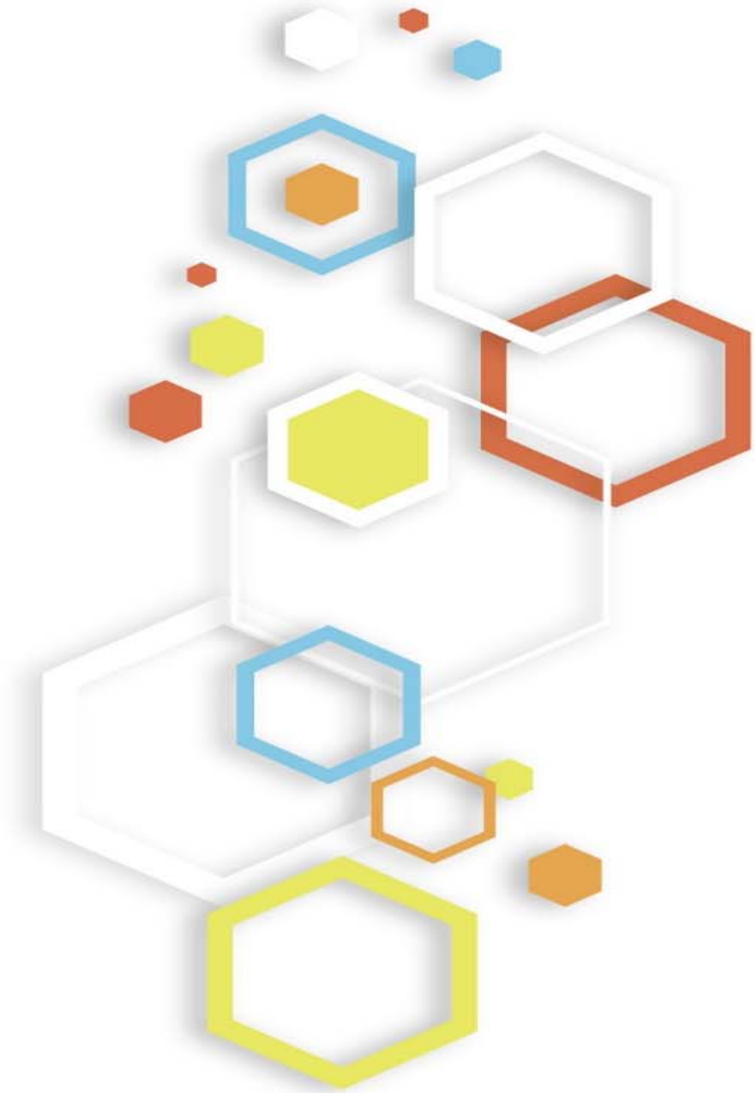
SAS® Forum
Switzerland 2014
make connections • share ideas • be inspired



SAS® 9.4 – What's new in SAS Foundation

Hans-Rainer Pauli
Senior Solution Specialist Business Analytics, SAS

sas THE POWER TO KNOW.

Agenda

- New Features in Base SAS
- Output Delivery System
- ODS GRAPHICS
- New Programming Language DS2
- New SAS Web Application: SAS Studio

Overview

- New / Enhanced Procedures
- Extended Attributes for Datasets and Variables
- Variable checking for the DATA Step
- ZIP File Access Method
- DOSUBL function
- More

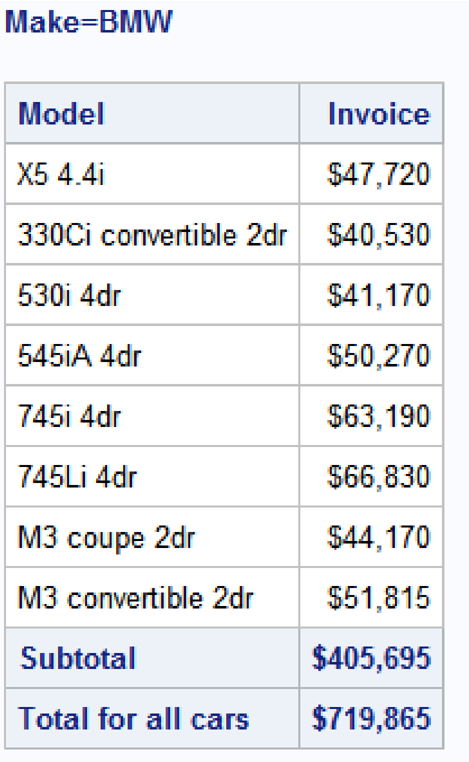
New / Enhanced Procedures

- Proc PRINT

  New option for labeling subtotal and totals

  - GRANDTOTAL_LABEL
  - SUMLABEL

```
proc print data=sashelp.cars
  grandtotal_label='Total for all cars'
  sumlabel='Subtotal'
  noobs
;
  where make in ('Audi','BMW')
   and  invoice > 40000;
  by make;
  var model invoice;
  sum   Invoice;
 run;
```



Make=BMW

Model	Invoice
X5 4.4i	$47,720
330Ci convertible 2dr	$40,530
530i 4dr	$41,170
545iA 4dr	$50,270
745i 4dr	$63,190
745Li 4dr	$66,830
M3 coupe 2dr	$44,170
M3 convertible 2dr	$51,815
Subtotal	**$405,695**
Total for all cars	**$719,865**

New / Enhanced Procedures

- Proc JSON, JSON = JavaScript Object Notation

  Read a SAS Dataset and write it out to external file using JSON represenation
  Provides statements to write additional data and control JSON containers

  Often used by JavaScript frameworks as data source

Proc JSON Example

```
filename myJson "basic_sasDataset.json";

proc json out=myJson pretty;
  export sashelp.class;
run;

filename myJson clear;
```

```
{
  "SASJSONExport": "1.0 PRETTY",
  "SASTableData+CLASS": [
    {
      "Name": "Alfred",
      "Sex": "M",
      "Age": 14,
      "Height": 69,
      "Weight": 112.5
    },
    {
      "Name": "Alice",
      "Sex": "F",
      "Age": 13,
      "Height": 56.5,
      "Weight": 84
    },
    ...
 ]
}
```

More Procedures

- Proc AUTHLIB

  For Metadata-Bound Libraries (only with Metadata Server)

- Proc STREAM

  Process input stream that may contain macro triggers, macro text is expanded and everything sent to a file

  Handy for Stored Process applications

- Proc PRESENV

  Preserve SAS Environment

Extended Attributes for Datasets and Variables

Extended attributes are customized metadata for your SAS files (BASE engine only).

