



Adding User Interfaces to SAS Programs Using Basic Web Languages



Jules van der Zalm

Manager, OCS Life Sciences / PHUSE SDE 2021 / The Netherlands

User interfaces



- A visual system that allows the user to communicate with a computer (or its software), for example by entering parameters or making selections.



User interface

✖A



For the boundary between computer systems, see [Interface \(computing\)](#). For other uses, see [Interface](#).

In the [industrial design](#) field of [human–computer interaction](#), a **user interface (UI)** is the space where [interactions between humans and machines occur](#). The goal of this interaction is to allow effective





User interfaces

- A visual system that allows the user to communicate with a computer (or its software), for example by entering parameters or making selections.



user interface noun

 Save Word

plural **user interfaces**

Definition of *user interface*

: software that is designed to allow a computer user to interact with the operating system of a machine or system (such as by selecting presented options or entering text commands)



Create account

Your name

Mobile number or email

Password

[i](#) Passwords must be at least 6 characters.

Re-enter password

Continue

By creating an account, you agree to Amazon's
[Conditions of Use](#) and [Privacy Notice](#).

Already have an account? [Sign-In](#) ▶

Buying for work? [Create a free business account](#) ▶

User interface

- SAS/AF
- Been a
- Gets th
- Clunky
- PC-SAS

The screenshot shows a YouTube video player for a video titled "Modernising SAS AF" by Allan Bowe. The video content is a screen recording of the SAS/AF (SAS Analytics Framework) interface. The interface includes a "Components window" on the left, a central "Frame" area, a "Source window" on the right, and a "Properties window" at the bottom right. The video player interface shows the video is 11:29 long, with a progress bar at 0:02. Below the video, the title "Modernising SAS AF" is displayed, along with "31 views · Sep 15, 2021". There are 5 likes and 0 comments. A "SHARE" button and a "SAVE" button are also visible. A small inset video shows the presenter, Allan Bowe, in the bottom right corner of the video frame.



Use

- SA
- CT
- CT
- Int
- Sli
- ch
- Sp



SAS Studio

Not secure | ocsro.internal.ocs-consulting.com/SASStudio/main?locale=en_US&zone=GMT%252B02%253A00&http%3A%2F%2Focsro.internal.ocs-c...

SAS Programmer Sign Out

Server Files and Folders

- ocsnl-sas2
 - Folder Shortcuts
 - Files (P:\)

Tasks and Utilities

Snippets

Libraries

File Shortcuts

*SDTM Define-XML Development.ctl

Settings Code/Results Split

Log Code

OPTIONS INFORMATION

Locations:

- * Folder containing the SDTM datasets: P:\Projects\Compound A\Study X\Data
- * Folder to output Excel file: P:\Projects\Compound A\Study X\Output
- * Name to give to the Excel file: define-xml.xlsx
- XDF file containing the CRF annotations: C:\Users\jules.van.der.Zalm\Documents\define-xml.xdf

Required options:

- * SDTM version: 3.2
- * Controlled Terminology version (ISO date): 2021-06-22

Other options:

Tasks and Utilities

Snippets

Libraries

File Shortcuts

CODE LOG RESULTS

```

1 /*
2 *
3 * Task code generated by SAS Studio 3.8
4 *
5 * Generated on '9/19/21, 8:15 PM'
6 * Generated by 'jhz'
7 * Generated on server 'OCSNL-SAS-INTERNAL-OC-CONSULTING.COM'
8 * Generated on SAS platform 'X64_64282 WIN'
9 * Generated on SAS version '9.04.020708052020'
10 * Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
11 * Generated on web client 'http://ocsro.internal.ocs-consulting.com/SASStudio
12 */
13
14 /*-----
15 Create macro variables with the information given from the options.
16 -----*/
17 %global global_input_path global_output global_indef global_sdm_version
18 %global global_ct_version global_vin_filter global_gpi_version global_default_language
19 %global global_crf global_default_sdm global_debug;
20 %let global_input_path = P:\Projects\Compound A\Study X\Data;
21 %let global_output = P:\Projects\Compound A\Study X\Output\define-xml
22 %let global_indef = P:\Projects\Compound A\Study X\Output\define-xml
23 %let global_sdm_version = 3.2;
24 %let global_ct_version = 2021-06-22;
25 %let global_vin_filter = 1;
26 %let global_gpi_version = 3.0;
27 %let global_default_language = eng;
28 %let global_default_crf = Y;
29 %let global_default_sdm = Y;
30 %let global_debug = N;
31
32 /*-----
33 Include standard macros for the creation of SDTM Define-XML.
34 -----*/
35 %filename macrolib "P:\Projects\OCS Life Sciences\00002 - Standard macros\SDTM
36 %options mautosource sasautos=(sasautos macrolib);
37 %options compress=y;
38
39
40 /*-----
41 Run macro %create_metadata.
42 -----*/
43 %create_metadata_task;
  
