

Back to the Studio

Consultancy | Managed Services | Graduate Placement
SAS Software | SAS Training





In This Presentation

- Available programming clients
- SAS Studio
 - What is it?
 - Why is SAS Studio Important?
 - Understanding Versions and Editions
- Common features from existing client applications
- Extensibility
 - With a real-world example
- Conclusions

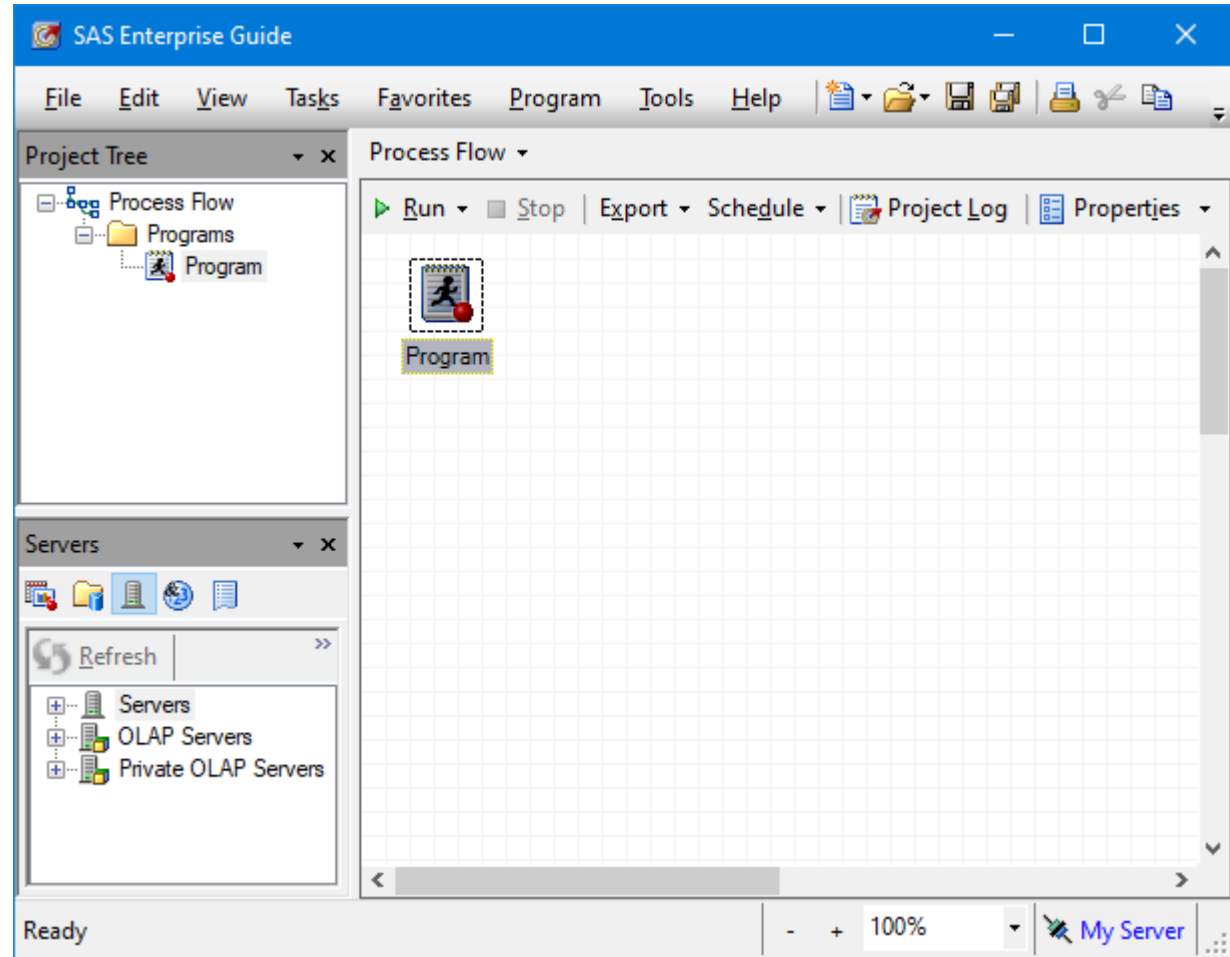


Available Programming Clients

The newer client application is called Enterprise Guide.

This is a “thin client”. Thus, provides a programming application, without the processing engine. Instead, a server connection is used to connect to a remote SAS server.

This is a more generally accepted method of software architecture, providing ease of deployment, administration and licencing cost efficiencies.



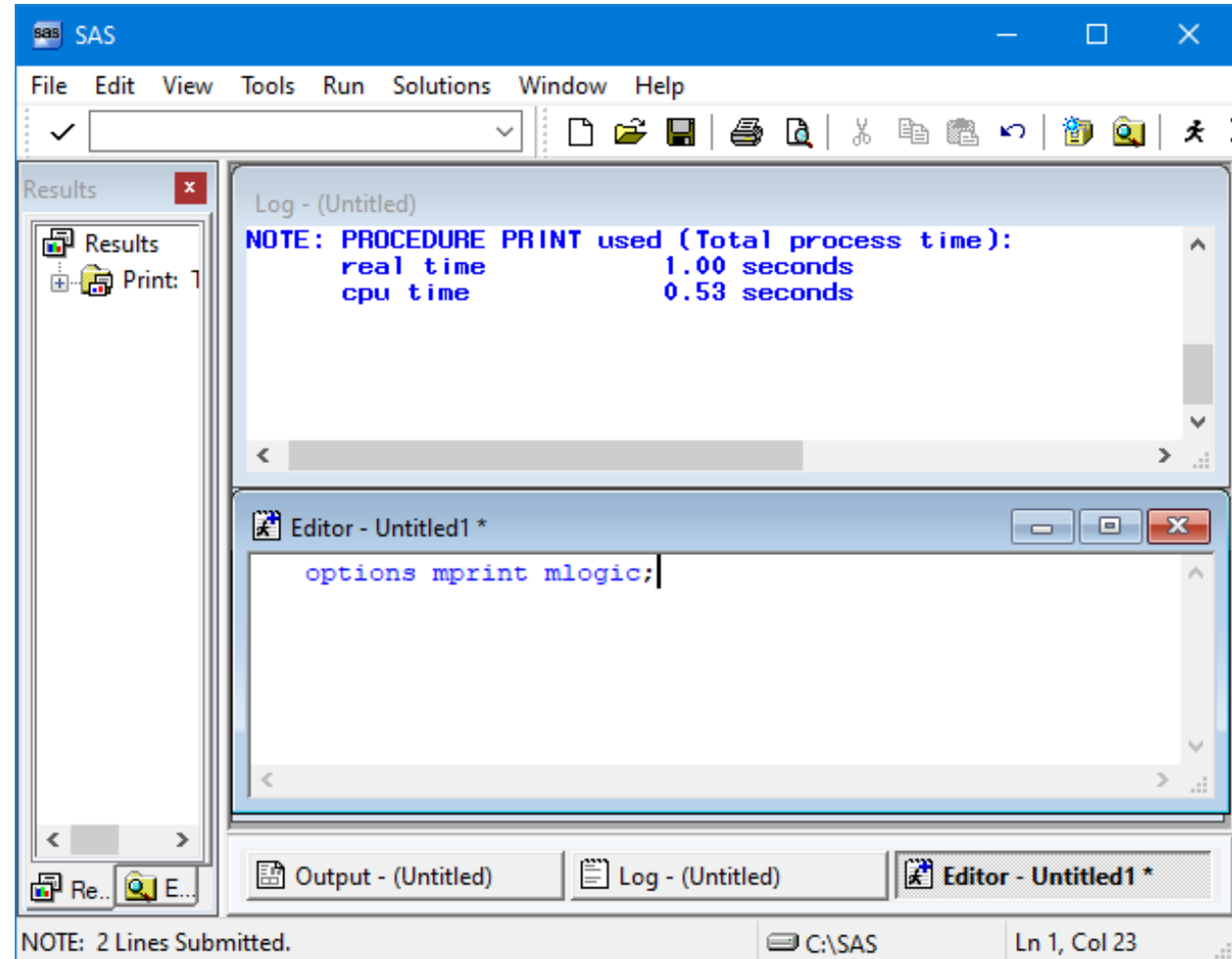


Available Programming Clients

For those who recognise this Window, you may refer to this application simply as "SAS".

