SAS® FORUM UNITED KINGDOM 2017

Accelerate Innovation with Analytics and Experience Your New Possible

Copyright © SAS Institute Inc. All rights reserved.

SAS® FORUM
UNITED KINGDOM 2017

Easy AF Modernisation

with HTML5 and SAS

Copyright © SAS Institute Inc. All rights reserved.

SAS® FORUM
UNITED KINGDOM 2017

Nik Markovic

Setting the Scene

Copyright © SAS Institute Inc. All rights reserved.

SAS® FORUM
UNITED KINGDOM 2017

Michael Saunders

The Challenge Ahead

Copyright © SAS Institute Inc. All rights reserved.

DHOLAP

- Month-end & ad-hoc reporting
- 100+ users across PL, CL & Engineering
- Various KPI's, e.g.
  - Business volumes & premium income
  - Date basis – underwriting, financial or loss
  - Loss ratios (claims incurred/GEP)
- Development triangles

boemska

Copyright © SAS Institute Inc. All rights reserved.

DHOLAP – Development Triangles

Copyright © SAS Institute Inc. All rights reserved.

SLIDES REMOVED

Copyright © SAS Institute Inc. All rights reserved.

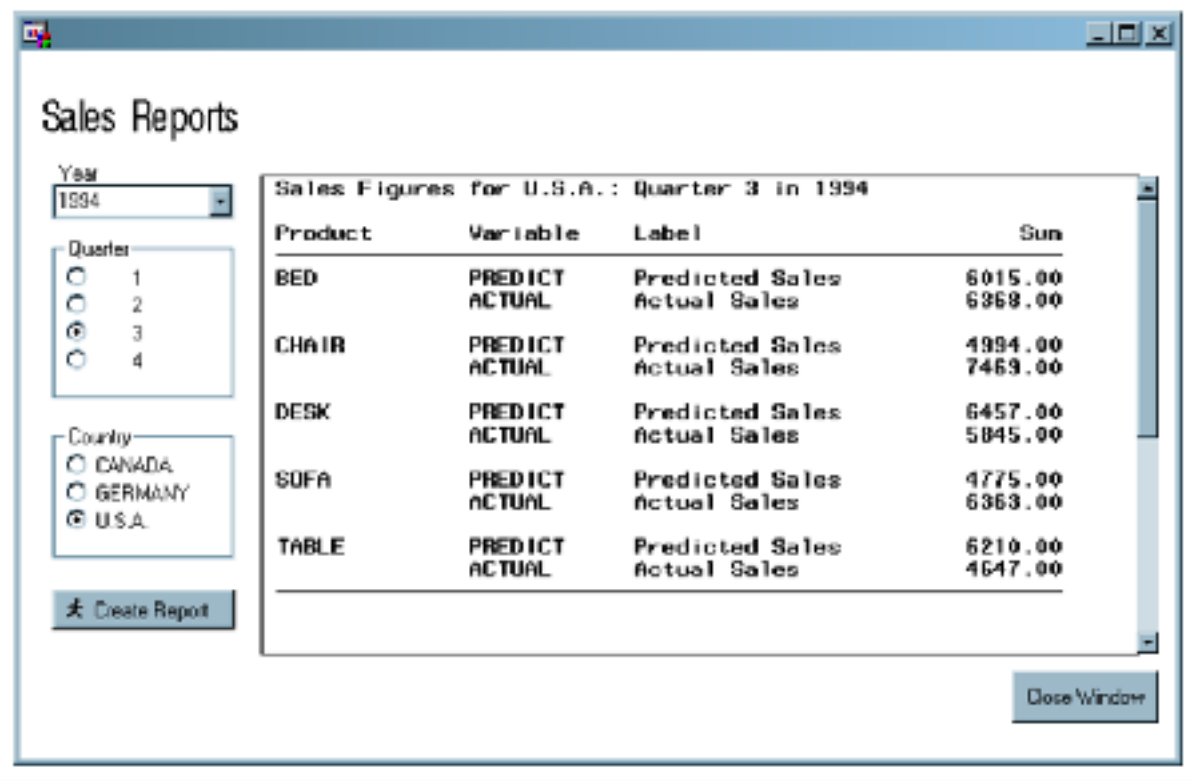
SAS® FORUM
UNITED KINGDOM 2017

Allan Bowe

How We Did It

Copyright © SAS Institute Inc. All rights reserved.