```

P:\Projects\OCS Life Sciences\00002 - Standard macros\SDTM UTF-8

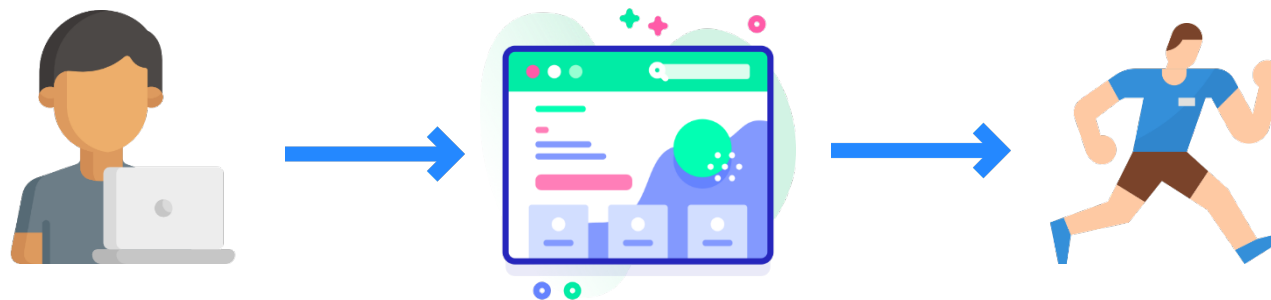
Messages User: Jules van der Zalm



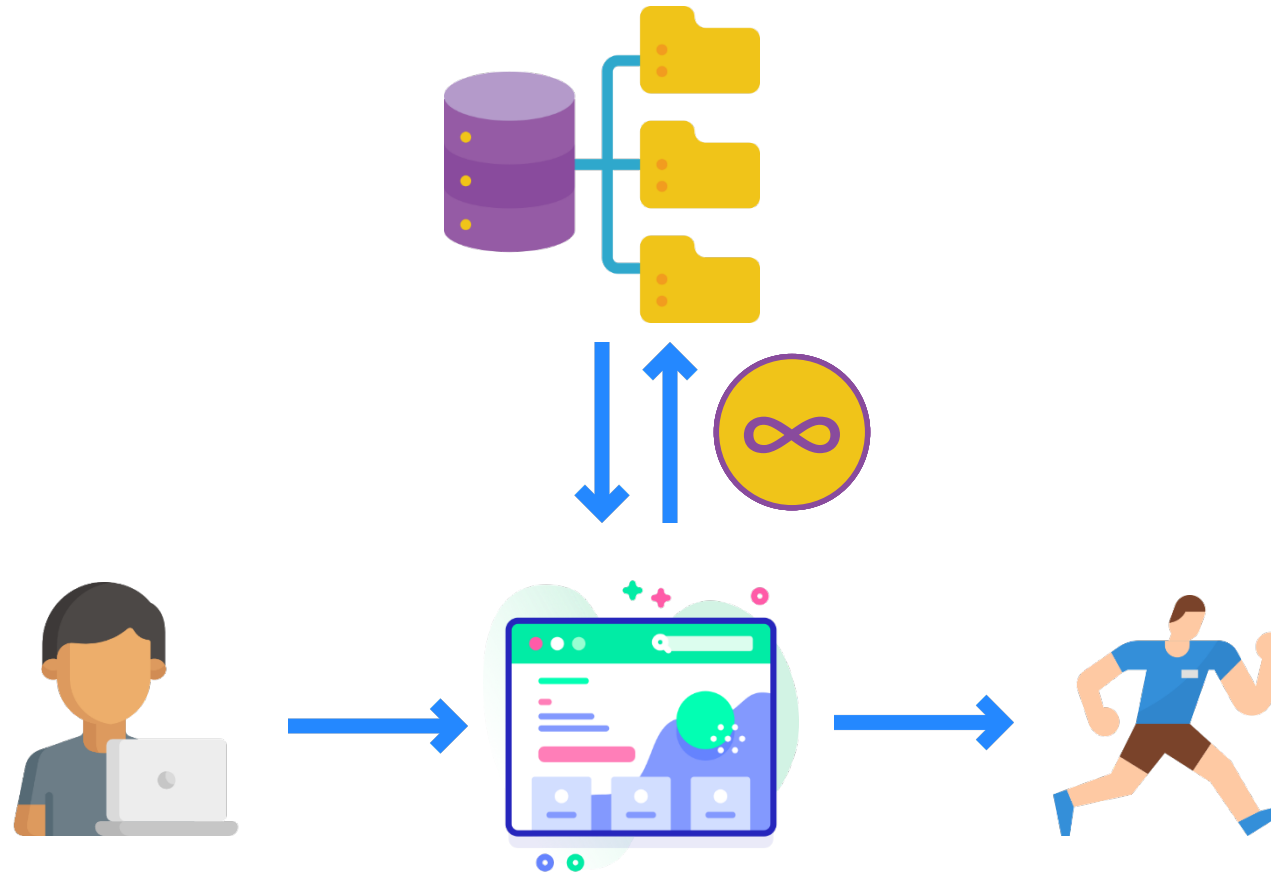
User interfaces in SAS

- Jobs in LSAF
- Integrated in SAS LSAF
- Great to standardise program execution
- Specific to SAS LSAF