They are user-defined characteristics that you associate with a SAS data set or variable

Extended attributes are organized into name=value pairs

Use Proc DATASETS to manage the extended attributes

- XATTR ADD, XATTR DELETE, XATTR REMOVE, XATTR SET, XATTR UPDATE

Proc CONTENTS will print information on extended attributes

There is a DICTIONARY.XATTRS table to be used with Proc SQL for querying

Variable checking for the DATA Step

- Help with typos and do not overlook the NOTE: Variable … is uninitialized.

New System Option VARINITCHK

```
data newClass;
  set sashelp.class;
  newAge = ag;
run;
```

```
options varinitchk = NOTE;
NOTE: Variable ag is uninitialized.

options varinitchk = WARNING;
WARNING: Variable ag is uninitialized.

options varinitchk = ERROR;
ERROR: Variable ag is uninitialized.
```

ZIP File Access Method

- Allows you to read and write files within a ZIP file

  Implemented as a file access method for the FILENAME statement

```
filename foo ZIP 'c:\temp\sample.zip' member="cars.csv" ;
proc export data=sashelp.cars dbms=csv outfile=foo;
run;

filename foo ZIP 'c:\temp\sample.zip' member="reports/report.pdf" ;
ods pdf file=foo;
proc print data=sashelp.class;
run;

proc sgplot data=sashelp.cars;
  vbar type / response=invoice stat=mean;
run;
ods pdf close;
```

DOSUBL Function

- Run a SAS Program while executing a DATA Step

DOSUBL allows you to run some SAS code from within a DATA Step

Execute complete steps only

Execute global statements

Can create tables in WORK library and read them in the calling SAS environment

DOSUBL Function

- An example

```
data dosubl_sample;
  set work.class_s;
  by sex;

  if first.sex = 1 then do;
    length t_sasstmt $ 1024;
    t_sasstmt = catx(" ",
      "proc sql noprint;"
      , "select avg(age) into :avgAge"
      , "from sashelp.class"
      , "where sex =", quote(sex), ";", '%put NOTE: &=avgAge;quit;'
    );
    t_rc = dosubl(t_sasstmt);
    retain avgAge;
    avgAge = input( symget("avgAge"), 12.);
  end;
  avgDiff = age - avgAge;
run;
```

More ...

CUROBS= option for SET and MODIFY to get the current observation number

You can set a TIMEZONE for the SAS session

- TZONES2U and TZONEU2S allow for datetime value conversion between SAS and UTC time

SHA256, convert a string to a message digest based on the SHA256 algorithm

FCOPY function to copy files

AES encryption for SAS data sets using the ENCRYPTKEY= and ENCRYPT= data set options

DO_OVER method for the Hash object in the DATA step

…

Additional Information

- There is much more to explore
- What's New in Base SAS 9.4: Details [HTML] [PDF]

Overview

New Destination: HTML5, EPUB/EPUB3, POWERPOINT

New Procedures: ODSLIST, ODSTABLE, ODSTEXT

ODS LAYOUT

Report Writing Interface for the DATA Step

ODS Graphics

New ODS Destinations

Write Procedure results to EPUB, good for accessibility

Send your SAS Analysis (tables and graphs) directly to Powerpoint, supports PPTX format

Proc ODSLIST, ODSTABLE, ODSTEXT

- Proc ODSTEXT

  Provides an easy way to write blocks of text as paragraphs

- Proc ODSTABLE

  Define a table defintion and bind with data in one proc, no need to create first a table definition

- Proc ODSLIST

  Provides an easy way to create bulleted lists, useful together with EPUB and POWERPOINT destinations

ODS LAYOUT

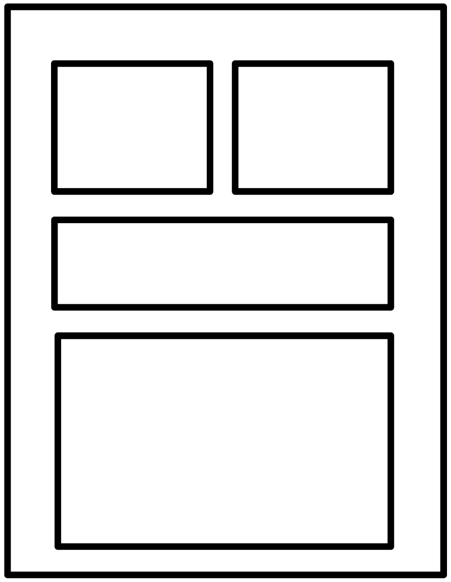
- ODS LAYOUT is production with SAS9.4

- ODS LAYOUT ABSOLUTE

  Only for printer destination

  Full control over where to place output on the paper

  Region may contain text, tables
  graphics, images



ODS LAYOUT

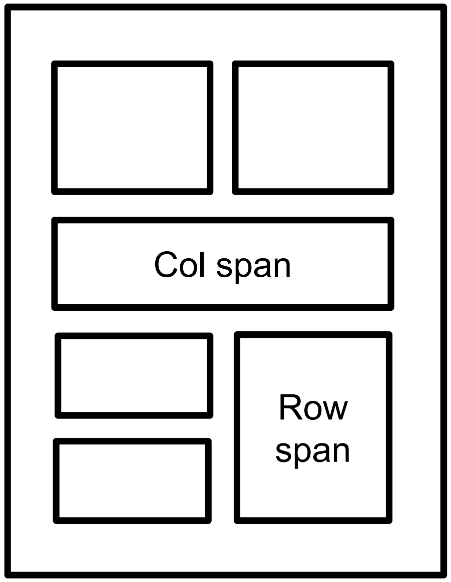
- ODS LAYOUT GRIDDED

  Supported with HTML, SASREPORT, PRINTER and POWERPOINT destination

  you specify columns and rows

  Column span and row span supported

  Example has 2 columns and 4 rows



Report Writing Interface for the DATA Step

Report Writing Interface is production with SAS9.4

Allows to create reports that are not only column oriented

Implemented as a DATA step object with methods

Support for ODS LAYOUT from within the DATA step

Build your own tables, similar to tables in HTML

Create forms and free form reports

Additional Information

- There is much more to explore
- What's New in the Output Delivery System
- Look out for:

  ODS and Cascading Style Sheets

- Tip Sheets:

  SAS 9 ODS CSS Tip Sheet
  SAS 9 ODS EPUB Tip Sheet
  SAS 9 ODS Layout Tip Sheet
  SAS 9 ODS EXCELXP Tip Sheet
  SAS 9 Report Writing Interface Tip Sheet
  SAS 9 ODS List and Text Block Tip Sheet

