code to Test SAS/CONNECT from SAS 9.4 to SAS Viya

After confirming that SAS/CONNECT works within the SAS Viya environment, you can test across environments by submitting code from SAS 9.4 to the SAS Visual Data Mining and Machine Learning machine on SAS Viya. The code will load the data into SAS Cloud Analytic Services (CAS) and perform some basic data tasks.

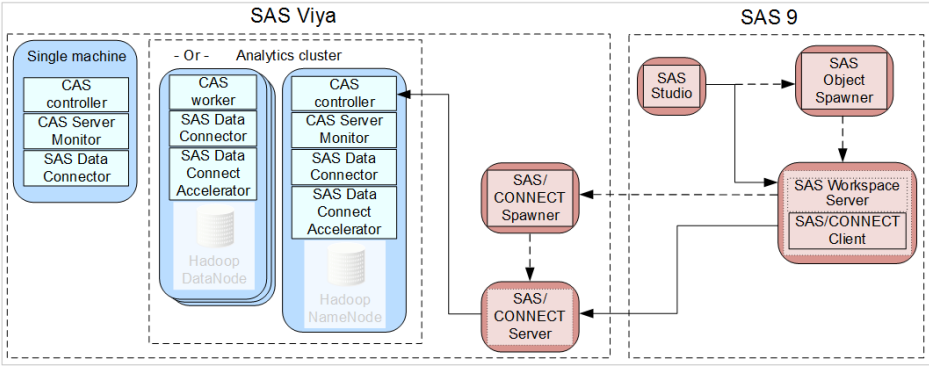
This example uses SAS Studio in a SAS 9.4 environment. When SAS Studio starts, it requests a SAS 9.4 workspace server using the object spawner. After the workspace server starts, it executes the test code and starts SAS/CONNECT client software. The SAS/CONNECT client in the SAS 9.4 environment submits a SIGNON statement to the SAS/CONNECT spawner in the SAS Viya environment, requesting a connection to the SAS Visual Data Mining and Machine Learning machine. Once the SAS/CONNECT server is running, a connection is established between machines in both environments. Code for the task to be performed is sent using RSUBMIT statements in an RSUBMIT block.

In this example, code submitted in an RSUBMIT block does the following:

- invokes the creation of a SAS Cloud Analytic Services (CAS) library through a CAS session
- uploads a table from SAS 9.4 into CAS memory
- performs basic analytic tasks on the CAS table.
- writes analytic output into CAS server memory.

The following diagram shows the servers that are involved and the initial flow of communications.

Testing SAS/CONNECT: SAS 9 to SAS Viya



The example test code is as follows:

```
/* Create copy of HEART in SAS 9.4 WORK library */

   PROC COPY in=sashelp out=work ;
```

```
   SELECT heart ;
   run;

/* Initial SAS/CONNECT session in Viya with credentials */
/* Change <myHost.myDomain.com> and <port> below to the */
/* host and port of the SAS/CONNECT spawner             */

   %let myserver=<myHost.myDomain.com> <port>;
   SIGNON myserver user=sasdemo passwd="myPassword";
   rsubmit;

/* Allocate CAS library named MYCAS as sasdemo */

   libname MYCAS CAS
     caslib=”CASUSER”
     host=”viyaserver01.race.sas.com”
     port=5570 ;

/* Upload HEART dataset from SAS 9.4 WORK to CAS library */

   PROC UPLOAD data=heart
   out=mycas.heart94;
   run;

/* Perform simple CAS analytics */

   PROC MDSUMMARY data=mycas.heart94 ;
     GROUPBY deathcause;
     VAR cholesterol systolic diastolic ;
   output out=mycas.heartsum94;
   run;

/* Verify CAS datasets in CAS library */

   PROC DATASETS lib=mycas;
   run;

   PROC PRINT data=mycas.heartsum94; run;
endrsubmit ;
signoff ;
```

The RSUBMIT code block creates a CAS library named MYCAS, and then it loads the SASHELP.HEART table into CAS memory. Next, PROC MDSUMMARY generates a basic statistical summary report from the CAS table. The PROC MDSUMMARY output is written into CAS server memory. Running PROC DATASETS verifies that the previous steps to load the data sets were successful.

This code example verifies that a SAS data set from the local workspace is loaded into CAS memory and that CAS procedures are running on the in-memory CAS table. The SAS log from the data load verifies the presence of the in-memory CAS table MYCAS.HEART.

