ASK THE EXPERT

How Do I Use Python in SAS® Viya®?

Peter Styliadis
Technical Training Consultant, SAS
Peter Styliadis
Technical Training Consultant, SAS

Peter delivers data analytics and data visualization customer trainings and workshops online and in person, working with subject-matter experts to research and design courses. He teaches SAS customers how to process data in all stages of the analytics life cycle using a variety of programming languages such as SAS, SQL, CASL and Python, and tools like SAS Visual Analytics and SAS Data Studio. He also instructs customers on how to process big data using the SAS Viya platform.
Workshop Overview

SAS Viya Overview

Python and SAS Viya

Demonstration
Workshop Overview

SAS Viya Overview

Python and SAS Viya

Demonstration
SAS Viya Overview

Analytics Life Cycle
SAS® Viya®

SAS Compute Server
Cloud Analytic Services (CAS)

Browser

Access
Explore
Prepare
Visualize
Analyze

Cloud-native
Scalable
In-Memory
All Users

SAS Compute Server

Cloud Analytic Services (CAS)
Parallel Processing
Cloud Analytic Services (CAS) in SAS Viya

Cloud-native, high-performance in-memory analytics and distributed computing engine
Data persists in memory. Data is automatically distributed across the worker nodes. Data Source.
Multiple users *can* process the same table.

Data Processes in *Parallel*.

Returns *summarized* results.

CAS *maximizes* parallel processing and *minimizes* disk I/O.
CAS Server Benefits

- Persists in memory, reducing I/O
- Processes data in parallel
- Avoids data duplication
- Utilize a variety of SAS Viya applications
Cloud Analytic Services (CAS) in SAS Viya

The CAS server can access a variety of data sources:
- Database
- Hadoop
- Streaming
- Path
- Cloud
Data in CAS is accessed through caslibs

Caslibs provide common interface into accessing different data sources
Caslib

In-Memory Tables

Data Source (Files)

Tables available for processing

Connection information to a data source

Access controls
Cloud Analytic Services (CAS) in SAS Viya

**Languages**
- SAS
- Python
- REST API
- FedSQL
- R
- Lua

**Applications**
- SAS Visual Analytics
- SAS VDMML
- SAS Data Preparation
- SAS Visual Forecasting
- SAS Optimization
- SAS Visual Investigator
- SAS Visual Text Analytics
- SAS Customer Intelligence 360

**CAS Actions**
- Manage Data
- Process Data
- Analyze Data
- Model Data
- DATA Step
- FedSQL
- DS2

**CAS Server**
- Controller
- Worker

**Data Sources**
- Database
- Hadoop
- Streaming
- Path
- Cloud

*Optimized* units of work for the *distributed* CAS server.
CAS Actions Overview

CAS Actions return a dictionary
<action-set-name>.action-name
table.fetch(table={'name': 'cars', 'caslib': 'casuer'}, to=10, index=False)
Python is *case-sensitive*, SAS is not.

table.fetch(TABLE={'NAME':'CARS',
                 'CASLIB':'CASUSER'},
            TO=10,
            INDEX=False)
CAS Action Benefits

- Optimized for the CAS server
- The same action is used
- Equivalent results
- Actions are multi-purpose
Execute commands in CAS
CAS Considerations

CAS Server
Is CAS always faster?

- Starting services.
- Explicitly loading data and code to worker nodes.
- Monitoring data processing on worker nodes.
- Collating worker node results on the controller.
- Delivering results to the client.
Which programs should you **consider** running in CAS?

- Data source *larger* than your client machine can handle
- *Long-running* code
- Computationally *demanding* processes
- Preparing CAS tables for use in another *SAS Viya application*
- Accessing a variety of data sources
Client and CAS Scenarios

1. Processing data in the CAS server
2. Copying a CAS table to your client for processing
3. Uploading data to CAS for use in other applications
Processing data in the CAS server

Analyze the smaller summarized results on the client.

Explore, prepare, analyze, and model big data in CAS.
Copying a CAS table to your client for processing

Process the copy of the table on your client

Copy a small table from the CAS server
Uploading data to CAS for use in other applications

Send data to the CAS server from your client
Use the best tool for the job

Python + CAS Server
Introduction to the SWAT Package

Native python packages do not execute in CAS
Many familiar methods

SWAT Package

Many familiar methods

CAS actions and CAS methods

CAS API

CAS Server
Many familiar methods

Python

casTbl.head()

casTbl.fetch / ...;
CAS actions

Python

casTbl.fetch(...)
Many familiar methods

The objects returned to the client differ.
SWAT Package

Many familiar methods

- CASTable
- CASColumn
- SASDataFrame
- Pandas.DataFrame
- Pandas.Series

CAS actions

- CASResults object
CASTable and CASColumn are client-side references to a CAS table. SASDataframe is a subclass to a Pandas DataFrame. CASResults is like a Python dictionary.
Getting Started with CAS
import swat

conn = swat.CAS(host, port, authentication)

# Execute CAS actions and SWAT methods
conn.terminate()
Workshop Overview

- SAS Viya Overview
- Python and SAS Viya
- Demonstration
Questions?

Please submit your questions using the Q&A icon located in the menu at the bottom of your screen.
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.
Thank you for joining us for this SAS webinar.