ODS GRAPHICS

- ODS GRAPHICS is part of Base SAS

  Can create a wide variety of different graphics

  Use SGPLOT, SGPANEL, SGSCATTER procedures for ease of use

  Use Graph Template Language (GTL) for more demanding graphics together with SGRENDER

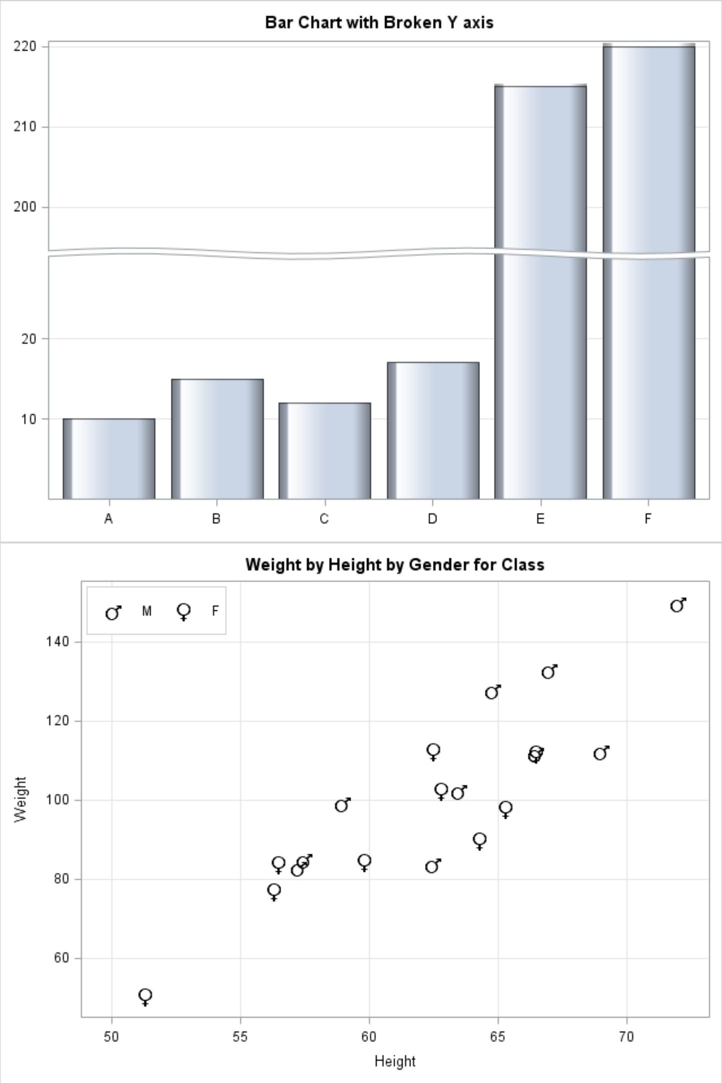
  Always creates an image, several output formats are available also depending on ODS destination (PNG, GIF, SVG, …)

  Supports animated GIF and SVG files

ODS GRAPHICS new with SAS9.4

- Many new features, too many to list them all

- Handy new features:
  - Support for POLYGON statement
  - Broken axis (x or y axis, use GTL and Proc SGRENDER)
  - Support for special marker characters and marker images



Additional Information

- There is much more to explore

  SAS 9.4 ODS Graphics: Getting Started with Business and Statistical Graphics PDF (2.1MB) | HTML

  SAS 9.4 ODS Graphics: Procedures Guide, Second Edition PDF (25.6MB) | HTML

  SAS 9.4 ODS Graphics Designer: User's Guide, Second Edition PDF (6.76MB) | HTML

  SAS 9.4 Graph Template Language: Reference, Second Edition PDF (26.56MB) | HTML

  SAS 9.4 Graph Template Language: User's Guide, Second Edition PDF (12.11MB) | HTML

  Blog Graphically Speaking

What is DS2?

- DS2 is a new SAS programming language

  has syntax that is similar to the Base SAS DATA step

  provides advanced techniques for data manipulation like methods, packages, and scoping

- Does not replace the DATA step language
- DATA step DNA is clearly visible

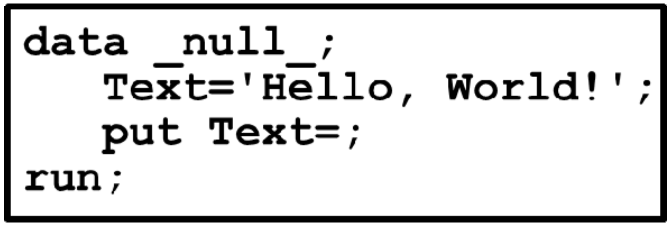
  DATA and SET statements, but no INFILE, INPUT, MERGE, UPDATE, MODIFY statement

  IF...THEN...ELSE, DO loops

  Expressions and Functions

  Arrays

Base SAS DATA Step



```
data _null_;
   Text='Hello, World!';
   put Text=;
run;
```

DS2 DATA Step

```
proc ds2;
data _null_;
   method init();
      Text='Hello, World!';
      put Text=;
   end;
enddata;
run;
quit;
```

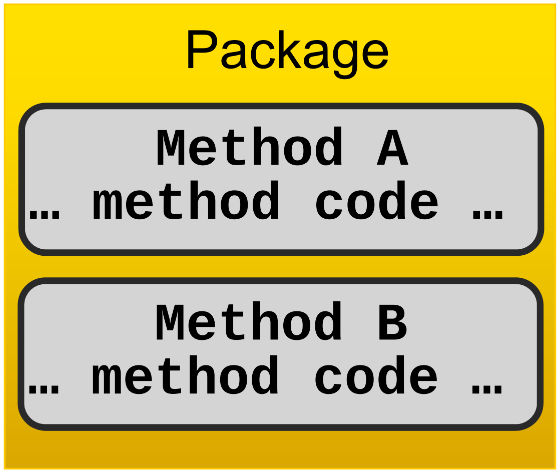
Why DS2

- Do SAS-y things in places SAS has never run before

- Modernized programming techniques and structures
  - Strict variable declaration (optional)
  - Variable scoping
  - Methods
    - Predefined (INIT, RUN, TERM)
    - User-defined
    - Variable encapsulation
  - Packages
    - Predefined (HASH, SQLSTMT, Logger, FCMP, …)
    - User-defined

What Is New in DS2?

- DS2 improves the extensibility and reusability of code through the use of methods and packages.



Package

```
    Method A
… method code …

    Method B
… method code …
```

Methods and packages can be pre-defined or user-defined.

What Is New in DS2?

