



# How Can I Run My DATA Step Programs in SAS Viya

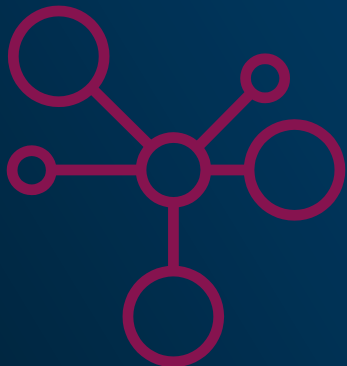
**T Winand**  
Senior Solutions Architect  
Customer Success Organization



## William “T” Winand Senior Presales Solutions Architect, SAS



T is a computer software professional and analyst with 24 years of experience in program design and development, data management, statistical analysis and reporting. A part of SAS since 1995, he helps organizations grow through analytics. He specializes in statistical analysis, descriptive and predictive data mining, text mining, forecasting and model management.



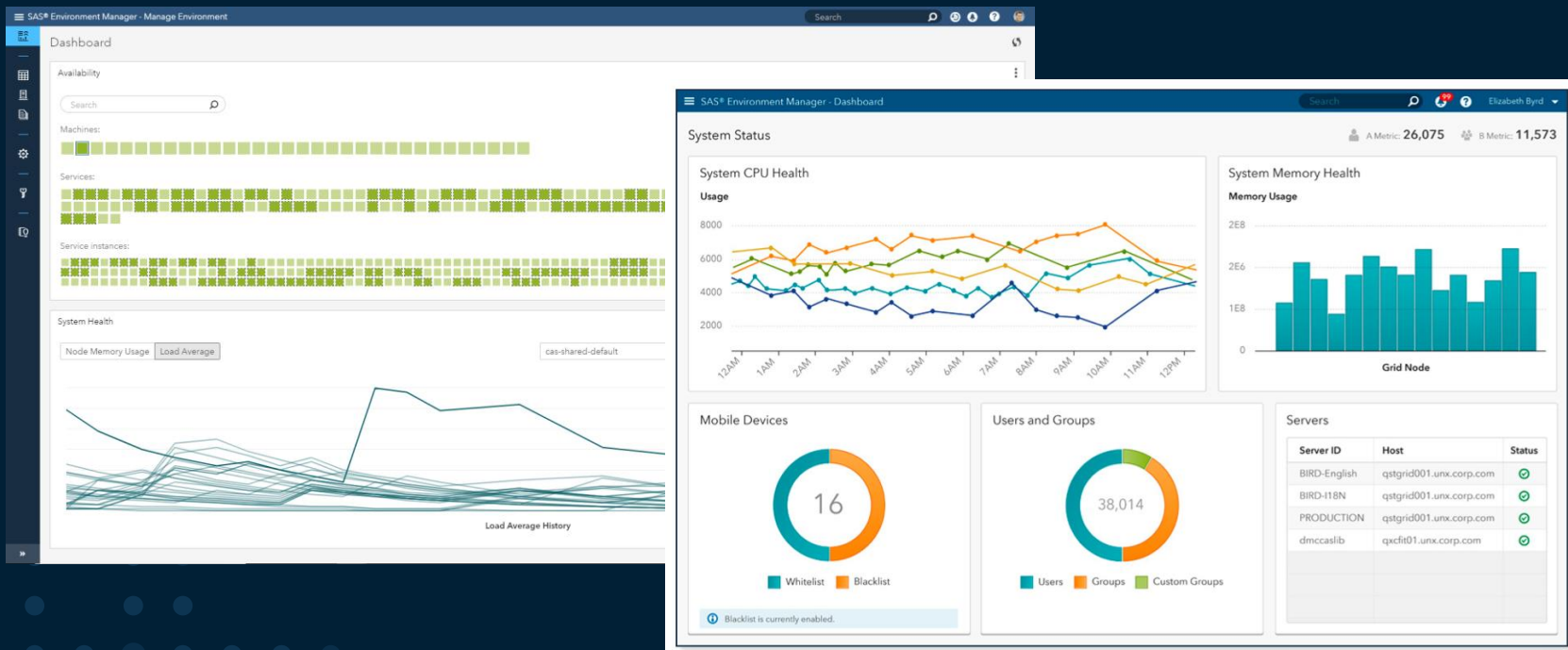
- Introduction to Viya
- What is CAS?
- SAS9 and Viya
- Programming in Viya
- Considerations
- Resources for Learning More

# Introduction to Viya

# What is SAS Viya?



SAS Viya is a cloud-enabled, in-memory analytics environment that provides quick, accurate and reliable analytical insights.





# What is CAS?



# Cloud Analytic Services (CAS) In-Memory Engine



The CAS in-memory engine is a fast, scalable, and resilient run-time environment for data management and analytics for SAS® Viya™



## Fast

- Multi-threaded
- Distributed In-Memory
- Efficient Inter-node Communication



## Scalable

- Single Machine to Distributed MPP
- Memory-mapping to process data larger than physical memory \* (No memory failures)



## Resilient

- Fault-tolerant to Node and Network Failures
- Worker and Controller\* Failover
- Session Independence

# Multiple interfaces, single code base

## Clients ask CAS to run “actions” on data

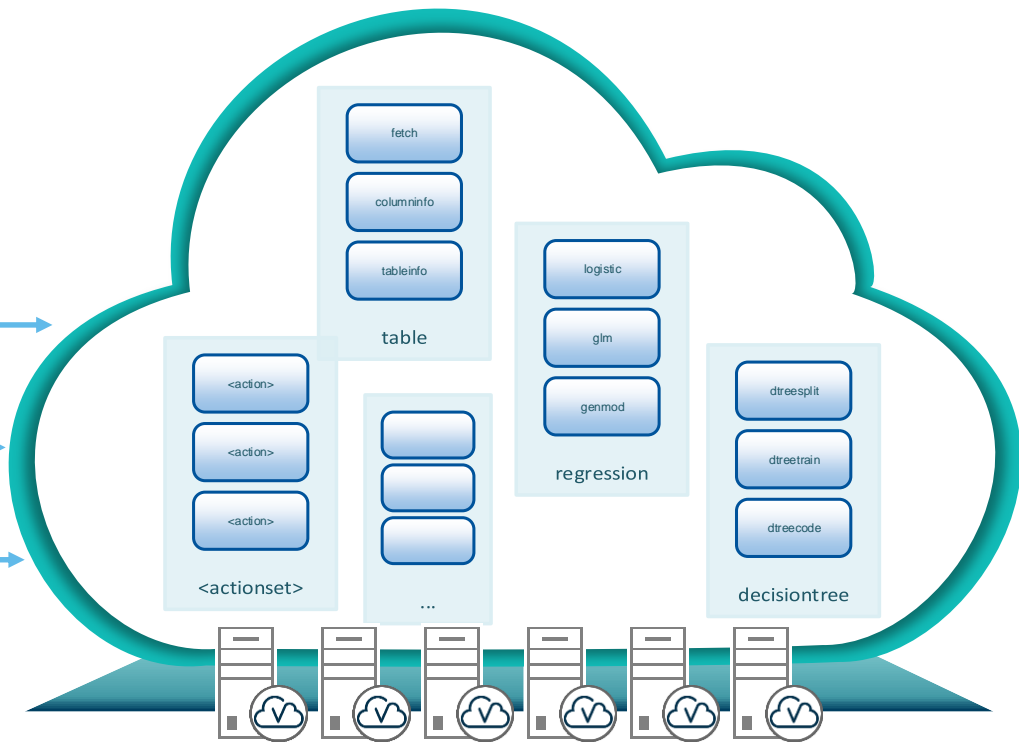
### Visual Interfaces



### Programming Interfaces



### API Interfaces

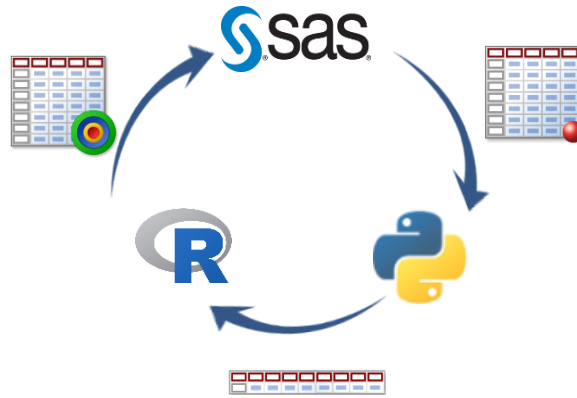


CAS Server



# Open Architecture Advantage

- Same CAS action regardless of the interface and API.
- Results will be equivalent (given that all random elements are seeded).
- Information can be passed between different languages.
  - Promote tables to persist on the server and apply CAS actions from different APIs.