SAS/AF & SCL

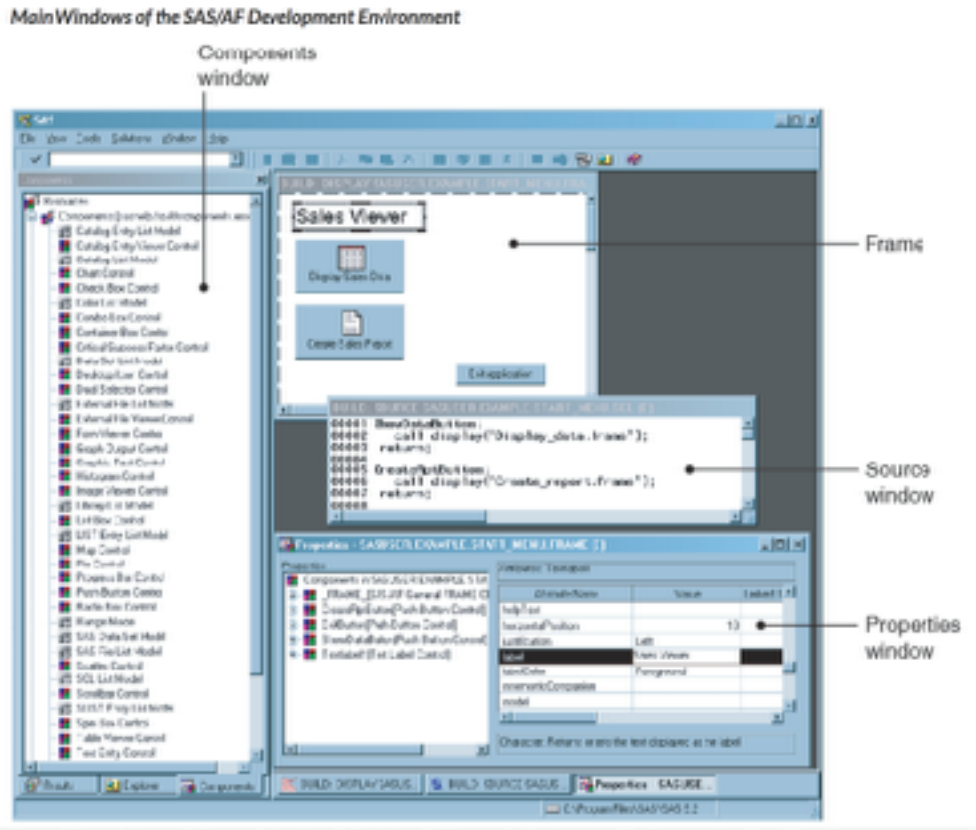


boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.

SAS/AF & SCL



Copyright © SAS Institute Inc. All rights reserved.

Drawbacks

- Security
- Scalability
- Support
- Cost

boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.

Benefits

- Security
- Scalability
- Support
- Cost

boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.