This is the full client. i.e.: the default application when SAS is installed directly on your computer.

The correct name for this application is SAS Display Manager.



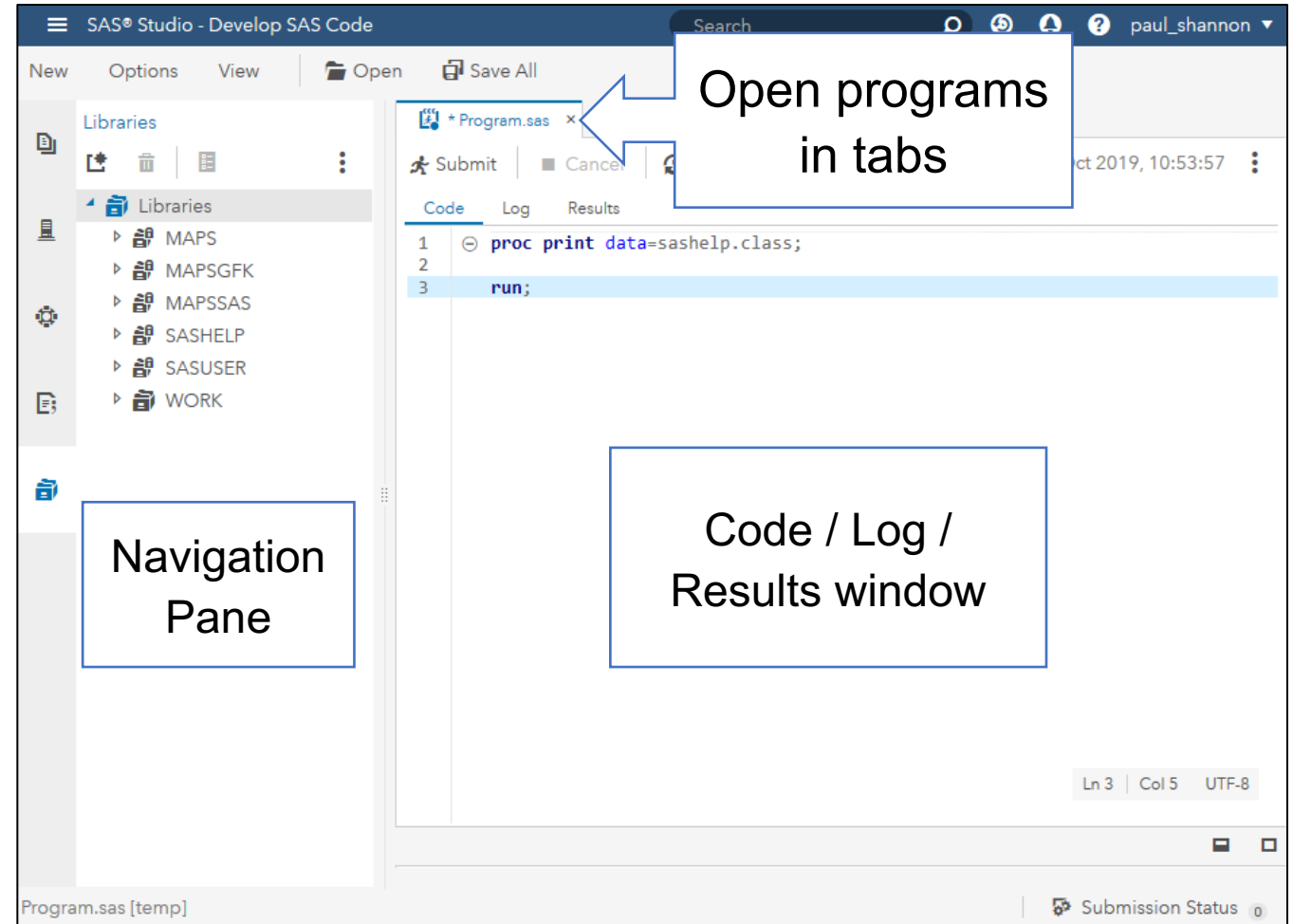


SAS Studio

Welcome to SAS Studio. The latest programming client.

It's an HTML5 application, most up-to-date browsers are supported.

Windows Internet Explorer is not supported, instead Microsoft Edge is.





Why Is SAS Studio Important?



- Consistent Interface, whether running on a server, local PC or in the cloud.
- Deployment is much simpler, meaning less impact on IT administrators.
- Aids novice programmers with easy assistance, whilst being an un-intrusive programming application for experienced programmers.
- Run from anywhere on any device.
 - With caveats – phones or small tablets may not be practical.



SAS Studio Versions

- There are multiple versions of SAS Studio.
- First launched in 2014 as v3.1.
 - Degree of maturity to the application.
- Version numbers are not strictly sequential:
 - V3.x is for SAS 9.4.
 - V4.x is for Viya.
 - V5.x is for Viya, alongside v4.x.

<https://support.sas.com/en/software/studio-support.html>



SAS Studio Editions

- There are three editions available:
- **Single User Edition** is delivered with Base SAS and runs on Windows. This is designed for PC's with SAS installed directly.
- **Basic Edition** runs on Windows and Unix. Any user with an operating system account can log on over a network.
- **Enterprise Edition** is available if your licence includes SAS Integration Technologies, and works with SAS Intelligence Platform.