# How APIs Work



```
proc logselect data=mycas.getStarted;  
class c1;  
model y = c1 x1 x2;  
selection method=forward details=all plots=all;  
run;
```



```
proc cas;  
regression.logistic /  
table={name='getStarted', caslib='mycas'},  
class={{vars={'c1'}}}, model={depVars={{name='y'}},  
effects={{vars={'c1'}},{vars={'x1'}},{vars={'x2'}}}},  
selection={method='forward', details='ALL'};  
run;quit;
```

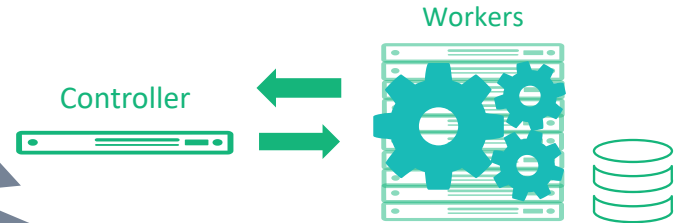


python

```
mdl = s.logistic(table='name':  
'getStarted', caslib:'mycas',  
classvars='c1',  
model='depvar':y, 'effects':['c1', 'x1', 'x2'],  
selection=('method':'forward','details':'all'))
```



```
cas.regression.logistic(s,  
table=list('name'='getStarted', 'caslib'='mycas'),  
classvars='c',  
model = list('depvar'='y', 'effects' =c('c1','x1','x2')),  
selection = list('method' = 'forward', 'details' = 'all'))
```



## Translated Action

```
action regression.logistic /  
table={name='getStarted', caslib='mycas'},  
class={{vars={'c1'}}},  
model={depVars={{name='y'}},  
effects={{vars={'c1'}},{vars={'x1'}},{vars={'x2'}}}},  
selection={method='forward', details='ALL'}
```

R  
E  
S  
T

```
curl -X POST http://.../cas/sessions/.../actions/regression.logistic \  
-u sasdemo:XXXXXXXX -H 'Content-Type: application/json' \  
-d '{"table":{"caslib":"demodata","name":"getStarted"},  
,"class":{"c1"},"model":{"depvar":"y","effects":["c1","x1","x2"]},  
,"selection":{"method":"forward","details":"all"}' "
```





# SAS 9.4 and SAS Viya

## The SAS Platform



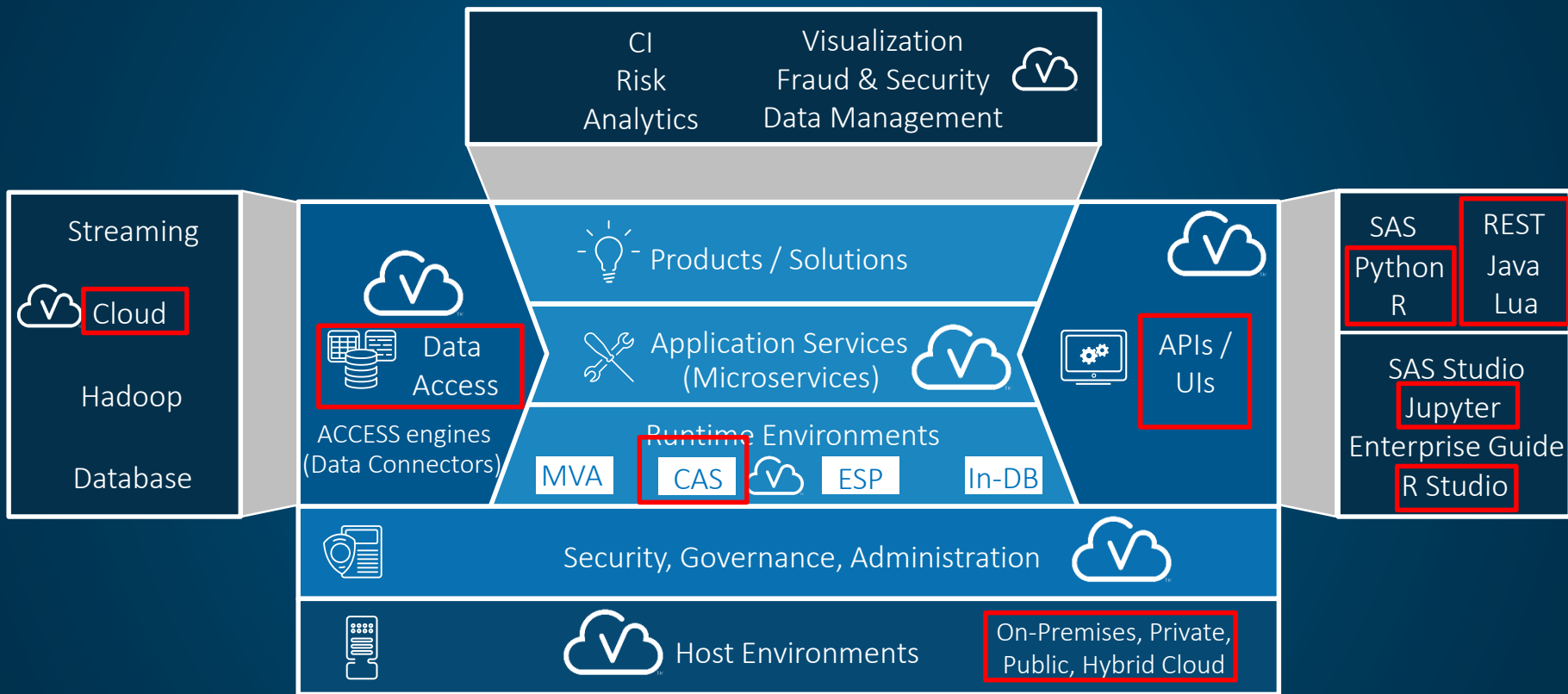
# SAS® 9.4 and SAS® Viya®

## A Single Platform

- SAS Platform
- Analytics Life Cycle
  - Data Management
  - Visualization
  - Analytics
  - Decision Management