- DS2 natively supports ANSI SQL data types for precise data manipulation.
- Examples:

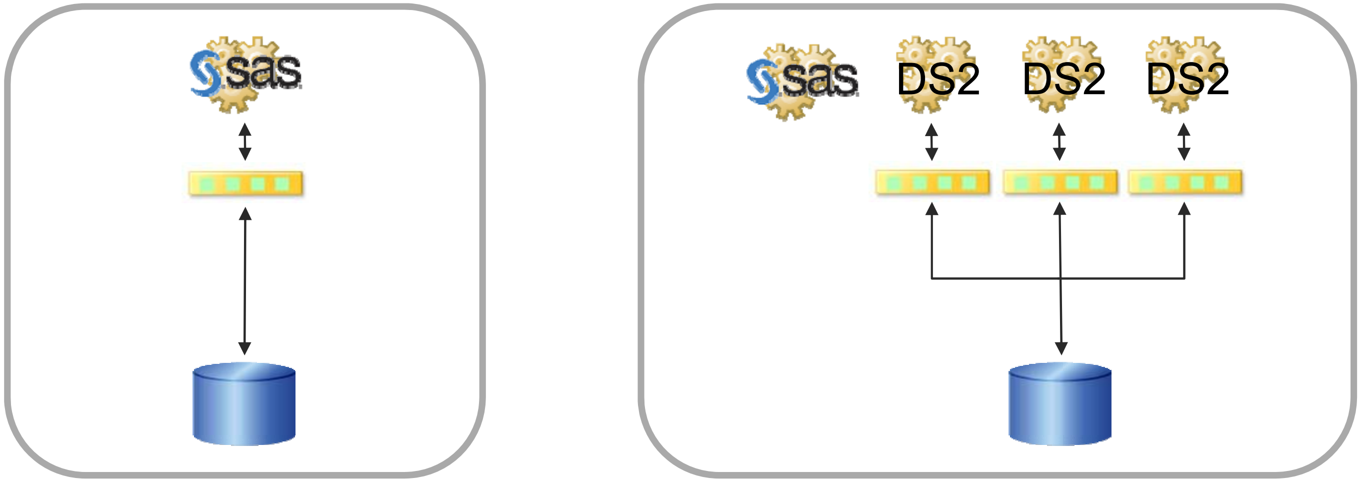
Data Type	Examples
Fractional Numerics	DECIMAL, DOUBLE, FLOAT, REAL
Integer Numerics	BIGINT, INTEGER, SMALLINT, TINYINT
Date and Time	DATE, TIME, TIMESTAMP
Character	CHAR, NCHAR, VARCHAR, NVARCHAR

SAS numeric variables are processed as DOUBLE.
SAS character variables are processed as CHAR.

What Is New in DS2?

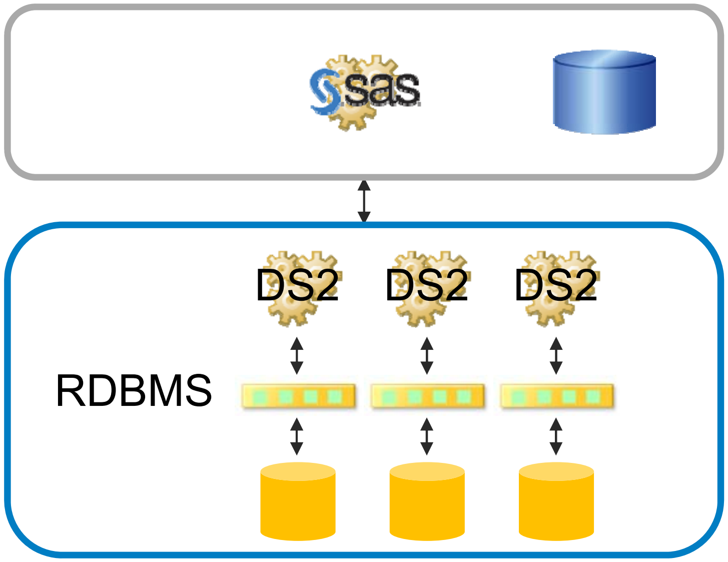
The Base SAS DATA step processes each observation sequentially.

DS2 can process observations in parallel.



What Is New in DS2?

- With the SAS In-Database Code Accelerator, DS2 can also be executed in-database (Greenplum, Teradata).



How Are DS2 Programs Created?

- DS2 programs are written using the following Base SAS procedures:

  PROC DS2

  PROC HPDS2

Where Can DS2 Programs Execute?

- PROC DS2 can execute DS2 programs in the following:

  Base SAS

  a database with the SAS In-Database Code Accelerator installed (Greenplum, Teradata)

Where Can DS2 Programs Execute?

- PROC HPDS2 can execute DS2 programs in the following locations:

SAS High-Performance Grid

SAS High-Performance Analytics Server

SAS In-Memory Analytics Server

When to Use DS2?

- DS2 can access data from the following data sources:

  - Aster
  - DB2 for UNIX
  - DB2 for PC
  - Greenplum
  - MYSQL
  - Netezza
  - ODBC-compliant databases
  - Oracle
  - SAS data set
  - SAP
  - SPD Engine data set
  - Sybase IQ
  - Teradata

- It uses the FedSQL Language to access data from the DBMS

When to Use DS2?

- DS2 programs are best suited for applications that do the following:

  take advantage of threaded processing

  - are computationally complex
  - can execute in massively parallel processing (MPP) databases

  require the precision that the DS2 data types offer

  leverage the convenient reusability of DS2 methods and packages

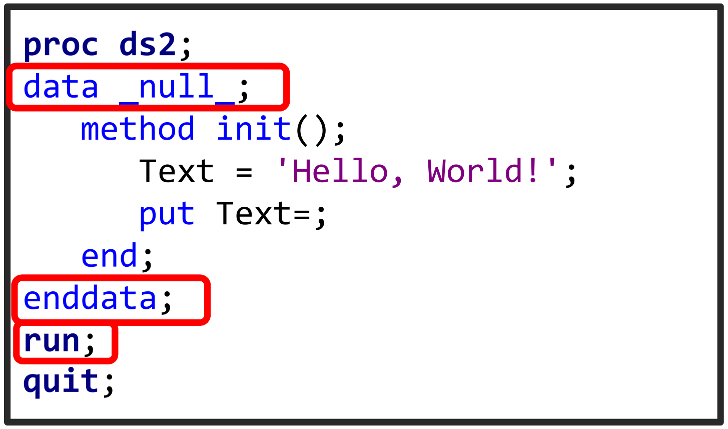
Basic DS2 Syntax

- The DS2 DATA step

  begins with a DATA statement

  ends with an ENDDATA statement

  requires a RUN statement to execute.



```
proc ds2;
data _null_;
   method init();
      Text = 'Hello, World!';
      put Text=;
   end;
enddata;
run;
quit;
```

ds2_helloWorld

Basic DS2 Syntax

- New in DS2: Methods

  Methods are named, executable blocks of code.