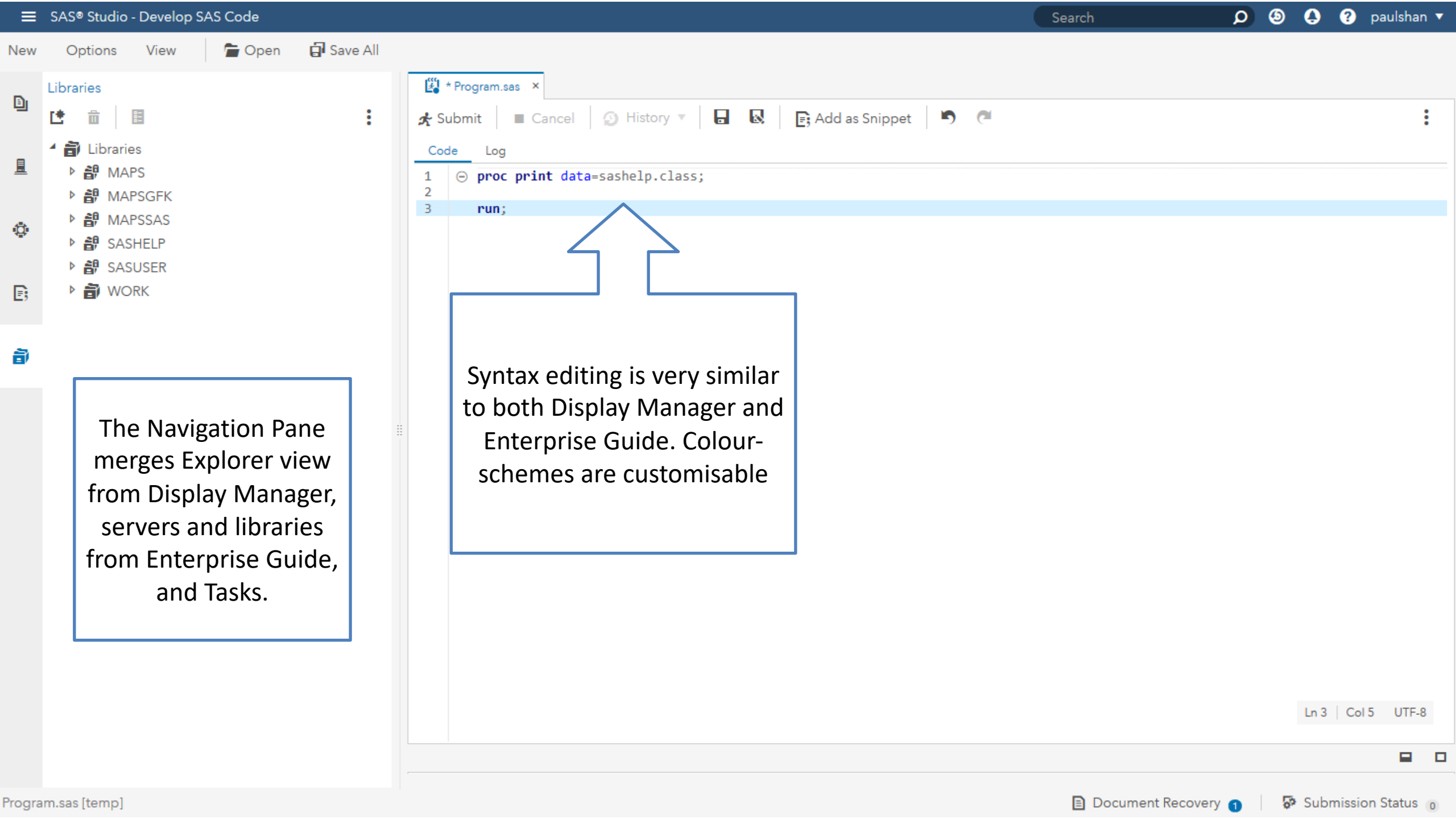
Features: Familiarity with Display Manager

- There are features that will be familiar to many users of SAS Display Manager.
- Explorer window, for browsing libraries and files.
- Syntax colouring, SAS programs, log and output windows.
- Text abbreviations – now called Snippets.
 - We will see a detailed demonstration shortly



Features: Familiarity with Enterprise Guide

- SAS Studio also provides familiar features for Enterprise Guide users.
- Intellisense – enhanced to provide better help tips
- Process Flow – now called Visual Programmer
 - Only available currently in v3.x
- Customisation – Custom Tasks far easier to develop.
 - We will take a closer look at this as well.



SAS® Studio - Develop SAS Code

NewOptionsViewOpenSave All

Explorer

- Folder Shortcuts
- SAS Content
 - Model Repositories
 - Products
 - Projects
 - Public
 - Users
 - paul_shannon
 - Application Data
 - My Favorites
 - My Folder
 - My History
 - Recycle Bin
 - Repos

*Program.sas

SubmitCancelHistoryAdd as Snippet

25 Oct 2019, 14:17:12

CodeLogResults

1proc print data=sashelp.class;

2

3run;

4

5proc rep

6

7

8%PUT

9

10%PUT

11

12%PUT

Procedures

REPORT

Keyword: REPORT

Context: [PROCEDURE DEFINITION] PROC REPORT

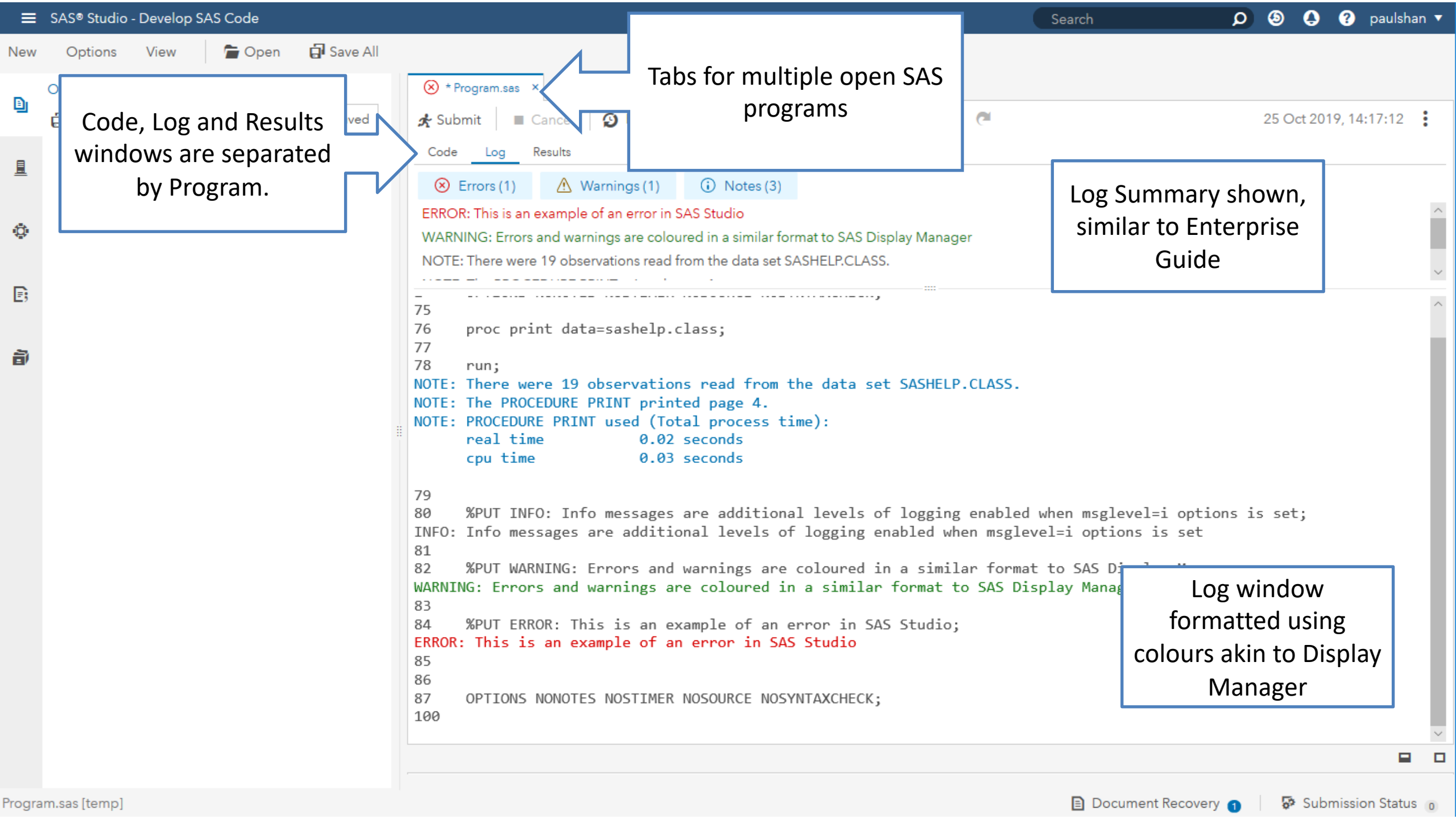
Syntax: PROC REPORT <option(s)>;
BREAK location break-variable</ option(s)>;
BY <DESCENDING> variable-1 <...<DESCENDING> variable-n> <NOTSORTED>;
COLUMN column-specification(s);
COMPUTE location <target> </ STYLE=<style-element-name> <[style-attribute-specification(s)]>>;
LINE specification(s);
... select SAS language elements ...
ENDCOMP;
COMPUTE report-item </ type-specification>;
CALL DEFINE (column-id, 'attribute-name', value);
... select SAS language elements ...
ENDCOMP;
DEFINE report-item / <usage> <attribute(s)> <option(s)>
<justification> <COLOR=color> <'column-header-1' <... 'column-header-n'>> <style>;
FREQ variable;
RBREAK location </ option(s)>;
WEIGHT variable;