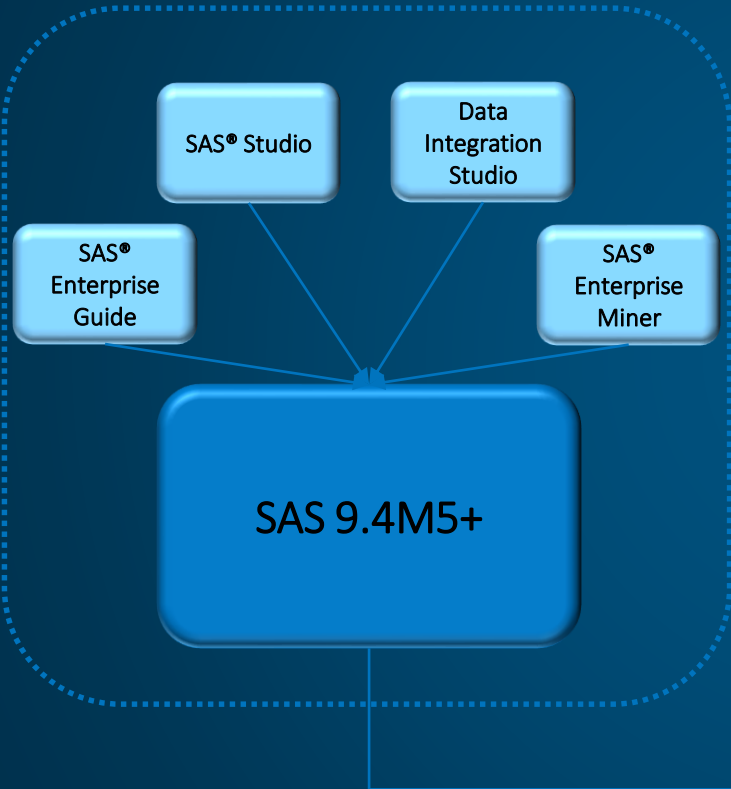
# The SAS Platform



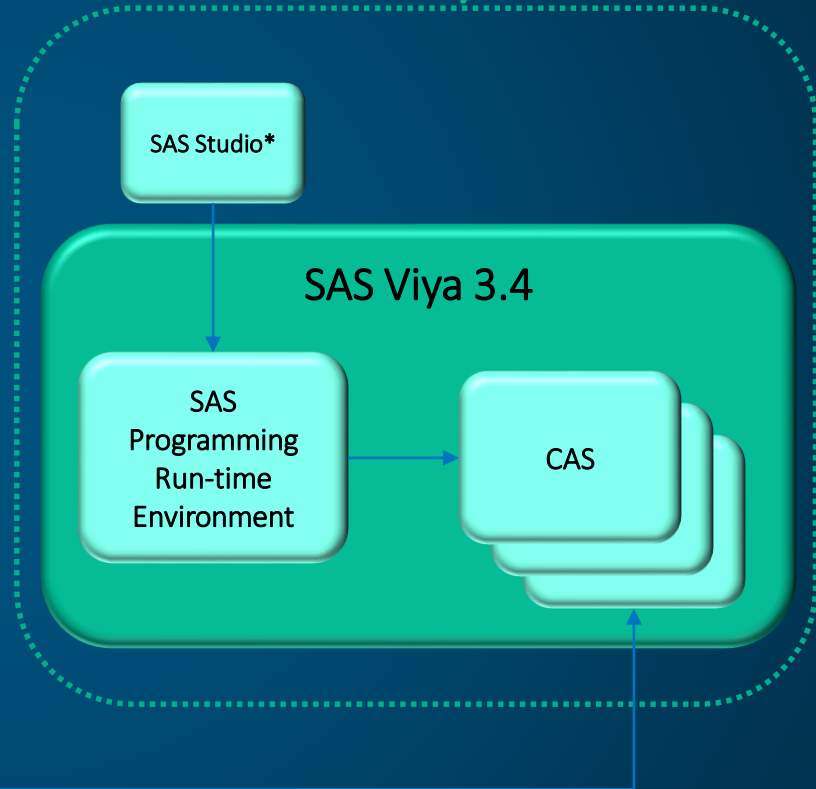
# The SAS Platform

## Language Execution

SAS 9.4



SAS Viya

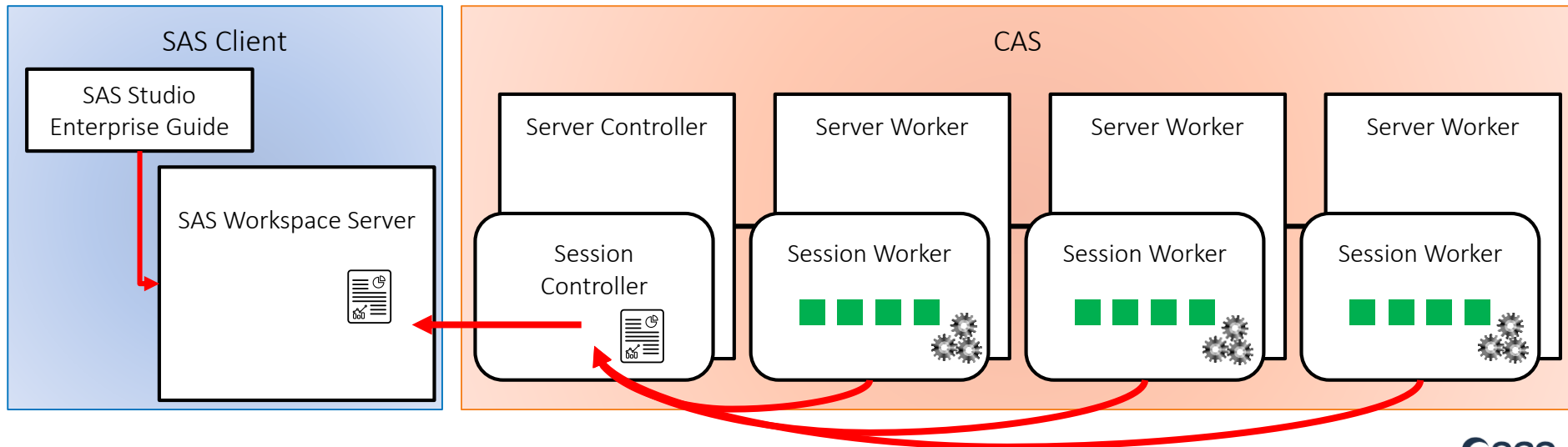


\*Many SAS Viya clients (e.g., SAS Visual Analytics) do not submit SAS code, but rather call CAS directly

# SAS Viya Data Processing

## CAS-enabled data processing

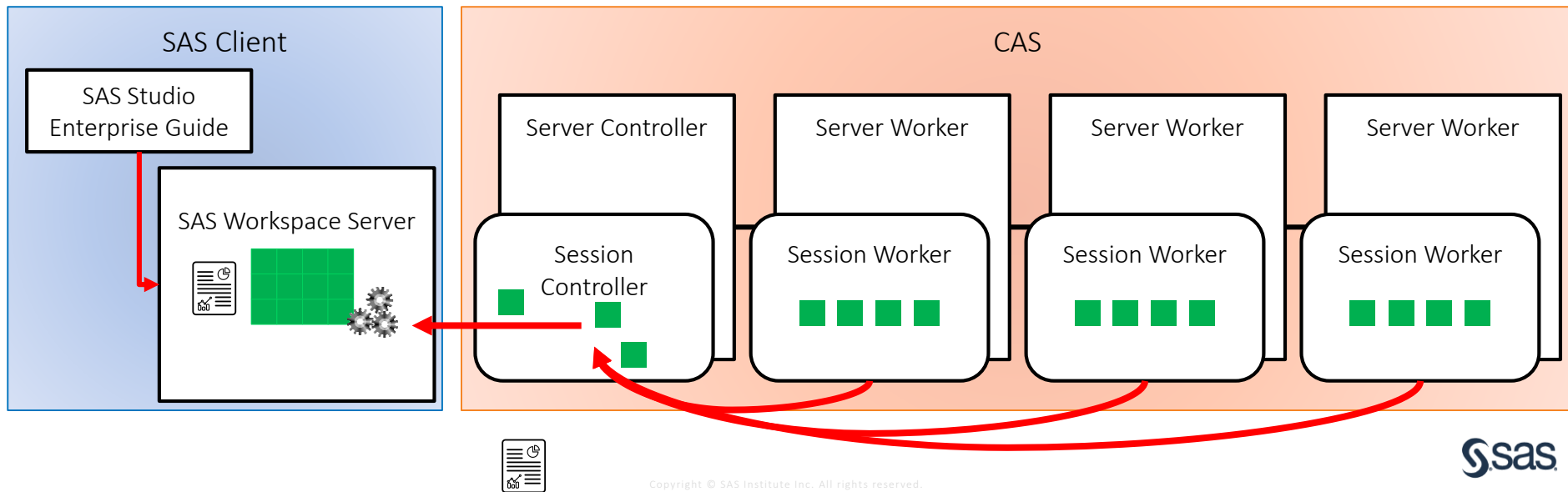
- Work is done by the CAS Session Workers
- Results passed to the CAS Session Controller for consolidation
- Passed back to the SAS 9 client



# SAS Viya Data Processing

## NOT CAS-enabled data processing

- Data is retrieved from CAS
- Passed to the SAS 9 Workspace Server or Compute Server
- **Work is done by the Workspace Server/Compute Server** single-threaded





# The SAS Platform

## Base SAS

- All 9.4 Base in Viya 3.3+
- SAS Viya = *speed!*
  - Multi-threaded DATA step
    - Rework code to leverage
    - Steven Sober's [SGF Paper #1710-2018](#)
      - SAS Viya Readiness Utility
- [Procedures That Use CAS Actions](#) (16)



# SAS Viya Data Processing

## General information

- **Most analytics run in memory**
  - Visual Statistics, Visual Forecasting, VDMML, Opt
- **SAS Foundation PROCs**
  - CAS-enabled
  - Not CAS-enabled
- **CAS Actions**
  - PROC CAS (CASL)
  - Python
  - Lua...