  - The METHOD statement names the method.
  - The END statement terminates the method.

  All executable code is encapsulated in methods.

  - Method identifiers (names) are global.
  - Variable identifiers can be global or local.

```
proc ds2;
data _null_;
   method init();
      Text = 'Hello, World!';
      put Text=;
   end;
enddata;
run;
quit;
```

Basic DS2 Syntax

- System methods execute automatically.

INIT()	–	once at start
RUN()	–	once for every row
TERM()	–	once at termination

```
data _null_;
   method init();
      dcl varchar(20) Text;
      Text='**> Starting';
      put Text;
   end;
   method run();
      set work.newClass;
      put _all_;
   end;
   method term();
      dcl char(11) Text;
      Text='**> All done!';
      put Text;
   end;
enddata;
run;
```

Familiar from the DATA Step

- Many things work just like DATA step
  - A SET statement reads in a table (dataset)
  - BY group processing
    - First. and Last.
  - DO groups & loops
    - DO I= *start* TO *stop* BY *interval*
    - DO WHILE () / DO UNTIL ()
    - CONTINUE, LEAVE, END
  - RETURN, GOTO, STOP, OUTPUT
  - KEEP, DROP, RETAIN, RUN
    - In DS2, these are valid only outside a METHOD definition

DS2 Basic Method Example

- User-defined Methods

  May accept arguments

  May return a value

  Execute when referenced in a DATA step

  May be referenced multiple times

```
PROC DS2;
data c2f_conv(overwrite=yes);
  declare double degC degF;
  method c2f(double tC) returns double;
    declare double tempValue;
    /* Celsius to Farenheit */
    tempValue = ( ( tC * 9 ) / 5 ) + 32;
    return ( tempValue );
  end;
  method init();
    do degC = 0 to 30 by 2;
       degF = c2f(degC);
       output;
    end;
  end;
enddata;
run;
QUIT;
```

The “Subtle Six” Dissimilarities

1. Executable code must be part of a METHOD definition.
2. Data set, thread and package definitions are not overwritten by default, use overwrite=yes.
3. New variables should be declared.
4. Keywords are ***reserved*** words, there are quite a few ;-)
5. PUT statement column and line parameters
   are not supported (there is no PUTLOG).
6. Follow ANSI quoting standards:
   - double quotation marks ⇒ identifiers (case sensitive)
   - single quotation marks ⇒ literals
   - Use %TSLIT to single-quote resolved macro text.

Additional Information

- **DS2 and FedSQL**
- SAS 9.4 DS2 Language Reference, Second Edition
  PDF (24.47MB) | HTML
- SAS 9.4 FedSQL Language Reference, Second Edition
  PDF (13.13MB) | HTML

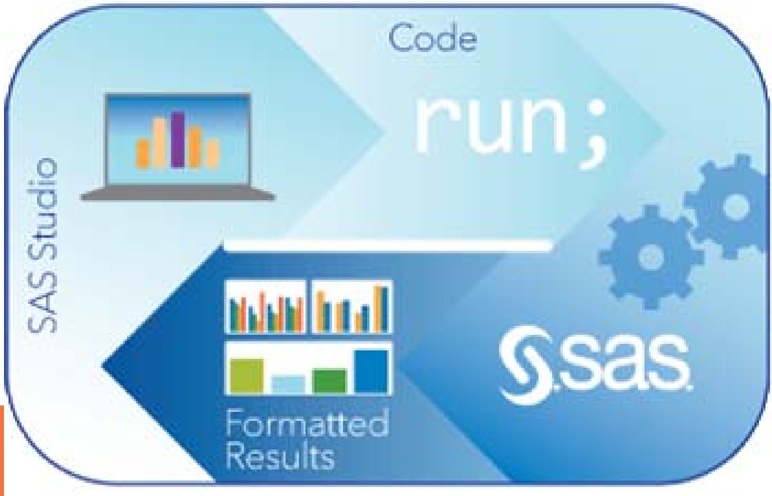
SAS Studio Overview

A web browser-based interface for SAS programmers.

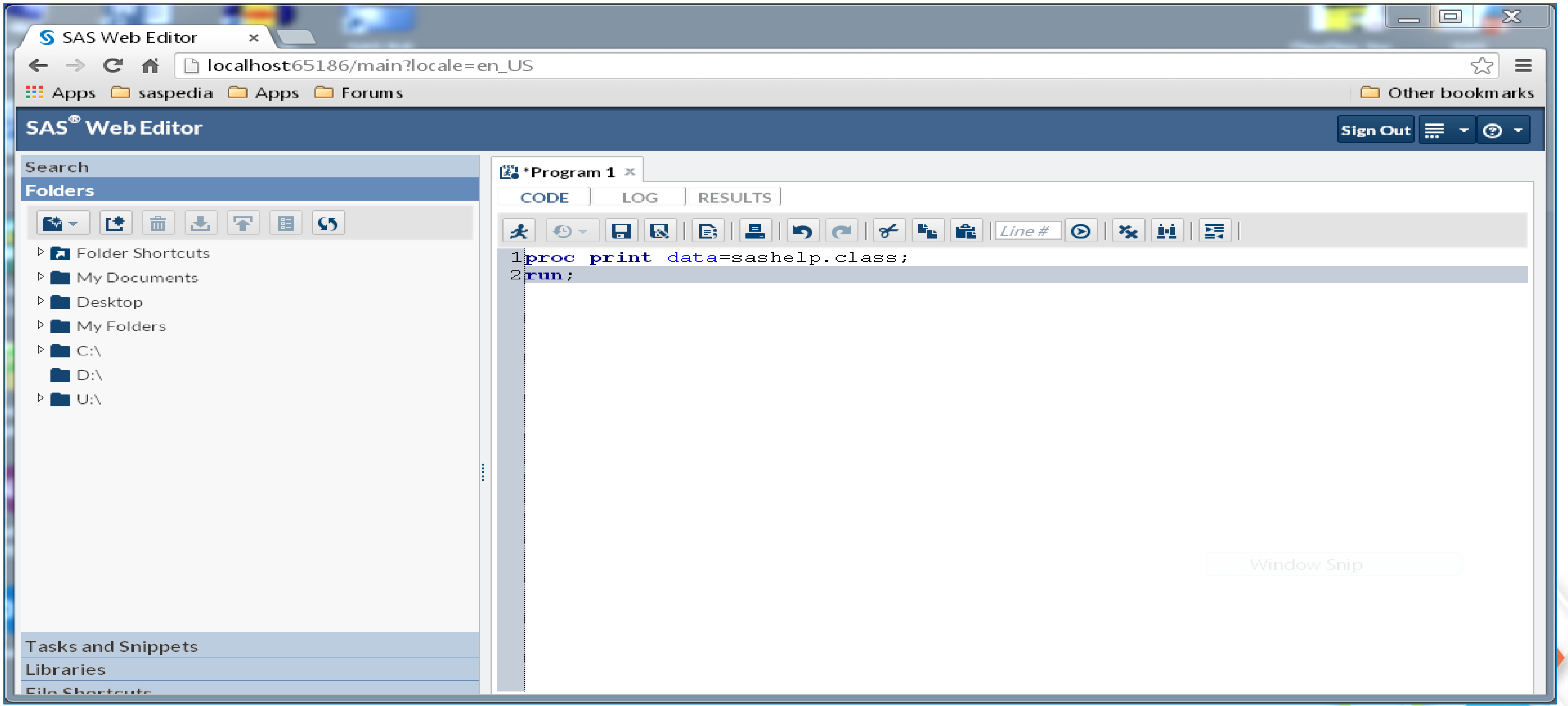
- **Consistent** - Learn one interface that you can use throughout your career - as a student, an individual SAS user/consultant, a departmental user, and an enterprise user.
- **Available** - You can use the same interface wherever you need it – my Mac in my dorm, my Windows desktop at work, my laptop at home, and my iPad on the road.
- **Assistive** - For programmers, the code is front & center, but you can use EG-like functions such as code-generating tasks or process flows to help if you need them.