The REPORT procedure combines features of the PRINT, MEANS, and TABULATE procedures with features of the DATA step in a single report-writing tool that can produce a variety of reports. You can use PROC REPORT in three ways:

Ln 5Col 9UTF-8

Document Recovery1Submission Status0

Help tips appear alongside intellisense, whilst typing



Code, Log and Results windows are separated by Program.

Tabs for multiple open SAS programs

Log Summary shown, similar to Enterprise Guide

Log window formatted using colours akin to Display Manager

```
* Program.sas
Submit | Cancel | Refresh

Code Log Results

Errors (1) Warnings (1) Notes (3)

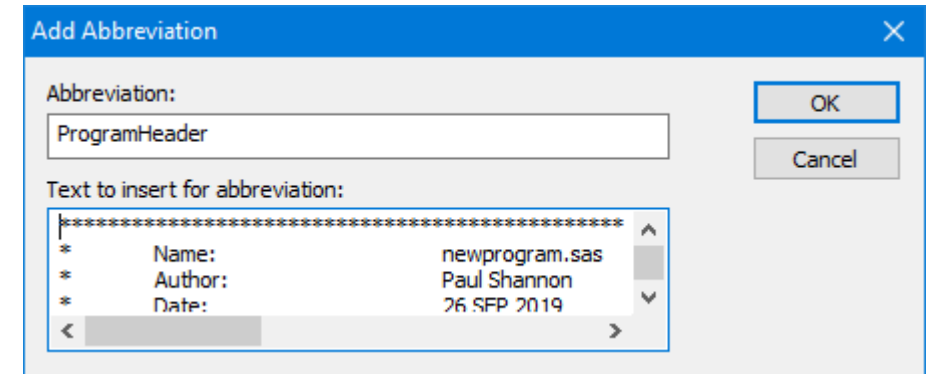
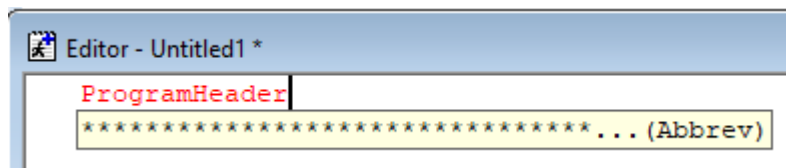
ERROR: This is an example of an error in SAS Studio
WARNING: Errors and warnings are coloured in a similar format to SAS Display Manager
NOTE: There were 19 observations read from the data set SASHELP.CLASS.

-----
75
76 proc print data=sashelp.class;
77
78 run;
NOTE: There were 19 observations read from the data set SASHELP.CLASS.
NOTE: The PROCEDURE PRINT printed page 4.
NOTE: PROCEDURE PRINT used (Total process time):
      real time          0.02 seconds
      cpu time           0.03 seconds

79
80 %PUT INFO: Info messages are additional levels of logging enabled when msglevel=i options is set;
INFO: Info messages are additional levels of logging enabled when msglevel=i options is set
81
82 %PUT WARNING: Errors and warnings are coloured in a similar format to SAS Display Manager;
WARNING: Errors and warnings are coloured in a similar format to SAS Display Manager
83
84 %PUT ERROR: This is an example of an error in SAS Studio;
ERROR: This is an example of an error in SAS Studio
85
86
87 OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
100
```

In Detail: Code Snippets

- SAS Display Manager offers an “abbreviations” function to allow easy entry of common code, based on a template.
- Display Managers’ editor is rather primitive.
- No user interface to see existing abbreviations, instead the programmer had to remember the name of an abbreviation.
- Abbreviations are case-sensitive.



In Detail: Code Snippets

- SAS Studio addresses these issues:
- A program window is used to edit a snippet. Allowing for full syntax editing, rather than a small plain-text dialog.
- Insertion of a snippet can be made through drag-n-drop or by abbreviation name, using the @ prefix.
- Abbreviations recognised by intellisense and are now case-insensitive.
- Sharable between users – to an extent.

In Detail: Code Snippets

- Creating a snippet is as simple as writing code in a program window, then clicking the “Add as Snippet” button
- Enter a name (shown in Navigation Pane)
- Enter an abbreviation (for programmatic reference)

Add to My Snippets

Name:

Program Header Block

Abbreviation:

@

ProgramHeader

Description:

Inserts a standard program header template

Folder:

My Snippets

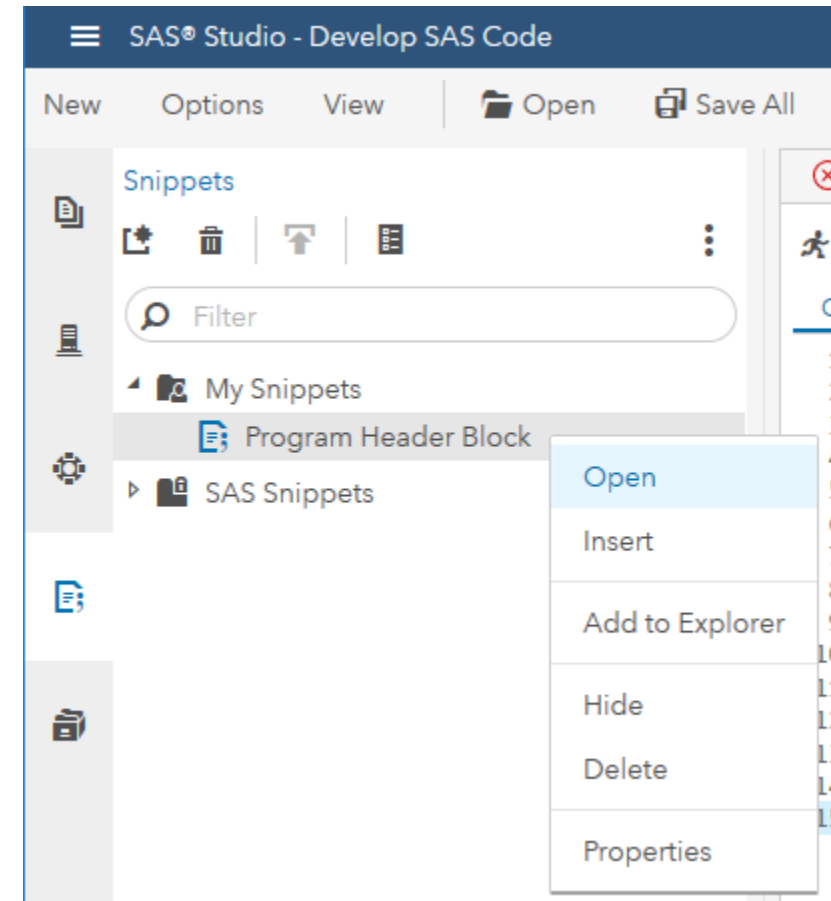


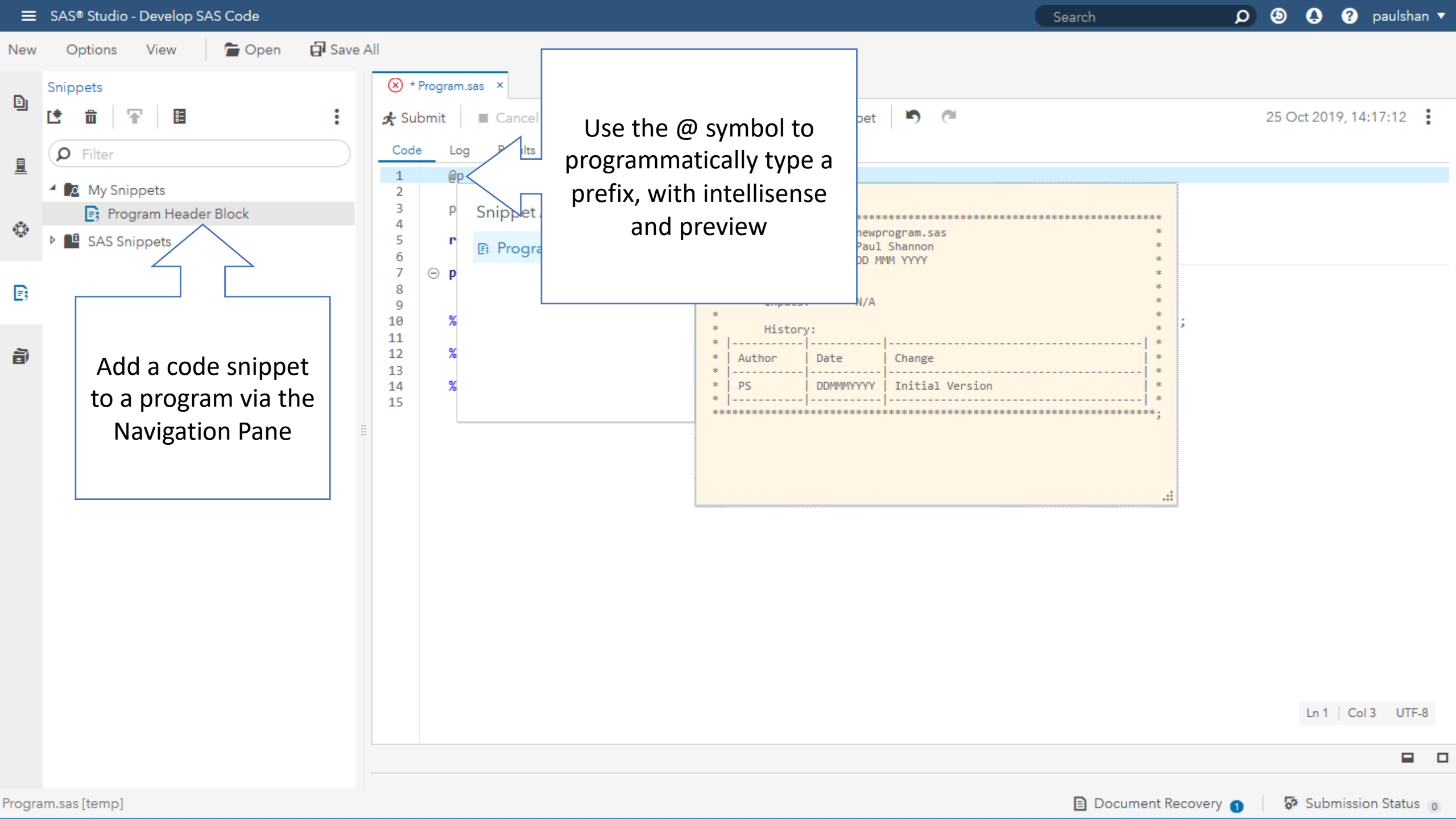
OK

Cancel

In Detail: Code Snippets

- Editing snippets is simple:
- Double-click (or right-click, open) on a snippet.
- Edit in the program window as required and save.
- Snippets can be shared with other programmers by adding to Explorer, to save in a folder accessible to other SAS users.





Add a code snippet to a program via the Navigation Pane

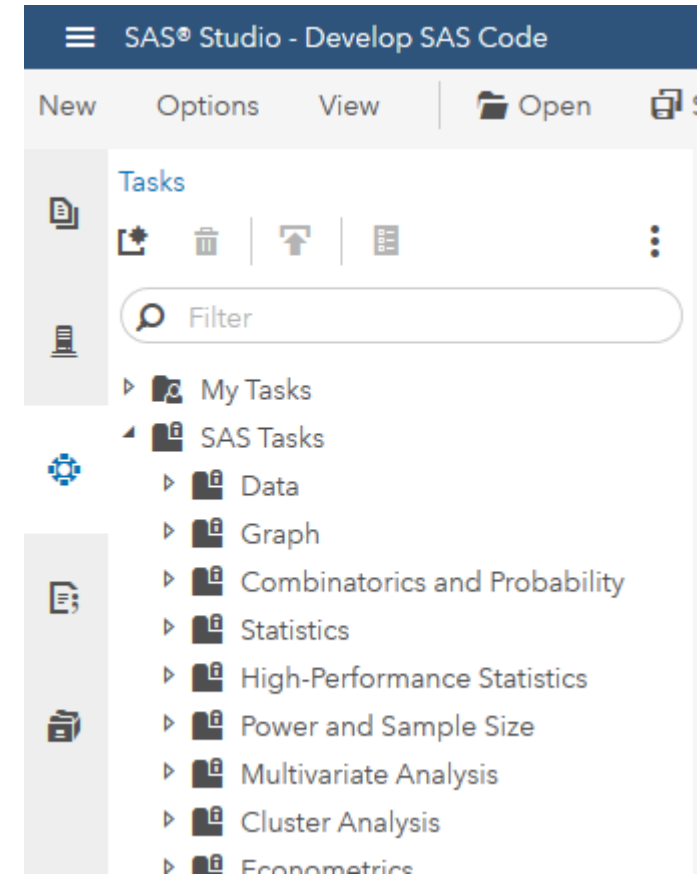
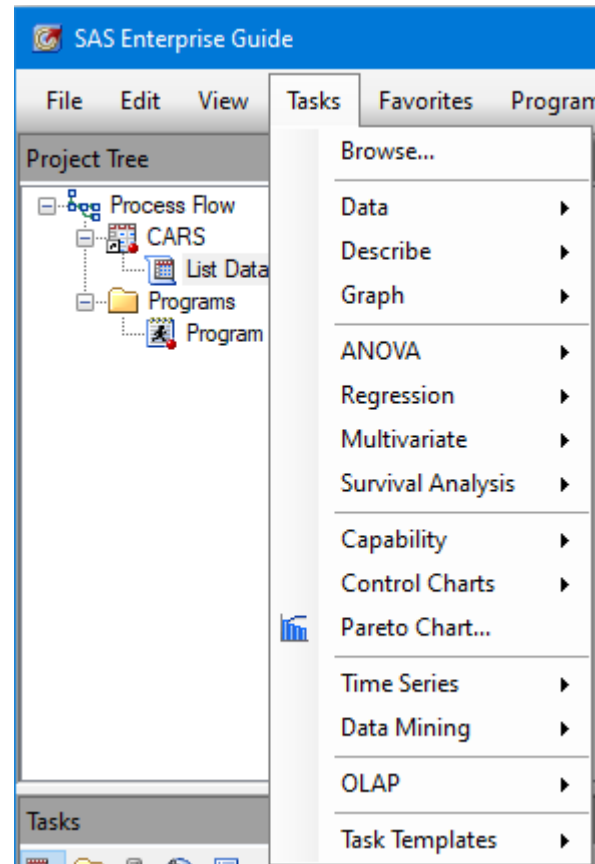
Use the @ symbol to programmatically type a prefix, with intellisense and preview