Append, Contents, Copy, Datasets, Delete, DS2, FCMP, FedSQL, Format, Lua, Means, Report, ScoreAccel, Summary, Tabulate, Transpose

Catalog, Compare, Download, DSTODS2, Export, FMTC2ITM, Hadoop, HDMD, HTTP, Import, JavaInfo, JSON, MapImport, Options, Print, PrintTo, Product\_Status, PWEncode, Registry, S3, SGPanel, SGPlot, SGRender, SGScatter, Sort, SQL, Stream, Template

[Procedures That Use CAS Actions](#)

[CAS Processing of Base SAS Procedures](#)

# Viya Programming

in four easy steps

# SAS Studio

SAS Studio 4.1  
Not secure | sasserwer.demo.sas.com/SASStudio/man?locale=en\_US&zone=GMT-04%253A00&http%3A%2F%2Fsasserwer.demo.sas.com%2FSASStudio%2F=

SAS® Studio

Program 1 x BasicSASExample.sas x Decision Tree x Supervised Learning 1 x

CODE LOG RESULTS OUTPUT DATA

- Server Files and Folders
- Tasks and Utilities
- Snippets
  - New CAS Session
  - Disconnect CAS Session
  - Reconnect CAS Session
  - Terminate CAS Session
  - List CAS Session Options
  - List CAS Sessions for SAS Client
  - List CAS Sessions for User ID
  - New caslib for Path
  - Generate SAS librefs for caslibs
  - Save table to caslib
  - Load data to caslib
  - Delete Table or File from caslib
  - Delete caslib
- SAS Viya Machine Learning
  - Load Data
  - Prepare and Explore Data
  - Compare Two ML Algorithms
  - Compare Several ML Algorithms
  - Generalized Linear Models
  - Unsupervised Learning
  - Supervised Learning
- SAS Viya Image Processing
- Libraries
- File Shortcuts

```
121 * Documentation [GRADBOOST Procedure](http://go.documentation.sas.com/?cdcId=vdmml)
122 */
123
124
125 proc gradboost data=mycas.&part_data. maxdepth=8 minleafsize=5 seed=9878 outmodel=
126 *autotune;
127 target &target. / level=nominal;
128 input &class_inputs. / level=nominal;
129 input &interval_inputs. / level=interval;
130 partition rolevar = _partind_(train='0' valid='1' test='2');
131 output out=mycas._scored_gradboost copyvars=( _partind_ &target.);
132 title "Gradient Boost";
133 run;
134
135 /*
136
137 ## Forest
138 * Documentation [FOREST Procedure](http://go.documentation.sas.com/?cdcId=vdmml)
139 */
140
141 proc forest data=mycas.&part_data. ntrees=50 minleafsize=5 outmodel=mycas.model_fo
142 target &target. / level=nominal;
143 input &class_inputs. / level=nominal;
144 input &interval_inputs. / level=interval;
145 partition rolevar = _partind_(train='0' valid='1' test='2');
146 output out=mycas._scored_forest copyvars=( _partind_ &target.);
147 title "Random Forest";
148 run;
149
150 /*
151
152 ## Neural Network
153
```

### The FOREST Procedure

Model Information	
Number of Trees	50
Number of Variables Per Split	4
Seed	2018937499
Bootstrap Percentage	60
Number of Bins	20
Number of Input Variables	12
Maximum Number of Tree Nodes	235
Minimum Number of Tree Nodes	121
Maximum Number of Branches	2
Minimum Number of Branches	2
Maximum Depth	20
Minimum Depth	20
Maximum Number of Leaves	118
Minimum Number of Leaves	61
Maximum Leaf Size	1325
Minimum Leaf Size	5
OOB Misclassification Rate	0.11661074

	Training	Validation	Test	Total
Number of Observations Read	3576	1788	596	5960
Number of Observations Used	3576	1788	596	5960

Variable Importance			
Variable	Importance	Std Dev	Relative Importance
IM_DELINQ	65.4795	9.5096	1.0000
VALUE	59.9028	17.4998	0.8980
DEROG	41.7050	6.6802	0.6369
IM_DEBTINC	36.4924	6.6232	0.5573
LOAN	32.1116	6.1436	0.4904



# PROC versus CAS Action

```
proc factmac data=mycas.movlens nfactors=10 learnstep=0.15
            maxiter=20 outmodel=mycas.factors;
  input userid itemid /level=nominal;
  target rating /level=interval;
  output out=mycas.out1 copyvars=(userid itemid rating);
run;
```

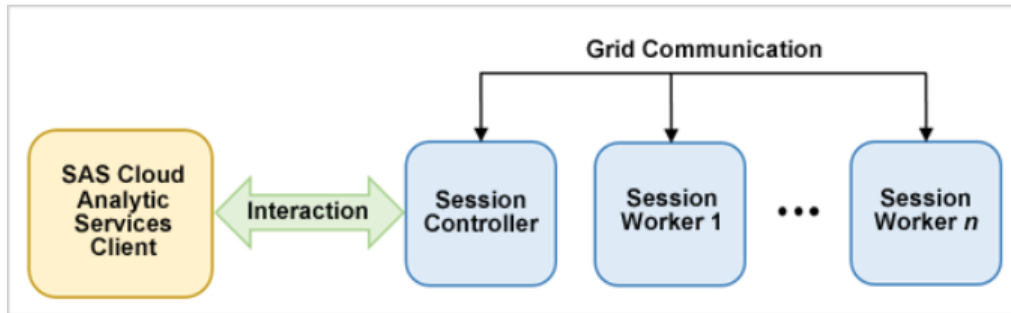
```
proc cas;
  action factmac result=R / table={name="movlens"},
  outModel={name="factors_out", replace=true},
  inputs={"userid", "itemid"},
  nominals={"userid", "itemid"},
  target="rating",
  maxIter=20, nFactors=10, learnStep=0.15,
  output={casout={name="score_out", replace="TRUE"},
  copyvars={"userid","itemid","rating"}};
run;
```

# Step 1: Establish a CAS Session

Purpose: Enables the client to communicate with the server

- User identification
- Fault isolation for each session
- Efficiency
- Resource tracking