Easy AF – Steps for Success

- Front End
- Back End
- API

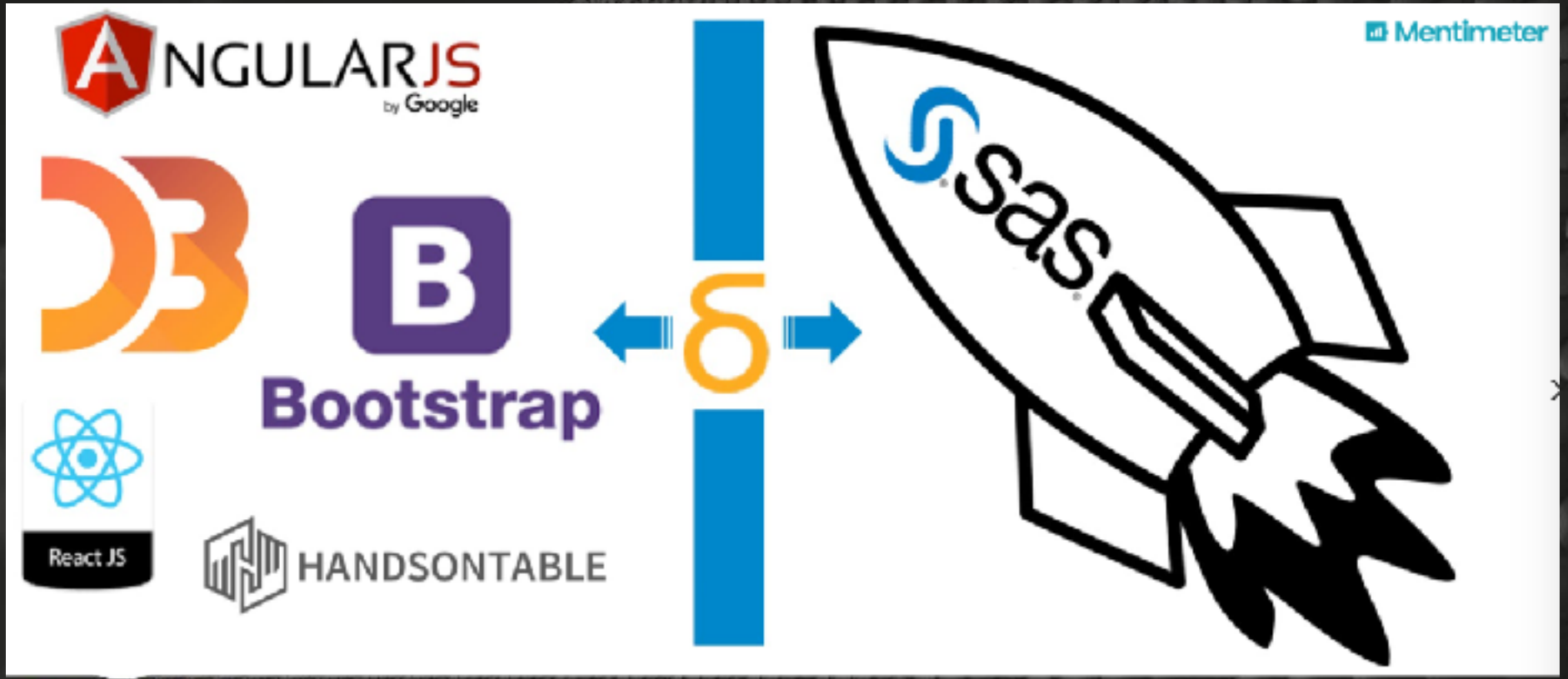
boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.

AF

SCL



boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.

Easy AF – Front End

- Screenshots
- Screen recordings
- Wireframes

boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.

Easy AF – Backend

- Extract source entries
- Extract slists
- Prepare services

boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.

SCL Functions

```
varname   = nameitem(l_types,i);
type_no   = getitemn(l_types,i);
l_typvar  = insertc(l_typvar,varname,-1);
l_typnum  = insertn(l_typnum,type_no,-1);
```

boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.

SCL Functions

```
varname   = nameitem(l_types,i);
type_no   = getitemn(l_types,i);
l_typvar  = insertc(l_typvar,varname,-1);
l_typnum  = insertn(l_typnum,type_no,-1);
```

proc fcmp

```
varname   = nameitem(l_types,i);
type_no   = getitemn(l_types,i);
l_typvar  = insertc(l_typvar,varname,-1,'');
l_typnum  = insertn(l_typnum,type_no,-1,'');
```

Copyright © SAS Institute Inc. All rights reserved.

SUBMIT BLOCKS

```
submit;
    data _trans4_(drop=i) ;
        set &i_dsname (keep=
endsubmit;
```

boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.

SUBMIT BLOCKS

```
submit;
    data _trans4_(drop=i) ;
        set &i_dsname (keep=
endsubmit;
```

```
rc=bscl_submit(bafref,'
    data _trans4_(drop=i) ;
        set '!!i_dsname!!' (keep=');
```

boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.