Add a code snippet to a program via the Navigation Pane

Use the @ symbol to programmatically type a prefix, with intellisense and preview

In Detail: Tasks

- SAS Enterprise Guide has a series of Tasks available.
- These offer a point-and-click interface to common SAS programming tasks.
- Tasks in SAS Studio are in the Navigation Pane.
- Tasks are re-categorised in places.



DATA OPTIONS OUTPUT INFORMATION

DATA

SASHELP.CARS

▼ ROLES

Analysis variables: *

- ☐ # EngineSize
- ☐ # Cylinders
- ☐ # Horsepower
- ☐ # MPG_Highway

Classification variables:

☐  Type

☐  Origin

Task Console

No items

[Code](#) [Log](#)

```

1  /*
2  *
3  * Task code generated by SAS® Studio 5.1
4  *
5  * Generated on '25/10/2019, 16:33'
6  * Generated by 'paul_shannon'
7  * Generated on server 'viya-demo'
8  * Generated on SAS platform 'Linux X64 3.10.0-957.5.1.el7.x86_64'
9  * Generated on SAS version 'V.03.04M0P050819'
10 * Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit
11 * Generated on web client 'http://viya.amadeus-cloud.co.uk/SASStudioV/main?lc
12 */
13
14 ods noproctitle;
15 ods graphics / imagemap=on;
16
17 proc means data=SASHELP.CARS chartype mean std min max n vardef=df;
18     var EngineSize Cylinders Horsepower MPG_Highway;
19     class Type Origin;
20 run;

```

SAS code
generated
automatically

In Detail: Tasks

- Custom Tasks are available in Both Enterprise Guide and SAS Studio
- Architecture for these client applications are very different:
- Custom Tasks in Enterprise Guide are developed in Microsoft Visual Studio, written in the a .NET language (e.g.: C#, VB).
- SAS Studio Tasks are created using an XML schema, directly within SAS Studio.

Tasks

Filter

My Tasks

Git Tasks

GitClone

List Table Attributes

Sort Data

SAS Tasks

Data

Graph

Combinatorics and Probability

Statistics

High-Performance Statistics

Power and Sample Size

Multivariate Analysis

Cluster Analysis

Econometrics

Forecasting

Network Optimization

SAS Viya Prepare and Explore

SAS Viya Unsupervised Learning

SAS Viya Supervised Learning

SAS Viya Evaluate and Implement

SAS Viya Text Analytics

SAS Viya Network Analysis and Visualization

SAS Viya Econometrics

SAS Viya Forecasting

* Program.sas

* Summary Statistics.ctl

* Task.ctl

GitClone.ctl

Save

Print

Add as Task

Open

<!-- Define the task options. -->

<Options>

<Option name="optiontab" inputType="string">OPTIONS</Option>

<Option name="resultsgrp" inputType="string">CONFIG</Option>

<Option name="lblRepos" inputType="string">Remote Repository:</Option>

<Option name="repos" inputType="inputtext" required="true" promptMessage="Enter the URL to your Git repository"

<Option name="lblLocal" inputType="string">Remote Repository:</Option>

<Option name="local" inputType="inputtext" required="true" promptMessage="Enter the local folder for your repository"

</Options>

</Metadata>

<!-- Define the UI for task options -->

<!-- The UI engine will determine the best representation. -->

<UI>

<Container option="optiontab">

<Group option="resultsgrp" open="true">

<OptionItem option="lblRepos"/>

<OptionItem option="repos"/>

<OptionItem option="lblLocal"/>

<OptionItem option="local"/>

</Group>

</Container>

</UI>

<!-- Define the velocity code template. -->

<CodeTemplate>

<![CDATA[

data _null_;

rc = gitfn_clone("\$repos", "\$local");

run;

]]>

</CodeTemplate>

</Task>

Task input objects defined

User interface definition

Output code template

Ln 55 | Col 8

Document Recovery 1 | Submission Status 0

In Detail: Source Control

- There are many methods employed to manage source code:
 - File naming convention.
 - SAS/AF applications.
 - History in program headers / comments.
 - 3rd-party software for managing source control connections.
- SAS now has a series of functions to enable integration with a Git repository.
- Git is a widely used source control solution. Originally developed by Linus Torvalds.



In Detail: Source Control

- Enables full version and source control, with multiple concurrent programmers.
- Users familiar with Git will be aware of terms such as “clone”, “fetch”, “pull”, “commit” and “push”.
- 24 Git functions are available in Base SAS (from 9.4M6).
- Currently only a programmatic interface exists in SAS.

In Detail: Source Control

- All Git functions in SAS start with the name `gitfn_`
- The example below demonstrates how to create a clone of a Git repository on your SAS server.

```
data _null_;  
    rc = gitfn_clone(  
        "https://github.com/sascommunities/sas-dummy-blog/",  
        "/home/paulshan/phuse/2019"  
    );  
RUN;
```

- This is not dependent on the use of SAS Studio.



Custom Tasks, For Source Control

- We can easily create Custom Tasks in SAS Studio to execute common Git commands, without resorting to writing code.
- Let's build a custom task to handle the first of the common Git commands: `gitfn_clone`
 - Essentially a command to connect a local directory to a remote code repository.
- Two parameters required:
 - URL to remote Git repository.
 - Target local repository (directory).

Building a Custom Task - Syntax

Specifies all the required inputs – labels, textboxes, dropdowns etc.