*Session Processes in a Distributed System*



# Step 1: CAS Statement

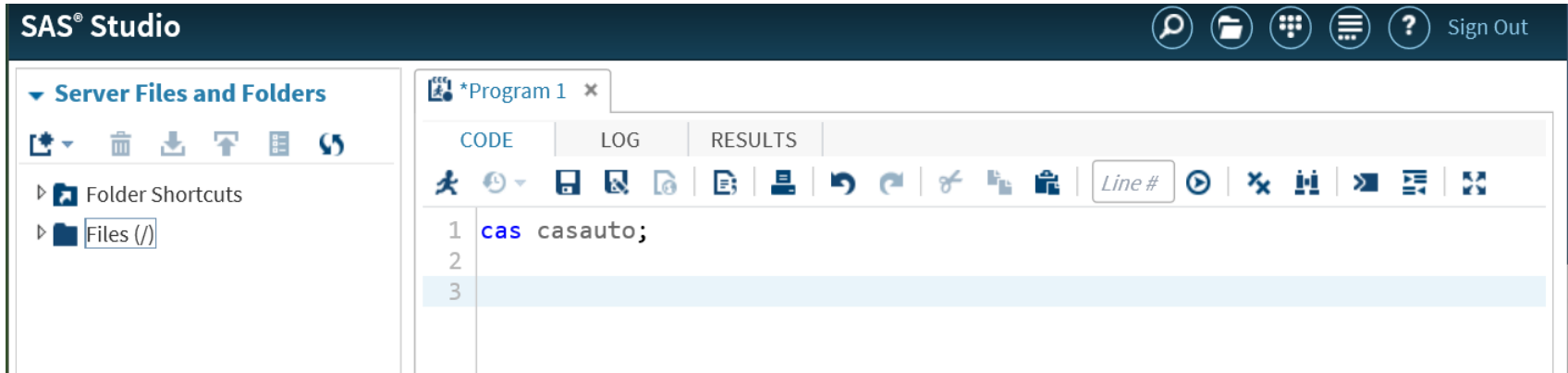
Purpose: Start and manage your CAS session

- Create a session
- Terminate a session
- List information about a specific CAS session or all your sessions
- List the properties of a session
- Manage format libraries in a session
- Change one or more session properties
- Disconnect a session
- Connect to an existing session

## Step 1 *continued*

### Syntax

CAS *session-name* <*option(s)*>;



The screenshot shows the SAS Studio interface. On the left is the 'Server Files and Folders' pane with a tree view showing 'Folder Shortcuts' and 'Files (/)'. The main area is a code editor titled '\*Program 1' with tabs for 'CODE', 'LOG', and 'RESULTS'. The code editor contains three lines of code: line 1 is 'cas casauto;', line 2 is blank, and line 3 is blank. The code editor has a toolbar with various icons and a 'Line #' input field.

```
73 cas casauto;
```

NOTE: The session CASAUTO connected successfully to Cloud Analytic Services viya47.dept-tbt.sashq-r.xxxxxxxxxx.sas.com using port 5570. The UUID is #####-f###-###c-a###-#####c. The user is sasxxx and the active caslib is CASUSER(sasdemo).

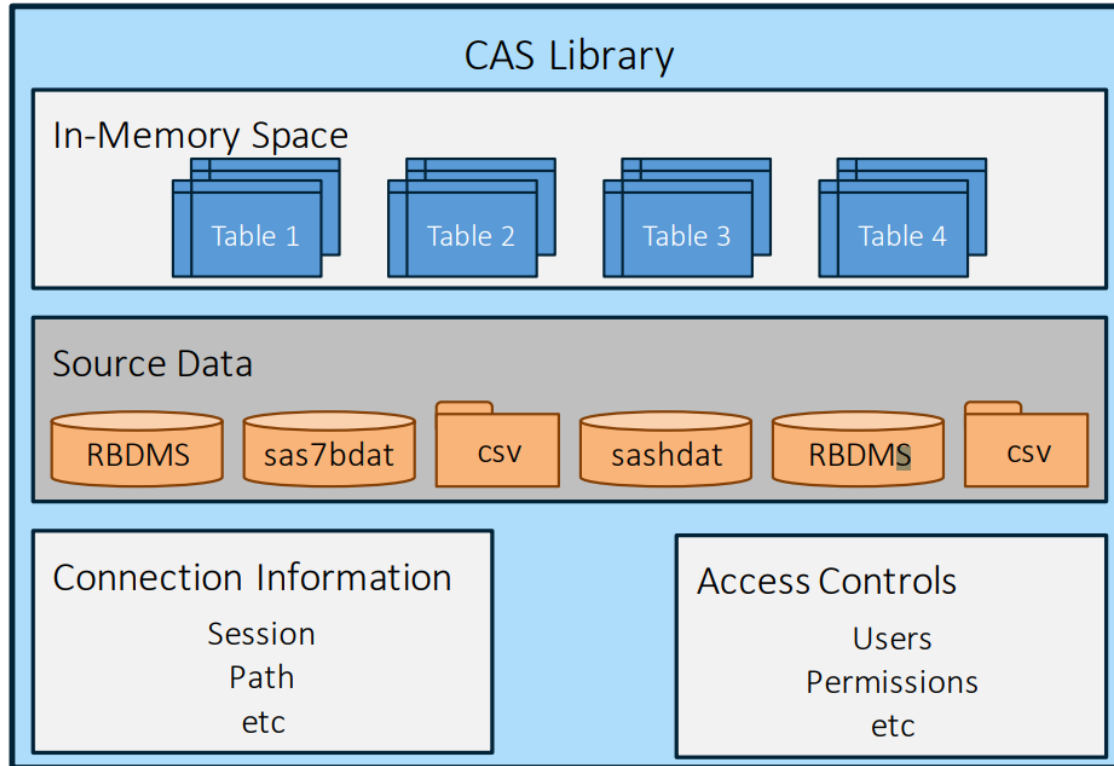
NOTE: The SAS option SESSREF was updated with the value CASAUTO.

NOTE: The SAS macro \_SESSREF\_ was updated with the value CASAUTO.

NOTE: The session is using 4 workers.



# What is a CASLIB?

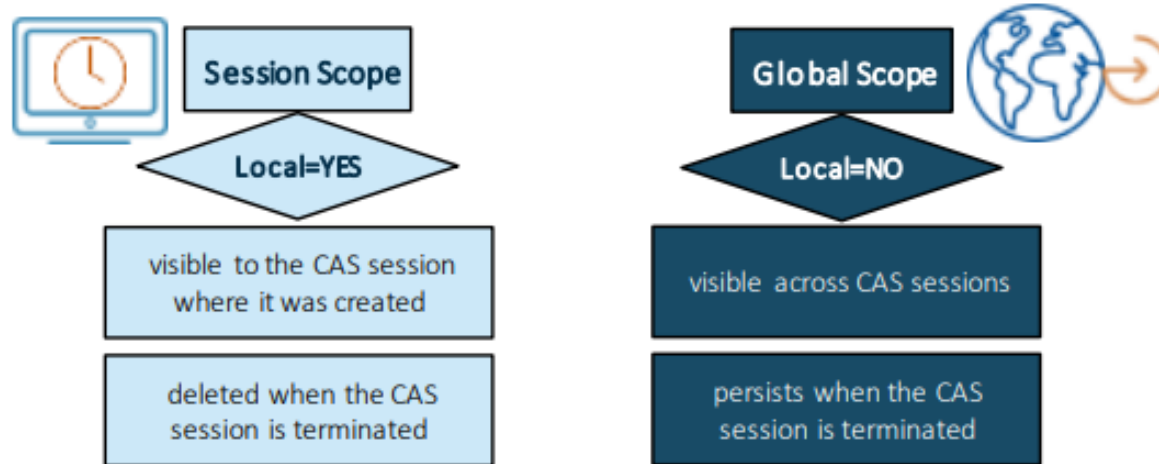


Logical CAS Library Representation



# CASLIB SCOPE

## Caslib Scope Session Versus Global



## Step 2: Read Data into CAS

Multiple file types, including the following:

- SAS data set
- Delimited text files
- Microsoft Excel files
- Hive
- Oracle
- Teradata

Multiple methods of reading in data, including the following:

- DATA step
- PROC CASUTIL
- PROC CAS

## Step 2: Read a SAS Data Set Using the DATA Step

Code:

```
libname mycas cas  
caslib='casuser';
```

```
data mycas.cars;  
  set sashelp.cars;  
run;
```