Note: For clarity, the HEART data set was copied to the local WORK library.

```
4       proc upload data=heart
5          out=mycas.heart ;
6       run;


NOTE: Upload in progress from data=WORK.HEART to out=MYCAS.HEART
NOTE: Status window cannot be initialized.
NOTE: 875112 bytes were transferred at 30176230 bytes/second.
NOTE: The data set WORK.HEART has 5209 observations and 17 variables.
NOTE: Uploaded 5209 observations of 17 variables.
NOTE: The data set MYCAS.HEART has 5209 observations and 17 variables.
NOTE: PROCEDURE UPLOAD used (Total process time):
      real time           0.03 seconds
      cpu time            0.01 seconds
```

After the data is loaded into CAS, PROC MDSUMMARY generates simple statistical summaries for several variable fields in the table. The PROC MDSUMMARY results are returned to CAS memory.

```
7       proc mdsummary data=mycas.heart ;
8        groupby deathcause ;
9        var cholesterol systolic diastolic ;
10      output out=mycas.heartsum ;
11      run;

NOTE: The Cloud Analytic Services server processed the request in 0.003046 seconds.
NOTE: The data set MYCAS.HEARTSUM has 18 observations and 17 variables.
NOTE: PROCEDURE MDSUMMARY used (Total process time):
      real time           0.03 seconds
      cpu time            0.03 seconds
```

Looking at the CAS library, you should see two data sets: one that was loaded to CAS (HEART) and one that was created by PROC MDSUMMARY, (HEARTSUM).

```
12      proc datasets lib=mycas ;
                                                    Directory

                                 Libref                MYCAS
                                 Engine                CAS
                                 Physical Name         a4396f95-2e24-fb42-892c-229886ffc294
                                                         Directory

                                 Server Session UUID   a4396f95-2e24-fb42-892c-229886ffc294
                                 Server Session Name   CASAUTO
                                 Server Host           viyaserver01.race.sas.com
                                 Server Session Port   5570
                                 Server Session CASLIB CASUSER


                                         Number   Number
                                 Member     of       of                                Data
                        #  Name  Type     Rows  Columns  Last Modified                Encoding  Scope

                        1  HEART     DATA     5209       17  05/31/2016 15:53:43          utf-8     Session
                        2  HEARTSUM  DATA       18       17  05/31/2016 15:53:43          utf-8     Session
12 !                            run;
```

Viewing the Uploaded Data Set Using SAS Studio in SAS Viya

By default, you cannot use a SAS Studio session in SAS Viya to view the in-memory CAS table that you uploaded from the SAS 9.4 environment. This behavior is expected with in-memory CAS tables, and is related to the scope of the CAS session. The scope of the CAS session can be either global or session, with session defined as the default configuration.

Session-scope CAS library contents can be seen only by the session that added the CAS library. Global-scope CAS library contents can be seen and accessed by other sessions. To view the in-memory CAS table from the SAS Viya environment, add the PROMOTE= option to the PROC UPLOAD OUT= library specification that you

issued in the SAS 9.4 environment, and then refresh the CAS library in the SAS Viya environment. (A similar option is also available using PROC CASUTIL as well.)

The code for PROC UPLOAD (to enable viewing the in-memory CAS table with SAS Viya) would resemble the following:

```
PROC UPLOAD data=heart out=mycas.heart (promote=yes);       run;
```

Sample Code to Test SAS/CONNECT from SAS Viya to SAS 9.4

After confirming that SAS/CONNECT works across environments by submitting code from SAS 9.4 to the SAS Visual Data Mining and Machine Learning machine on SAS Viya, you can verify that the connection works in both directions by submitting code from SAS Viya to SAS 9.4.