Designs the user interface for the task inputs

Template for resulting SAS Code

```
<Task schemaVersion="7.1">
  <!-- Define the registration properties for this task. -->
  <Registration>
  <Metadata>

    <!-- Define the data and roles for this task. -->
    <DataSources></DataSources>

    <!-- Define the task options. -->
    <Options>
      <Option name="optiontab" inputType="string">OPTIONS</Option>

      <Option name="lblRepos" inputType="string">Remote Repository:</Option>
      <Option name="repos" inputType="inputtext" required="true" promptMessage="Enter the URL to your Git repository."></Option>

      <Option name="lblLocal" inputType="string">Local Repository:</Option>
      <Option name="local" inputType="inputtext" required="true" promptMessage="Enter the local folder for your repository."></Option>
    </Options>
  </Metadata>

  <!-- Define the UI for task options -->
  <!-- The UI engine will determine the best representation. -->
  <UI>
    <Container option="optiontab">
      <OptionItem option="lblRepos"/>
      <OptionItem option="repos"/>
      <OptionItem option="lblLocal"/>
      <OptionItem option="local"/>
    </Container>
  </UI>

  <!-- Define the velocity code template. -->
  <CodeTemplate>
    <![CDATA[
      data _null_;
      rc = gitfn_clone(
        "$repos",
        "$local"
      );
      run;
    ]]>
  </CodeTemplate>
</Task>
```

SAS® Studio - Develop SAS Code

Search

paulshan

NewOptionsViewOpenSave All

Tasks

Filter

My Tasks

Git Tasks

GitClone

GitPull

Data Exploration

List Table Attributes

Sort Data

SAS Tasks

* Program.sas

* GitClone.ctlk

SubmitCancelHistoryAdd as Task

OPTIONSINFORMATION

Remote Repository:

https://github.com/sascommunities/sas-dummy-blog

Local Repository:

/home/paulshan/phuse/2019

Task Console

No items

CodeLog

1/*

2*

3* Task code generated by SAS® Studio 5.1

4*

5* Generated on '28/10/2019, 16:17'

6* Generated by 'paul_shannon'

7* Generated on server 'viya-demo'

8* Generated on SAS platform 'Linux LIN X64 3.10.0-957.5.1.el7.x86_64'

9* Generated on SAS version 'V.03.04M0P050819'

10* Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537

11* Generated on web client 'http://viya.amadeus-cloud.co.uk/SASStudioV/main?locale

12*/

13

14data _null_;

15rc=gitfn_clone("https://github.com/sascommunities/sas-dummy-blog",

16"/home/paulshan/phuse/2019");

17run;

Input dialog for Git Clone task

SAS code generated by task logic

GitClone.ctlk [temp]

Document Recovery 2Submission Status 0

Building a Custom Task – Advanced Logic

- This task can be extended include more advanced logic:
 - What if your Git repository requires credentials?
 - Git supports username/password or SSH keys
 - We can handle common errors with Git commands
 - Create local directory if it doesn't exist
-
- Let's add this functionality to our custom task.

Tasks

Filter

- My Tasks
 - Git Tasks
 - GitClone**
 - GitPull
 - Bar Chart
 - Data Exploration
 - List Table Attributes
 - One-Way ANOVA
 - Sort Data
- SAS Tasks
 - Data
 - Graph
 - Bar Chart
 - Bar-Line Chart
 - Box Plot
 - Bubble Plot
 - Heat Map
 - Histogram
 - Line Chart
 - Mosaic Plot
 - Pie Chart
 - Scatter Plot
 - Series Plot
 - Combinatorics and Probability
 - Statistics

```
* Program.sas
GitClone.ctm
* GitClone.ctlk

Add as Task
Open

22      <Option name="lblRepos" inputType="string">Remote Repository:</Option>
23      <Option name="repos" inputType="inputtext" promptMessage="Enter the URL to your Git repository."></Opt
24
25      <Option name="lblLocal" inputType="string">Local Repository:</Option>
26      <Option name="local" inputType="inputtext" promptMessage="Enter the
27      <Option name="chkCreateDir" inputType="checkbox" >Create local repo
28
29      <Option name="chkCredentials" inputType="checkbox">Require
30      <Option name="grpCredentials" inputType="string"></Optio
31      <Option name="lblUsername" inputType="string">User name:</
32      <Option name="lblPassword" inputType="string">Password:</Option>
33
34      <Option name="txtUsername" inputType="inputtext"/>
35      <Option name="txtPassword" inputType="passwordtext"></Option>
36
37      </Options>
38    </Metadata>
39
40    <!-- Define the UI for task options -->
41    <!-- The UI engine will determine the best representation. -->
42    <UI>
43      <Container option="optiontab">
44        <OptionItem option="lblRepos"/>
45        <OptionItem option="repos"/>
46        <OptionItem option="lblLocal"/>
47        <OptionItem option="local"/>
48        <OptionItem option="chkCreateDir"/>
49      </Container>
50      <Container option="credsTab">
51        <OptionItem option="chkCredentials"/>
52        <Group option="grpCredentials" open="true">
53          <OptionItem option="lblUsername"/>
54          <OptionItem option="txtUsername"/>
55          <OptionItem option="lblPassword"/>
56          <OptionItem option="txtPassword"/>
57        </Group>
58      </Container>
```

Additional elements
for credentials

Second tab for user
interface

Tasks

Filter

- My Tasks
 - Git Tasks
 - GitClone
 - GitPull
 - Bar Chart
 - Data Exploration
 - List Table Attributes
 - One-Way ANOVA
 - Sort Data
- SAS Tasks
 - Data
 - Graph
 - Bar Chart
 - Bar-Line Chart
 - Box Plot
 - Bubble Plot
 - Heat Map
 - Histogram
 - Line Chart
 - Mosaic Plot
 - Pie Chart
 - Scatter Plot
 - Series Plot
 - Combinatorics and Probability
 - Statistics

```

48      </Container>
49      <Container option="credsTab">
50        <OptionItem option="chkCredentials"/>
51        <Group option="grpCredentials" open="true">
52          <OptionItem option="lblUsername"/>
53          <OptionItem option="txtUsername"/>
54          <OptionItem option="lblPassword"/>
55          <OptionItem option="txtPassword"/>
56        </Group>
57      </Container>
58    </UI>
59
60    <Dependencies>
61      <Dependency condition="$chkCredentials ==1">
62        <Target option="grpCredentials" action="hide" condition="condition">
63          <Target option="grpCredentials" action="show" condition="condition">
64        </Dependency>
65      </Dependencies>
66
67    <Requirements>
68      <Requirement condition="$repos">
69        <Message>You must specify a link to the remote repository.</Message>
70      </Requirement>
71      <Requirement condition="$local">
72        <Message>You must specify a folder path to the local repository.</Message>
73      </Requirement>
74      <Requirement condition="$txtUsername || $chkCredentials ne 1">
75        <Message>You must enter a username for your Git credentials</Message>
76      </Requirement>
77      <Requirement condition="$txtPassword || $chkCredentials ne 1">
78        <Message>You must enter the password for your Git credentials</Message>
79      </Requirement>
80    </Requirements>
81
82
83    <!-- Define the velocity code template. -->
84    <CodeTemplate>
85