Log:

```
libname mycas cas caslib='casuser';  
NOTE: Libref MYCAS was successfully assigned as  
follows:  
Engine: CAS  
Physical Name: b864dbe5-75d9-9e4a-aa07-7d15ba1d33f2  
76  
77 data mycas.cars;  
78 set sashelp.cars;  
79 run;  
NOTE: There were 428 observations read from the data  
set SASHELP.CARS.  
NOTE: The data set MYCAS.CARS has 428 observations  
and 15 variables.  
NOTE: DATA statement used (Total process time):  
real time 0.00 seconds  
cpu time 0.00 seconds
```

## Step 2: Read in an Excel File Using PROC CASUTIL

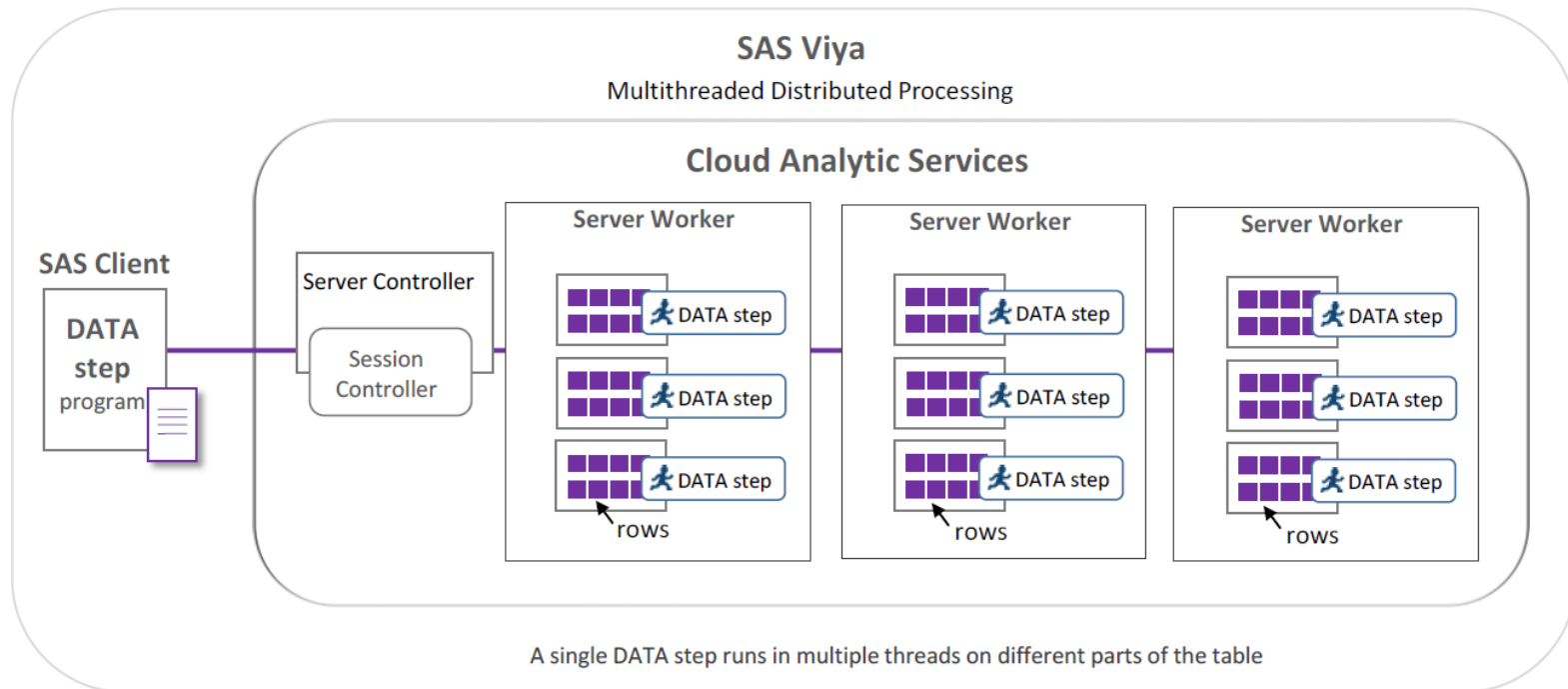
### Code:

```
proc casutil;  
load file='/tmp/cars.xls'  
casout='cars2'  
outcaslib='casuser'  
importoptions=(filetype='xls'  
getnames=true);  
quit;
```

### Log:

```
proc casutil;  
NOTE: The UUID 'b864dbe5-75d9-9e4a-aa07-  
7d15bald33f2' is connected using session CASAUTO.  
77 load file='/tmp/cars.xls'  
78 casout='cars2' outcaslib='casuser'  
79 importoptions=(filetype='xls' getnames=true);  
NOTE: Cloud Analytic Services made the uploaded file  
available as table CARS2 in caslib CASUSER(sasdemo).  
NOTE: The table CARS2 has been created in caslib  
CASUSER(sasdemo)  
from binary data uploaded to Cloud Analytic  
Services.  
80 quit;  
NOTE: PROCEDURE CASUTIL used (Total process time):  
real time 0.04 seconds  
cpu time 0.00 seconds
```

# Step 3: Manipulate the Data Using the DATA Step



## Step 3: Manipulate the Data Using the DATA Step *continued*

### Code:

```
data mycas.updated_transaction_history;
  set mycas.transaction_history;
  year=year(transaction_dt);
  fee=0;
  if transaction_status='SUCCESSFUL' then
do;
  if year=2013 then fee=1;
  else if year=2014 then fee=2;
  else if year>2014 then fee=3;
end;
new_transaction_amt=transaction_amt+fee;
new_act_balance=act_balance-
new_transaction_amt;
  put _threadid_ =;
run;
```

### Log:

NOTE: Running DATA step in Cloud Analytic Services.

NOTE: The DATA step will run in multiple threads.

\_THREADID\_=2

\_THREADID\_=4

\_THREADID\_=1

\_THREADID\_=3

NOTE: Duplicate messages output by DATA step:

\_THREADID\_=2 (occurred 720000 times)

\_THREADID\_=4 (occurred 719059 times)

\_THREADID\_=1 (occurred 721000 times)

\_THREADID\_=3 (occurred 720000 times)

NOTE: There were 2880059 observations read from the table TRANSACTION\_HISTORY in caslib CASUSER(sasdemo).

NOTE: The table updated\_transaction\_history in caslib CASUSER(sasdemo) has 2880059 observations and 8 variables.

NOTE: DATA statement used (Total process time):

real time 3.86 seconds

cpu time 0.00 seconds

## Step 3 *continued*

```
88 data updated_transaction_history;
89   set mylib.transaction_history4;
90   year=year(transaction_dt);
91   fee=0;
92   if transaction_status='SUCCESSFUL' then
do;
93     if year=2013 then fee=1;
94     else if year=2014 then fee=2;
95     else if year>2014 then fee=3;
96   end;
97   new_transaction_amt=transaction_amt+fee;
98   new_act_balance=act_balance-
new_transaction_amt;
99   run;
```

NOTE: There were 2880059 observations read from the data set MYLIB.TRANSACTION\_HISTORY4.