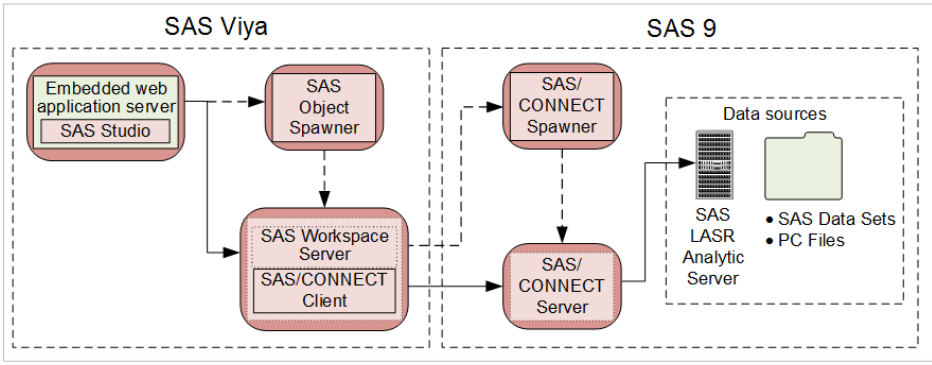
This example uses SAS Studio in a SAS Viya environment. When SAS Studio starts, it requests a SAS Viya workspace server using the object spawner. After the workspace server starts, it executes the test code and starts SAS/CONNECT client software.

The SAS/CONNECT client in the SAS Viya environment submits a SIGNON statement to the SAS/CONNECT spawner in the SAS 9.4 environment, requesting a connection to the SAS/CONNECT server. Once the SAS/CONNECT server is running, a connection is established between machines in both environments.

The SAS Viya workspace server also allocates a CAS library in CAS. The CAS library can then download data directly from the SAS/CONNECT server in the SAS 9.4 environment. Code for the task to be performed is sent using RSUBMIT statements in an RSUBMIT block.

The following diagram shows the servers involved and the initial flow of communication:

Testing SAS/CONNECT: SAS Viya to SAS 9



The sample code for this test is as follows:

```
/* Allocate CAS library named MYCAS as sasdemo    */

   %let caslibname = mycas;
   LIBNAME &caslibname cas caslib=casuser;

/* Connect to SAS 9.4 server using credentials and */
/* rsubmit code so you can download the HEART data */
/* set from the remote SASHELP                     */
/* Change <myHost.myDomain.com> and <port> below   */
/* to the host and port of the SAS/CONNECT spawner */

  %let myserver=<myHost.myDomain.com> <port>;
   SIGNON myserver user=sasdemo passwd="myPassword";
```

```
   rsubmit;


/* Download SASHELP.HEART dataset from SAS 9.4 host */
/* and save the table to memory in CAS library MYCAS */
/* Then use PROC CONTENTS to view the file in memory */

   PROC DOWNLOAD data=sashelp.heart out=mycas.heart94;
   run;
   endrsubmit;
   PROC CONTENTS data=mycas.heart94;

/* Create basic statistics using PROC MDSUMMARY and */
/* then save the summary output table to CAS memory */
/* as heartsum94.                                   */

   PROC MDSUMMARY data=mycas.heart94;
     GROUPBY deathcause;
     VAR cholesterol systolic diastolic;
     OUTPUT out=mycas.heartsum94;
   run;

/* Use PROC DATASETS to view the saved tables residing */
/* in memory in the CAS library MYCAS.                 */

   PROC DATASETS lib=mycas;    run;
   signoff ;
```

In this test you see that you can load data from the SAS 9.4 environment into CAS memory in the SAS Viya environment as a table, and then perform some basic statistics on the table using a CAS-based procedure, MDSUMMARY. Once the data is loaded into CAS, it is available for in-memory analytics.

Note: The CAS library is local to the SAS Viya workspace server, as well as the analytic code that is executed on the in-memory CAS tables in the Viya environment. Code that is submitted to the SAS/CONNECT server in the SAS 9.4 environment using RSUBMIT executes in the SAS 9.4 environment.

The output of PROC DATASETS for the CAS library MYCAS displays both the original table that was copied into CAS memory (HEART94) as well as the summary table that was generated in CAS memory using PROC MDSUMMARY.

Directory	
Libref	MYCAS
Engine	CAS
Physical Name	367a9a5c-f514-3a46-89a4-5f784c47c786
Server Session UUID	367a9a5c-f514-3a46-89a4-5f784c47c786
Server Session Name	CASAUTO
Server Host	pdcesx08060
Server Session Port	5570
Server Session CASLIB	CASUSER

#	Name	Member Type	Number of Rows	Number of Columns	Last Modified	Data Encoding	Scope
1	HEART94	DATA	5209	17	05/31/2016 16:10:42	utf-8	Session
2	HEARTSUM94	DATA	18	17	05/31/2016 16:10:42	utf-8	Session

These results indicate a successful use of SAS/CONNECT as a bridge between a SAS Viya environment and a SAS 9.4 environment.