```

Dependencies control the UI logic for the task

Requirements configures data validation for task inputs

Tasks

Filter

- Pie Chart
- Scatter Plot
- Series Plot
- Combinatorics and Probability
- Statistics
- High-Performance Statistics
- Power and Sample Size
- Multivariate Analysis
- Cluster Analysis
 - Compute Similarities and Dista
 - Cluster Variables
 - K-Means Clustering
 - Cluster Observations
 - Estimate Within-Cluster Covari
- Econometrics
- Forecasting
- Network Optimization
- SAS Viya Prepare and Explore
- SAS Viya Unsupervised Learning
- SAS Viya Supervised Learning
- SAS Viya Evaluate and Implement
- SAS Viya Text Analytics
- SAS Viya Network Analysis and Op
- SAS Viya Econometrics
- SAS Viya Forecasting

```

76      </Requirement>
77      <Requirement condition="$txtPassword || $chkCredentials ne 1">
78          <Message>You must enter the password for your Git credentials</Message>
79      </Requirement>
80  </Requirements>
81
82  <!-- Define the velocity
83  <CodeTemplate>
84      <![CDATA[
85          %macro TASK_GitClone
86          #if $chkCreateDir
87              filename f "$local"
88              %if %sysfunc(fexist(f)) = 0 %
89                  %sysfunc(dcreate(f));
90              %end;
91          #end
92
93          data _null_;
94              rc = gitfn_clone(
95                  "$repos"
96                  ,"$local"
97                  #if($chkCredentials == "1")
98                      ,"$txtUsername"
99                      ,"$txtPassword"
100                  #end
101              );
102          run;
103          %mend _TASK_GitClone_;
104          %_TASK_GitClone_;
105
106          ### CLEANUP - remove macro program ###
107          %symdel(_TASK_GitClone_);
108          ]]]>
109      </CodeTemplate>
110  </Task>
111

```

Input fields are referenced by their "name", preceded by \$ symbol

Conditional code enabled by #if, #end statements

Tasks

Filter

- Pie Chart
- Scatter Plot
- Series Plot
- Combinatorics and Probability
- Statistics
- High-Performance Statistics
- Power and Sample Size
- Multivariate Analysis
- Cluster Analysis
 - Compute Similarities and Dista
 - Cluster Variables
 - K-Means Clustering
 - Cluster Observations
 - Estimate Within-Cluster Covari
- Econometrics
- Forecasting
- Network Optimization
- SAS Viya Prepare and Explore
- SAS Viya Unsupervised Learning
- SAS Viya Supervised Learning
- SAS Viya Evaluate and Implement
- SAS Viya Text Analytics
- SAS Viya Network Analysis and Op
- SAS Viya Econometrics
- SAS Viya Forecasting

* Program.sas * GitClone.ctlk x

Submit Cancel History Add as Task

REPOSITORY CREDENTIALS INFORMATION

Remote Repository:

Local Repository:

☐ Create local repository if it doesn't exist

Task Console

- You must specify a link to the remote repository.
- You must specify a folder path to the local repository.

Code Log Edit

```
1 /* Code for this task cannot be generated because of an error.
2  * Please use the Task Console to view and fix the errors.
3  */
4
```

Read Only Ln 1 Col 1 UTF-8

Default view for a new task. Note Task console shows validation messages.

Code is not generated when validation fails

SAS® Studio - Develop SAS Code

Search

paulshan

NewOptionsViewOpenSave All

Tasks

Filter

Pie Chart

Scatter Plot

Series Plot

Combinatorics and Probability

Statistics

High-Performance Statistics

Power and Sample Size

Multivariate Analysis

Cluster Analysis

Compute Similarities and Dista

Cluster Variables

K-Means Clustering

Cluster Observations

Estimate Within-Cluster Covari

Econometrics

Forecasting

Network Optimization

SAS Viya Prepare and Explore

SAS Viya Unsupervised Learning

SAS Viya Supervised Learning

SAS Viya Evaluate and Implement

SAS Viya Text Analytics

SAS Viya Network Analysis and Op

SAS Viya Econometrics

SAS Viya Forecasting

* Program.sas

* GitClone.ctlk

SubmitCancelHistory

REPOSITORYCREDENTIALSINFORMATION

☒ Requires Credentials?

User name:

Password:

Task Console

You must enter a username for your Git credentials

You must enter the password for your Git credentials

Read OnlyLn 1Col 1UTF-8

Validation of credentials only active when checkbox selected

This task cannot be generated because of an error. View the Task Console to view and fix the errors.

GitClone.ctlk [temp]Document RecoverySubmission Status

Tasks

Filter

- Pie Chart
- Scatter Plot
- Series Plot
- Combinatorics and Probability
- Statistics
- High-Performance Statistics
- Power and Sample Size
- Multivariate Analysis
- Cluster Analysis
 - Compute Similarities and Distances
 - Cluster Variables
 - K-Means Clustering
 - Cluster Observations
 - Estimate Within-Cluster Covariance
- Econometrics
- Forecasting
- Network Optimization
- SAS Viya Prepare and Explore
- SAS Viya Unsupervised Learning
- SAS Viya Supervised Learning
- SAS Viya Evaluate and Implement
- SAS Viya Text Analytics
- SAS Viya Network Analysis and Optimization
- SAS Viya Econometrics
- SAS Viya Forecasting

* Program.sas * GitClone.ctlk x

Submit Cancel History Add as Task

REPOSITORY CREDENTIALS INFORMATION

☒ Requires Credentials?

User name:
paul.shannon@amadeus.co.uk

Password:
●●●●●●●●

Task Console
No items

SAS
Automatically
encodes
password input

Code Log Edit

```
1 /*
2 *
3 * Task code generated by SAS® Studio 5.1
4 *
5 * Generated on '29/10/2019, 11:32'
6 * Generated by 'paul_shannon'
7 * Generated on server 'viya-demo'
8 * Generated on SAS platform 'Linux LIN X64 3.10.0-957.5.1.el7.x86_64'
9 * Generated on SAS version 'V.03.04M0P050819'
10 * Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.4046.130 Safari/537.36'
11 * Generated on web client 'http://viya.amadeus-cloud.co.uk/SASStudioV/ma
12 */
13
14 %macro _TASK_GitClone_();
15     filename f "/home/paulshan/phuse/2019";
16
17     %if %sysfunc(fexist(f))=0 %then
18         %do;
19             %sysfunc(dcreate(f));
20         %end;
21
22     data _null_;
23         rc=gitfn_clone("https://github.com/sascommunities/sas-dump",
24             "/home/paulshan/phuse/2019",
25             "paul.shannon@amadeus.co.uk",
26             "{SAS002}E19F24391178DDC80CF880FA502328ED");
27     ;
28
29 %mend _TASK_GitClone_;
30
31 %_TASK_GitClone_;
32 %symdel(_TASK_GitClone_);
```

Read Only Ln 1 Col 1 UTF-8

Conclusions

- SAS Studio offers a user interface that is recognisable to users familiar with Display Manager or Enterprise Guide.
- User interface is common across all devices with a browser, no need to install software.
- Common-used features are ported into SAS Studio and in many cases, have been enhanced.
- SAS Studio is consistent in use across all device types (though use on phones is restrictive).

Conclusions

- Various editions of SAS Studio offer a solution for different types of deployment.
- Basic and Enterprise Editions provide a single point of installation.
 - Enables far simpler deployment and maintenance for platform administrators.
- Customisation/Extensibility is far easier than before.
 - Custom tasks can be developed in minutes.

Where Next?

- There is currently minimal support for mobile devices.
 - You can connect to SAS Studio on your phone/tablet.
 - * Subject to network configuration.
- Support for code writing while offline?
 - This is a limitation currently, although ever-reducing in relevance over time.
- How long before an “app” is produced?
 - The latest release of Enterprise Guide has a user interface more akin to SAS Studio.
 - Is there a gradual merging of technology taking place?



Do you have any questions?



Paul Shannon
Amadeus Software Ltd.

Tel: +44 (0)1993 848010
E-mail: paul.shannon@amadeus.co.uk
Web: www.amadeus.co.uk