NOTE: The data set WORK.UPDATED\_TRANSACTION\_HISTORY has 2880059 observations and 9 variables.  
NOTE: DATA statement used (Total process time):

real time	47.93 seconds
cpu time	1.42 seconds



## Step 4: Analyze the Data Using Base SAS Procedures

PROC MEANS:

```
proc casutil;  
  load data=sashelp.cars outcaslib="CASUSER"  
  casout="cars" replace;  
quit;
```

```
proc means data=casuser.cars n mean max min;  
  var weight mpg_city;  
  by type;  
run;
```

## Step 4: Analyze the Data Using Viya Procedures

### PROC MDSUMMARY:

```
data mycas.updated_transaction_history2;  
  set mycas.updated_transaction_history;  
  month=put(transaction_dt,monname8.);  
run;
```

```
proc mdsurvey  
data=mycas.updated_transaction_history2 (where=(fee ne 0));  
  var fee;  
  groupby year month / out=mycas.summary_transaction_history;  
run;
```

```
proc print data=mycas.summary_transaction_history label;  
  title 'Summarized Transaction History Data';  
  var year month _min_ _max_ _nobs_ _sum_;  
  label _sum_='Total collected ($)' _nobs_='Number of Fees  
  Collected';  
  format _sum_ dollar8.;  
run;
```

## Step 4: Analyze the Data Using PROC CAS

PROC CAS using the Simple Action Set:

```
proc cas;  
  session casauto;  
    simple.freq /  
      inputs={'transaction_status'}  
      table={caslib='casuser', name='updated_transaction_history2',  
            groupby={name='year'}}};  
quit;
```



# Demonstration



# Keep In Mind...

- Your SAS 9 code will work in Viya
- Code & Procedures that cannot effectively take advantage of CAS, run in SPRE
- SAS is smart (it will run process in CAS when it can – no user intervention required)
- If your tables are loaded into CAS, much of your data step code will run in CAS
- Use PROC CASUTIL to load your data into CAS (versus DATA Step)
- BUT REMEMBER... You don't need to convert everything to run in CAS
  - and you may not want to (e.g. By Variables with High Cardinality)
  - think architecture, strengths, and tradeoffs
  - Go for the big wins – long running steps, large tables, analytical procedures
- If use SAS Enterprise Guide in SAS 9, you can still use it in Viya
- CAS does not care about order
- PROC SQL runs in SPRE; PROC FEDSQL runs in CAS (if associated with a CAS Session)

# Resources

# Documentation

The screenshot shows a web browser window displaying the SAS documentation page for SAS Viya 3.4. The browser's address bar shows the URL: [https://go.documentation.sas.com/?cdcid=pgmsascdc&cdcVersion=9.4\\_3.4&docsetId=casdsprgm&docsetTarget=titlepage.htm&locale=en](https://go.documentation.sas.com/?cdcid=pgmsascdc&cdcVersion=9.4_3.4&docsetId=casdsprgm&docsetTarget=titlepage.htm&locale=en). The page header includes the SAS logo and navigation links: Home, Support, Learn, and Connect. The main content area is titled "SAS® 9.4 and SAS® Viya® 3.4 Programming Documentation / DATA Step Programming for CAS". A left-hand navigation menu lists various topics, with "DATA Step Programming for CAS" selected and expanded. The main content area displays the title "SAS® Cloud Analytic Services 3.4: DATA Step Programming" and includes links for "DATA Step Basics" and "DATA Step Examples". At the bottom of the main content area, there is a copyright notice: "Copyright © SAS Institute Inc. All Rights Reserved. Last updated: November 5, 2018".

SAS® Help Center: SAS® Cloud x +

https://go.documentation.sas.com/?cdcid=pgmsascdc&cdcVersion=9.4\_3.4&docsetId=casdsprgm&docsetTarget=titlepage.htm&locale=en

Apps 1-SAS HOM PER WCC SWW | SAS Wide W... MIDAS Tickets RACE EXNET Home SAS Video Portal Arlo Tip: How to execute... Sign in to Xfinity Inbox (28) - i

**sas** THE POWER TO KNOW. Home Support Learn Connect

SAS® 9.4 and SAS® Viya® 3.4 Programming Documentation / DATA Step Programming for CAS

Version ▾

**SAS® Cloud Analytic Services 3.4: DATA Step Programming**

[DATA Step Basics](#)  
[DATA Step Examples](#)

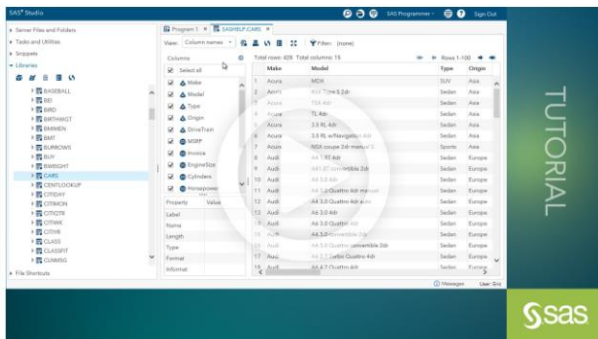
Copyright © SAS Institute Inc. All Rights Reserved.  
Last updated: November 5, 2018

- ▶ SAS Interface to Application Response Measurement (ARM)
- ▶ SAS Servers
- ▶ SAS Data Quality
- ▶ Learning SAS Programming
- ▼ SAS Viya Programming
  - Quick Start for SAS Viya
    - ▶ Introduction to SAS Viya Programming
    - ▶ SAS Viya Analytics Procedures
    - ▶ CAS Action Programming with CASL, Lua, Python, and R
  - Client Authentication Using an Authinfo File
  - ▶ Data Migration
  - ▶ CAS User's Guide
  - ▼ DATA Step Programming for CAS

## [Viya Programming Documentation](#)

# Video Tutorials

[Industry Solutions](#) [Products](#) [Learn](#) [Support](#) [Customer Stories](#) [Partners](#) [About SAS](#)



## Getting Started with SAS Studio

In this video, you get started with programming in SAS Studio. You view a data table, write and submit SAS code, view the log and results, and use interactive features to quickly generate graphs and statistical analyses.

[Learn about SAS Training - Programming path >](#)



## CATEGORIES

Trending

[Products & Solutions](#)

[Analytics in Action](#)

[How To Tutorials](#)

[SAS Add-In for Microsoft Office](#)

[Administration & Architecture](#)

[Advanced Analytics](#)

[SAS Analytics U](#)

[Customer Intelligence](#)

An Introduction to SAS Viya Programming for SAS 9 Programmers

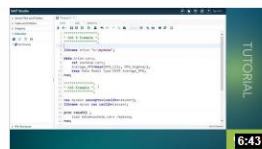
Search Videos

1-13 of 13

Sort By: Most Viewed



SAS® Viya™ Programming for SAS® 9 Programmers: Overview



Understanding Caslibs and Loading Data in SAS® Viya™



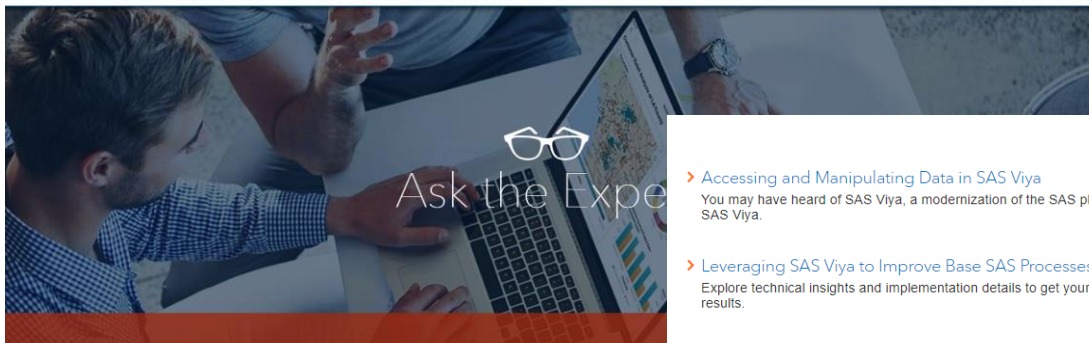
Using the DATA Step in SAS® Viya™

## Viya Programming for SAS 9 Programmers

Copyright © SAS Institute Inc. All rights reserved.







## Free Webinars

## Viya

### > [Accessing and Manipulating Data in SAS Viya](#)

You may have heard of SAS Viya, a modernization of the SAS platform, but you might not be familiar with how to access and manage your data in SAS Viya.

### > [Leveraging SAS Viya to Improve Base SAS Processes](#)

Explore technical insights and implementation details to get your code CAS-enabled, achieving faster results.