Where they fall short



What they should do



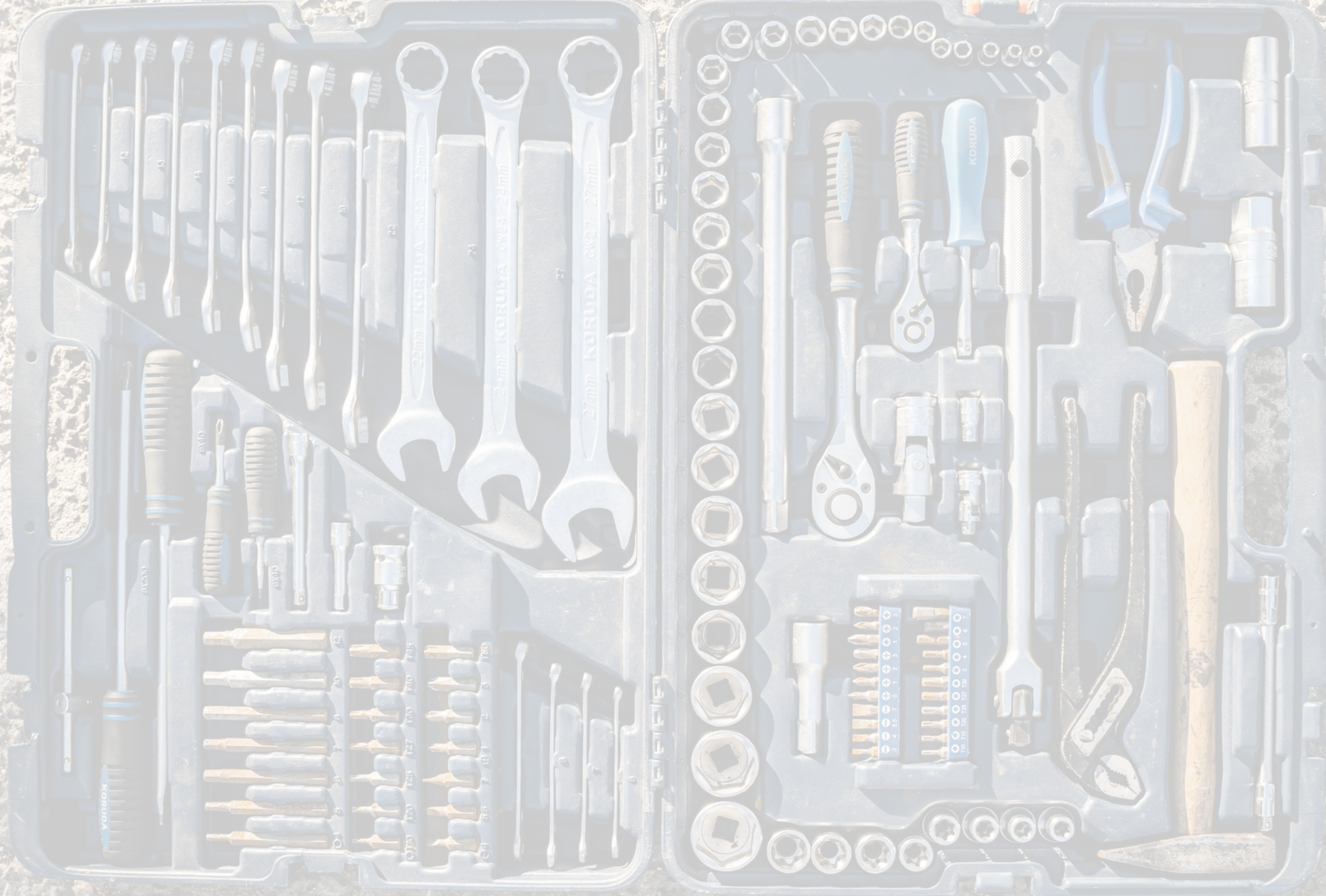
Our aim

- Build simple and easy-to-use user interfaces
- Use commonly available technology & knowledge





DISCLAIMER



PROGRAM CODE



**STORED
PROCESSES**



HTML



CSS



JS



BACK-END



PROGRAM CODE



**STORED
PROCESSES**

MIDDLEWARE



FRONT-END

HTML



CSS



Let me show you



- My user requirements:
 - See studies with (either few or many) adverse events
 - See overview of AEs in a study
 - See details of specific AEs

Let me show you



- The programming approach

```
1
2 %let studyid=STUDY005;
3 %let aeid =000014;
4
5
6 proc sql noprint;
7     create table work.studies as
8     select distinct(studyid) as studyid
9         , count(*) as numberofaes
10    from pooleddata.ae
11    group by studyid;
12 quit;
13
14 proc print data=pooleddata.ae (where = studyid = "&studyid.");
15     keep = usubjid aeseq aeterm aesev);
16 run;
17
18 proc print data=pooleddata.ae (where = aeid = "&aeid.");
19 run;
20
21
```

Let me show you

- The User Interface approach



**OCS' SASDEMO
ENVIRONMENT**



A breakdown





studies.sas7bdat



Apply filter to list of studies (optional)