OBJECT METHODS

```
do i = 1 to numanl;

    varname = nameitem(l_analys,i);
    MetaDataID.GetDataSetVariableAttribute(i_dsname,varname,"VARTYP",vartyp);
    MetaDataID.GetDataSetVariableAttribute(i_dsname,varname,"VARLEN",varlen);

    if vartyp = 'C' then varlenx = max(varlenx,varlen);

end;
```

Copyright © SAS Institute Inc. All rights reserved.

OBJECT METHODS

```
do i = 1 to numanl;

    varname = nameitem(l_analys,i);
    MetaDataID.GetDataSetVariableAttribute(i_dsname,varname,"VARTYP",vartyp);
    MetaDataID.GetDataSetVariableAttribute(i_dsname,varname,"VARLEN",varlen);

    if vartyp = 'C' then varlenx = max(varlenx,varlen);

end;
```

```
do i = 1 to numanl;

    varname = nameitem(l_analys,i);
    %baf_class(class=objects.dhobject.metadata
              ,method=GetDataSetVariableAttribute
              ,varlist='i_dsname,varname, "VARTYP" ,vartyp ');
    %baf_class(class=objects.dhobject.metadata
              ,method=GetDataSetVariableAttribute
              ,varlist='i_dsname,varname,"VARLEN",varlen ');
    if vartyp = 'C' then varlenx = max(varlenx,varlen);

end;
```

boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.

METHOD BLOCKS

```
5953  TYPES: method optional = StatType $ ;
5954
5955      AuditID    = _self_.OwnerID.AuditID;
5956      AuditID.LogFunctionCall('DAENGINE','SubmitTypes');
5957
5958      MetaDataID = _self_.OwnerID.MetaDataID;
5959      l_define   = MetaDataID.DefineID;
5960      l_typinf   = getniteml(l_define,'_TYPE_ INFO');
5961      l_typnum   = getniteml(l_typinf,'REQUIRED _TYPES_');
5962      l_rclcmb   = getniteml(l_typinf,'REQUIRED CLASS COMBINATIONS');
```

boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.

METHOD BLOCKS

```
5953  TYPES: method optional = StatType $ ;
5954
5955      AuditID     = _self_.OwnerID.AuditID;
5956      AuditID.LogFunctionCall('DAENGINE','SubmitTypes');
5957
5958      MetaDataID = _self_.OwnerID.MetaDataID;
5959      l_define   = MetaDataID.DefineID;
5960      l_typ
5961      l_typ
5962      l_rcl
```

```
function bscl_class(catalog $,method $,varlist $, src, bafmsig $);
  rc=bafhashexport();
  length baf_sysmsg $1000;
  rc=run_macro('bscl_class',catalog,method,varlist,src,listid,bafms
  file log; put 'bscl_class (out):' catalog= method= varlist= listi
  return (listid);
endsub;
```