### > [SAS Grid Manager and SAS Viya: A Perfect Pair](#)

Join us for this webinar and learn how SAS Grid Manager works with SAS Viya to process large volumes of data for rapid results.

### > [SAS Optimization on SAS Viya: Getting Started](#)

In this presentation, you'll get coding examples for linear programming, mixed integer linear programming, nonlinear programming and network optimization.

### > [SAS Visual Data Mining and Machine Learning: Getting Started](#)

The presentation and accompanying demonstration will offer a view into SAS Visual Data Mining and Machine Learning.

### > [SAS Visual Statistics on Viya: Getting Started](#)

Quickly create predictive models through the visual point-and-click interface or through programming using SAS Studio or other programming interfaces.

### > [SAS Viya Architecture and Data Movement: A SAS Programmer's Overview](#)

Experienced SAS programmers can learn more about SAS Viya architecture and SAS Viya programming.

### > [SAS Viya: Working With APIs](#)

Experienced and novice SAS users can join us to learn how and when to use APIs to run machine learning models in SAS Viya.

## [Ask The Expert Webinars](#)

# Training

The screenshot shows the SAS Learning Center website. The browser address bar displays the URL: <https://support.sas.com/edu/schedules.html?ctry=us&crs=PGVIYA#s1=3>. The SAS logo is prominently displayed at the top left, with the tagline 'THE POWER TO KNOW.' Below the logo are navigation links for Home, Support, Learn, and Connect. A 'LEARN / TRAINING' header is present, followed by a 'Training Console' section with a search bar and a 'My Training' link. On the left side, there is a 'LEARN' sidebar with various categories like Books, Training, My Training, Find a Course, e-Learning, Live Web Classes, Locations, Training Formats, Discounts, Free Tutorials, Ask the Expert, Academy for Data Science, Learning Subscription, and Certification. The main content area is titled 'Programming for SAS® Viya®' and has three tabs: Overview, Prerequisites, and Course Outline. The 'Overview' tab is active, showing an 'Introduction' section with a bulleted list of topics: overview of SAS Viya, accessing SAS libraries, session-scope and global-scope caslibs, accessing caslibs, changing the active caslib, and connecting to a CAS server using a snippet. Below this is a section for 'Loading Data into SAS Cloud Analytic Services (CAS)' with a bulleted list: overview of loading tables into CAS and saving the tables, session-scope and global-scope tables, accessing SAS data sets, CSV files, and Excel files and loading them into CAS, promoting tables in CAS, and saving and using SASHDAT files. The final section is 'Modifying Base SAS Programs to Run in SAS Viya' with a bulleted list: overview of single-threaded processing in Base SAS, overview of multi-threaded processing in SAS Viya, modifying Base SAS DATA step programs to run in CAS, understanding DATA step restrictions in CAS, and modifying SQL procedure code to execute in CAS using FedSQL.

To access free e-learning courses:

[Free e-Learning for Employees](#)

[Programming for SAS Viya](#)

# Useful Websites

## Developer.sas.com, Communities.sas.com



SAS for Developers [Give us your feedback](#)

Use the power of SAS Viya™ in your applications.

SAS Viya is an open...  
One underlying...

**sas**  
SAS Viya uses PROC CAS to run CAS actions in SAS Cloud Analytic Services.

**Java**  
Java APIs for using SAS Viya CAS actions.

Home - SAS Support Commu x  
https://communities.sas.com

**sas** THE POWER TO KNOW.

Products & Solutions Industries Support Learn SAS Partners Community About SAS

**Communities** Welcome to SAS Support Communities Register · Sign In · Help

Stuck on a problem? Ask the community for help.

While you're here, get a SAS tip and share what you know. This community of SAS experts is here to help you to succeed!

[Join Now >](#)

Find a Community

Latest Activity

Featured Posts Popular Posts Unanswered Questions

**MEMBERS** 115,545 **ONLINE** 1,576 **POSTS** 309,408

**MDX for authorization in a cube** 0 6 0  
In: Administration and Deployment | Latest post by: Cruptic New Replies Likes

[Post a Question](#)

# https://github.com/sassoftware



## SAS Software

Open Source from SAS Software

Cary, North Carolina, USA

<https://www.sas.com/>

[github@sas.com](mailto:github@sas.com)

Repositories 81

People 15

### Pinned repositories

#### [sas\\_kernel](#)

A Jupyter kernel for SAS. This opens up all the data manipulation and analytics capabilities of your SAS system within a notebook interface. Use the Jupyter Notebook interface to execute SAS code a...

Jupyter Notebook ★ 80 🍴 33

#### [dm-flow](#)

Library of SAS Enterprise Miner process flow diagrams to help you learn by example about specific data mining topics.

★ 32 🍴 28

#### [sas-viya-programming](#)

Code samples and materials to help you learn to access SAS Viya services by writing programs in Python and other open-source languages

Jupyter Notebook ★ 40 🍴 36

#### [sas-prog-for-r-users](#)

Teaching and lab materials for the "SAS Programming for R Users" course, including course notes, data, and code.

SAS ★ 31 🍴 21

#### [saspy](#)

A Python interface module to the SAS System. It works with Linux, Windows, and mainframe SAS. It supports the `sas_kernel` project (a Jupyter Notebook kernel for SAS) or can be used on its own.

Jupyter Notebook ★ 71 🍴 39

#### [python-swat](#)

The SAS Scripting Wrapper for Analytics Transfer (SWAT) package is the Python client to SAS Cloud Analytic Services (CAS). It allows users to execute CAS actions and process the results all from Py...

Python ★ 29 🍴 18

# Resources

## Programming

- [SAS Studio](#)
- [CAS actions documentation](#)
- [SAS Github page for SWAT-Python](#)
- [SAS Github page for SWAT-R](#)
- [More example scripts for using SWAT-R & SWAT-Python](#)

# SAS<sup>®</sup> Viya Resources

## SAS Visual Statistics User's Guide

<http://support.sas.com/software/products/visual-statistics/index.html#s1=2>

## SAS Visual Data Mining and Machine Learning User's Guide

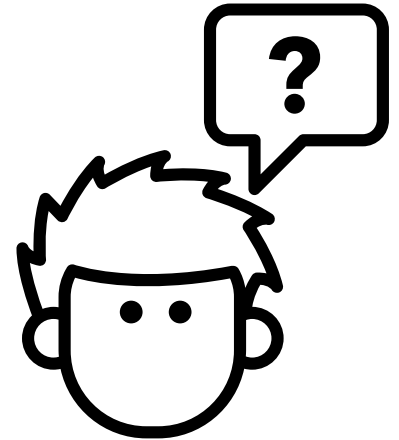
<http://support.sas.com/software/products/visual-data-mining-machine-learning/index.html#s1=1>

## SAS Visual Text Analytics User's Guide

<http://support.sas.com/software/products/visual-text-analytics/index.html>

## Overview, Training, Samples and Tips

- [SAS Viya Overview](#)
- [SAS Viya Training](#)
- [A Beginner's Guide to Programming in the SAS<sup>®</sup> Cloud Analytics Services Environment](#)



# Q&A

# Explore Helpful Resources

## [Ask the Expert](#)

View other user webinars that provide insights into using SAS products to make your job easier.

## [FREE Training](#)

Learn from home – free for 30 days. Get software labs to practice and online support if needed.

## [SAS Support Communities](#)

Ask questions, get answers and share insights with SAS users.

## [SAS Analytics Explorers](#)

An exclusive platform to collaborate, learn and share your expertise. Gain access to a diverse network to advance your career. Special rewards and recognition exclusively for SAS users.

## [SAS Users YouTube Channel](#)

A plethora of videos on hundreds of topics, just for SAS users.

## [Newsletters](#)

Get the latest SAS news plus tips, tricks and more.

## [Users Groups](#)

Meet local SAS users, network and exchange ideas – virtually.

## [SAS Profile](#)

If you haven't already done so, create your SAS Profile to access free training, SAS Support Communities, technical support, software downloads, newsletters and more.