List of studies

List of AEs in selected study

Details of selected AE

```
<table id="studiesTable" class="data">
</table>
```

```
function getStudiesData() {
  var _data = new FormData();

  value = document.getElementById("filterNumberOfAEs").value;
  if (value=="") {value = ">0";}

  _data.append("data2filter", "source.studies");
  _data.append("subjectswithaes", value);
  callStoredProcess(_data,"filterNumberOfAEs",true,"",_ready,true);

  function _ready(_JSONstring){
    console.log(_JSONstring);
    if (_JSONstring.length>3){studiesJSON = JSON.parse(_JSONstring);}
    console.log(studiesJSON);
    getStudiesMetadata();
  }
}
```

```
proc json out=_webout nosastags;
  export &dataset2JSON;
run;
```

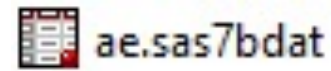
```
[{"s":fillTable("studiesTable",studiesJSON,,"numberofaes":3,"numberofsaes":0},
{"s":fillTable("studiesTable",studiesJSON,,"numberofaes":6,"numberofsaes":1},
{"s":fillTable("studiesTable",studiesJSON,,"numberofaes":1,"numberofsaes":0},
{"s":fillTable("studiesTable",studiesJSON,,"numberofaes":3,"numberofsaes":0},
{"s":fillTable("studiesTable",studiesJSON,,"numberofaes":6,"numberofsaes":0},
{"s":fillTable("studiesTable",studiesJSON,,"numberofaes":20,"numberofsaes":1},
```

Apply filter to list of studies (optional)

List of studies

List of AEs in selected study

Details of selected AE



```
var _data = new FormData();
_data.append("data2filter", "source.ae");
_data.append("studyid", studyid);
callStoredProcess(_data, "filterstudy", true, "", _ready, true);
```

```
data %sysfunc(scan("&data2filter",2,"."));
  set &data2filter;
  where studyid = "&studyid.";
run;

%let dataset2JSON=%sysfunc(scan("&data2filter",2,"."));
%include "&programroot.storedprocess\dataset2JSON.sas";
```

```
proc json out=_webout nosastags;
  export &dataset2JSON;
run;
```

```
function showAEs(){
  fillTable("AETable", AEJSON, ["id",
                                "STUDYID",
                                "USUBJID",
                                "AESEQ",
                                "AETERM",
                                "AESEV"],
    [], events_metadataJSON);
}

<table id="AETable" class="data">
</table>
```

Other examples



```
axis2 label = (height = 1.25 "&label_parameter") minor = none;

title "Boxplot X versus Y";
footnote;

proc boxplot data=work.adeff;
  plot aval*trtp / boxstyle=schematicid
                boxwidth=10
                idsymbol=circle
                vaxis=axis2;

  id usubjid;
run;
quit;
```

Other examples



```
/*  
  Code that produces the PDF goes here.  
*/  
  
data _null_;  
  file _webout;  
  line = "<a target=\"_blank\" href=\"&reporthttploc.\">Download as PDF</a>";  
  put line;  
run;
```

[Download as PDF](#)

So, what's my 2nd language?



- There is no single language that offers this all
- Languages and techniques combined here:
 - SAS
 - Base, SQL, Macro
 - Stored Processes (registered in SAS metadata)
 - HTML
 - CSS
 - JavaScript
 - Including AJAX (Asynchronous JavaScript and XML)
 - JSON



Lessons learned

- Hard to find programmers that combine all skills
- Those that do are usually not in-depth in all languages
- Could benefit further from web application frameworks (e.g. Angular)



Drawbacks

- Cannot be embedded in, for example, SAS Studio or LSAF
- Fairly steep learning curve for non-SAS programmers



Benefits

- Add parameters to programs in a safe and user-friendly manner
- Can run on the same server as your existing SAS environment (except LSAF)
- Beyond adding parameters, build applications
- No programming required for users
- Once components are there it's fast and only moderately difficult

Thank you! Time for questions



Jules van der Zalm



sasquestions@ocs-consulting.com

<https://www.linkedin.com/in/julesvanderzalm/>