SAS Studio

- You can access your programs, data files, and libraries.
- When you run a program or task, SAS Studio connects to a SAS server in order to process the generated SAS code.
- The SAS server can be a hosted server in a cloud environment, or it can be a server in your local environment.
- After the code is processed, the program and task results are returned to SAS Studio.



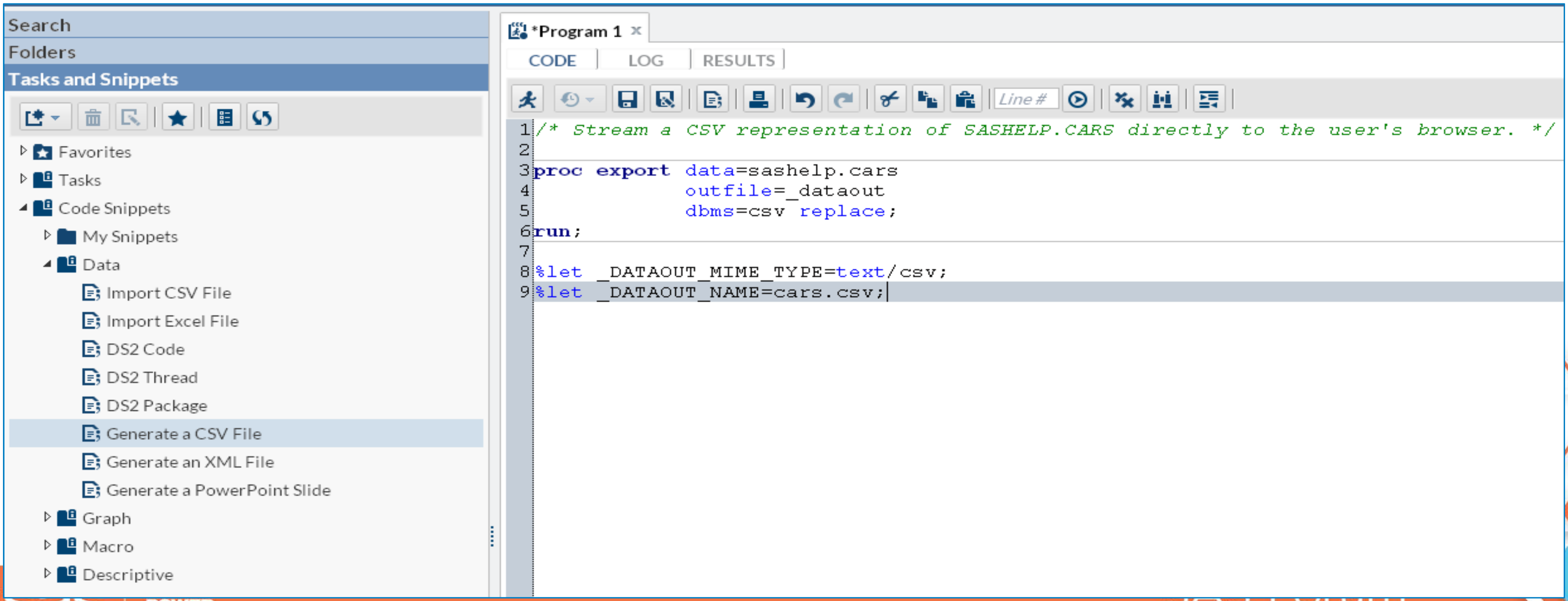
SAS Studio



```
proc print data=sashelp.class;
run;
```

Snippets

Code snippets enable you to quickly insert SAS code into your program and customize it to meet your needs. SAS Studio is shipped with several code snippets. You can also create your own snippets and create favorites.



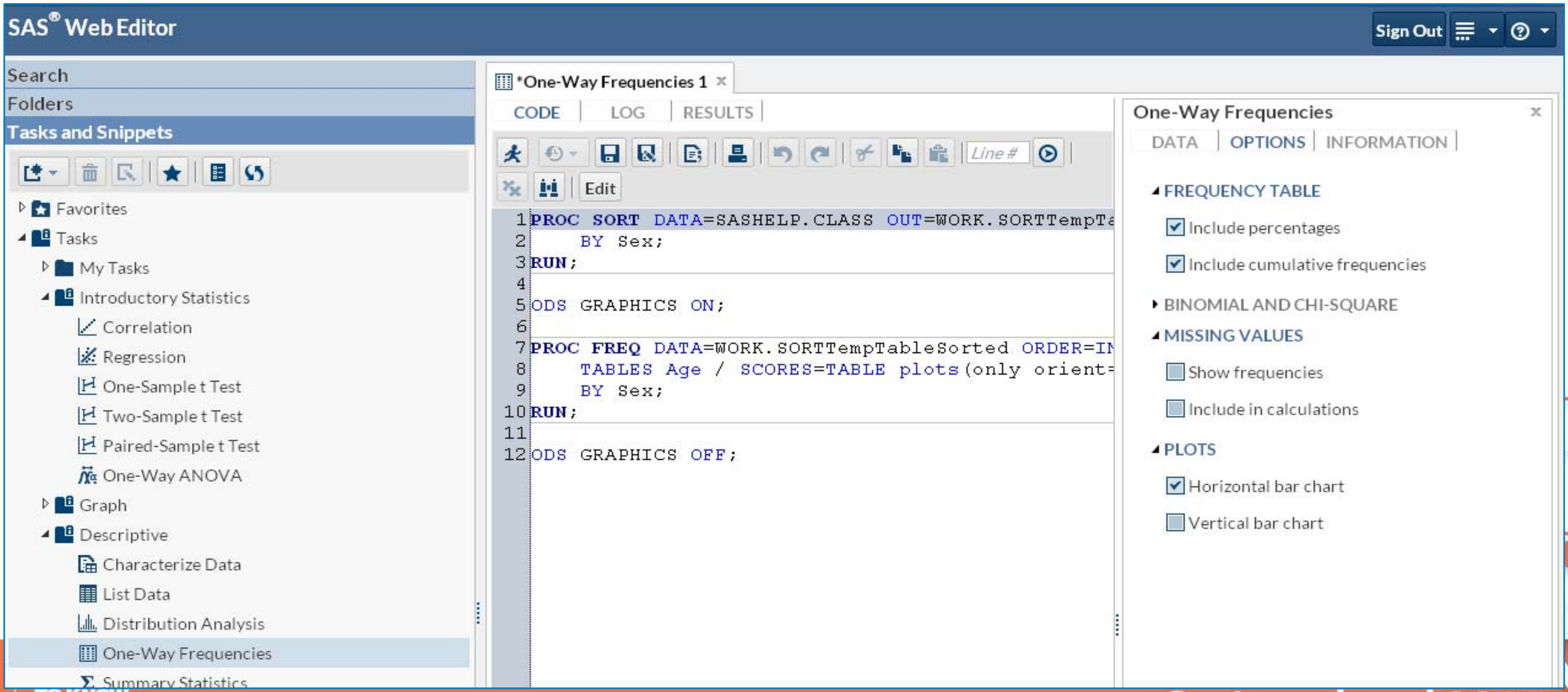
```
/* Stream a CSV representation of SASHELP.CARS directly to the user's browser. */

proc export data=sashelp.cars
            outfile=_dataout
            dbms=csv replace;
run;

%let  _DATAOUT_MIME_TYPE=text/csv;
%let  _DATAOUT_NAME=cars.csv;
```

Tasks

Tasks are point and click user interfaces which guide users through an analytical or other process. Behind the scenes, tasks generate SAS code which is run on a SAS server.



```
PROC SORT DATA=SASHELP.CLASS OUT=WORK.SORTTempTa
     BY Sex;
RUN;

ODS GRAPHICS ON;

PROC FREQ DATA=WORK.SORTTempTableSorted ORDER=IN
     TABLES Age / SCORES=TABLE plots(only orient=
     BY Sex;
RUN;

ODS GRAPHICS OFF;
```

FREQUENCY TABLE

☑ Include percentages

☑ Include cumulative frequencies

BINOMIAL AND CHI-SQUARE

MISSING VALUES

□ Show frequencies

□ Include in calculations

PLOTS

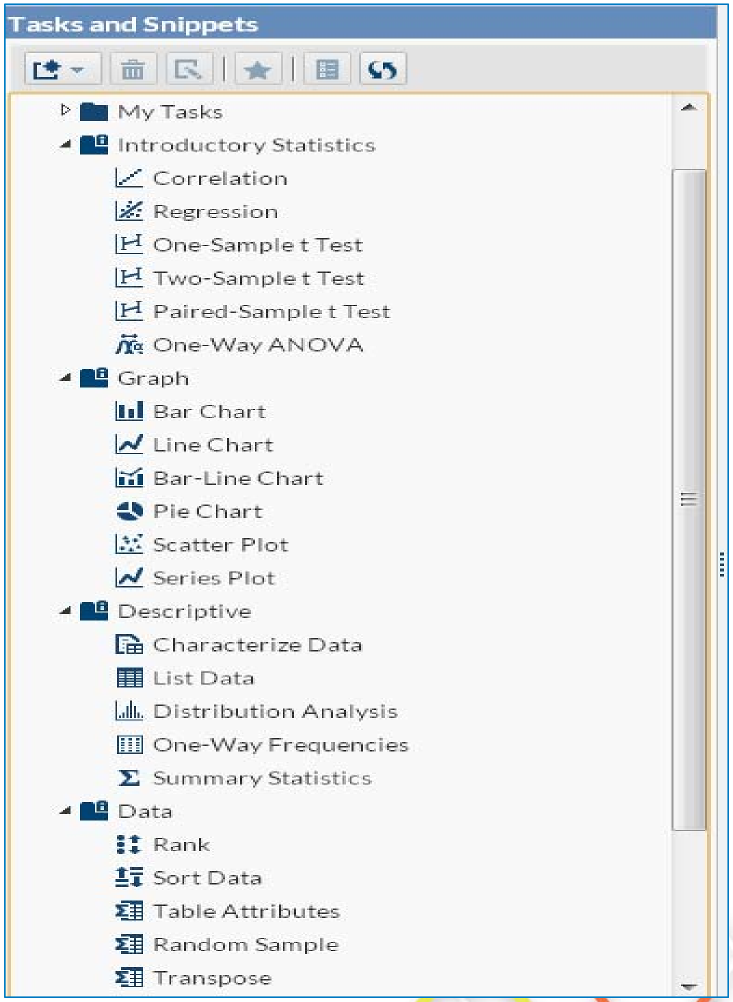
☑ Horizontal bar chart

□ Vertical bar chart

Tasks

SAS Studio allows for a flexible task framework where both internal and external customers can create tasks.

- SAS R&D groups can add tasks.
- Customers and consultants can easily create their own tasks without java coding or actionscript coding.
- Customers and consultants can copy delivered tasks and modify them.

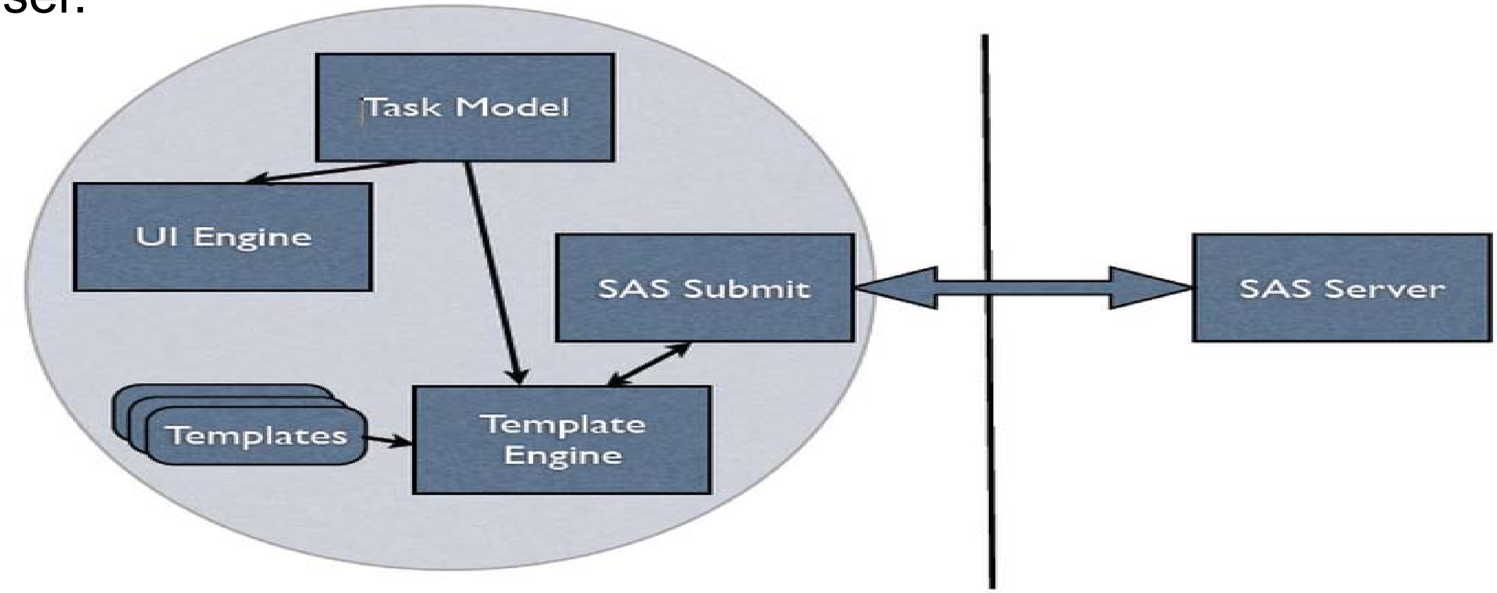


Tasks and Snippets

- My Tasks
- Introductory Statistics
  - Correlation
  - Regression
  - One-Sample t Test
  - Two-Sample t Test
  - Paired-Sample t Test
  - One-Way ANOVA
- Graph
  - Bar Chart
  - Line Chart
  - Bar-Line Chart
  - Pie Chart
  - Scatter Plot
  - Series Plot
- Descriptive
  - Characterize Data
  - List Data
  - Distribution Analysis
  - One-Way Frequencies
  - Summary Statistics
- Data
  - Rank
  - Sort Data
  - Table Attributes
  - Random Sample
  - Transpose

Tasks

The Common Task Model (CTM) defines customizable/parameterizable SAS execution tasks. A task is defined by describing the type of data it needs and the options it presents to the user.



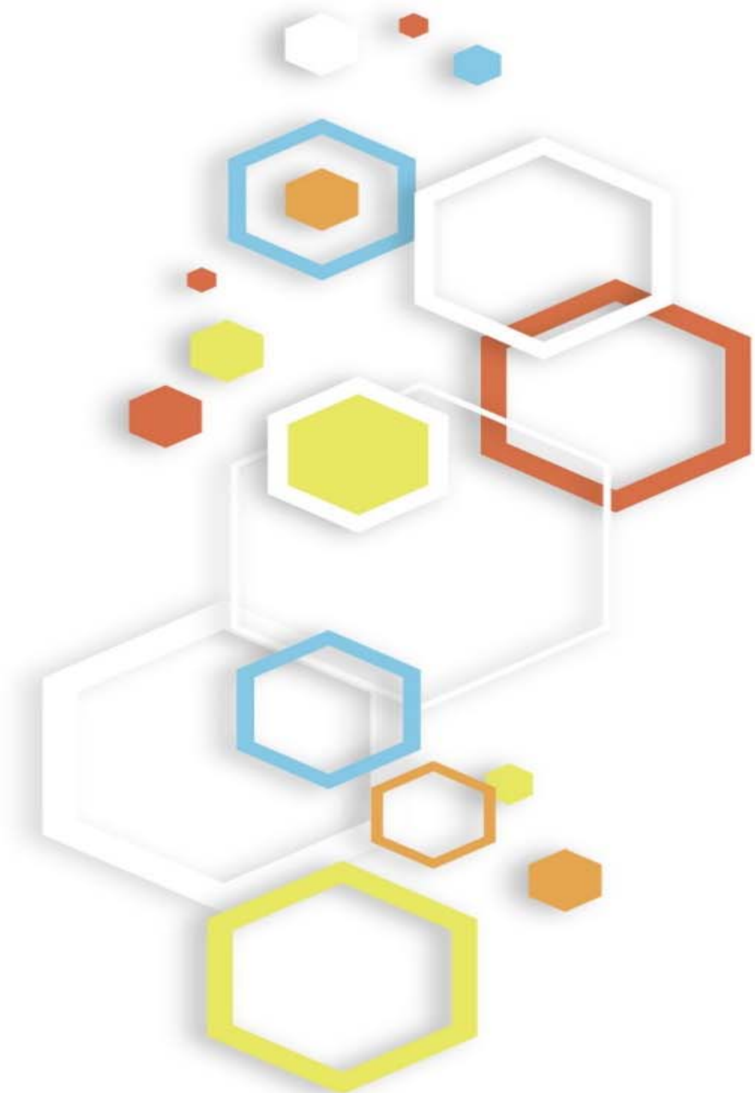
Perspectives

SAS Studio is intended to be more than just an application. It's a collection of common application elements, perspectives and views with a common look and feel.

A perspective is a personality for a given set of functionality. In some areas functionality will overlap - for example in the "SAS Programming Perspective" and the "Visual Programming Perspective". The objective of a perspective is to not exclusively own a given set of functionality but to provide a clean user experience tailored to a specific user type that leverages common elements.

SAS® Forum Switzerland 2014

make connections • share ideas • be inspired



Fragen?

Hans-Rainer Pauli
SAS Institute Switzerland

sas THE POWER TO KNOW.