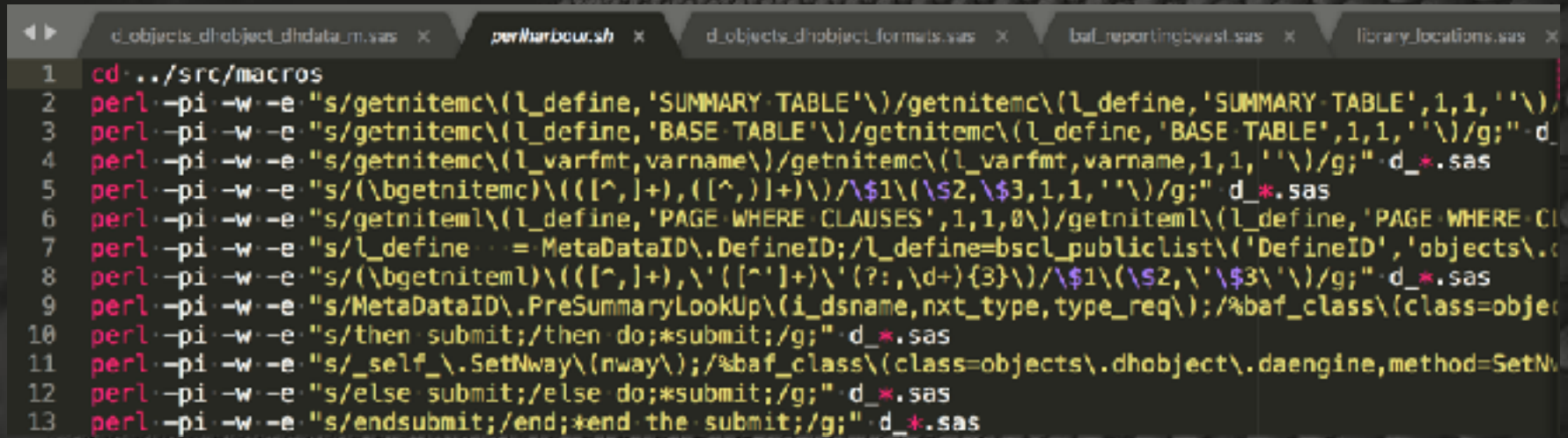
boemska

Copyright © SAS Institute Inc. All rights reserved.

sas Alliance Silver Member

```
%else %if &swltch=DTCUTOFF %then %do; /* GetDateCutOff */
  * DTCUTOFF: method cutdate 8 optional = whr_mnth out_mnth $200 ;
  length cutdate 8 whr_mnth out_mnth $200;
  %baf_assignvars(varlist=&attr[&varlist],methodvars=&methodvars,src=&src);

   * l_define = _self_.DefineID;
   l_define=bsql_publiclist["DefineID","objects.dbobject.metadata"];
   cutdate  = .   ;
   whr_mnth = ''  ;
   out_mnth = ''  ;

   if nameditem(l_define,'DATECARD') then do;

      dcard=getnitemc(l_define,'DATECARD',1,1,'');

      * submit remote commns;
      rc=doexbl("
         filename datecard '"||trim(dcard)||"';
         data _null_;
            infile datecard;
            input date $8.;
            newdate=mdy(substr(date,3,2),substr(date,1,2)
                       ,substr(date,5,4))+1;
            call symput('datecard',newdate,'c');
         run;
         filename datecard;
         %sysrput datecard=&datecard;");
      endsubmit;

      cutdate=input(symget('datecard'),8.);

   end;

   else cutdate=today();

   pos      = nameditem(l_define,'STATISTIC DATE VARIABLES');

   if pos then do; /* If date variables have been registered then check fo
                   /* date cut-off processing

      l_across  = getnitemi(l_define,'ACROSS');
      l_down    = getnitemi(l_define,'DOWN');
      lastmnth  = 'N' ;

      * _self_.GetDateVariables(year_var,qtr_var,mnth_var);
     %baf_class(class=objects.dbobject.metadata
               ,method=GetDateVariables
               ,varlist='year_var,qtr_var,mnth_var ');

      /* Get cut-off at last month indicator */
      lastmnth  = getnitemc(l_define,'UP TO LAST MONTH ONLY',1,1,'N');

      /* Get last month cut-off offset if it exists - else it is 0 */
      lmoffset  = getnitemn(l_define,'MONTH CUT-OFF OFFSET',1,1,0);
```

```
1140 DTCUTOFF: method cutdate 8 optional = whr_mnth out_mnth $200 ;
1141
1142    l_define = _self_.DefineID;
1143
1144    cutdate  = .   ;
1145    whr_mnth = ''  ;
1146    out_mnth = ''  ;
1147
1148    if nameditem(l_define,'DATECARD') then do;
1149
1150       dcard=getnitemc(l_define,'DATECARD');
1151
1152       submit remote commns;
1153
1154          filename datecard '&dcard';
1155          data _null_;
1156             infile datecard;
1157             input date $8.;
1158             newdate=mdy(substr(date,3,2),substr(date,1,2)
1159                        ,substr(date,5,4))+1;
1160             call symput('datecard',newdate);
1161          run;
1162          filename datecard;
1163          %sysrput datecard=&datecard;
1164       endsubmit;
1165
1166       cutdate=input(symget('datecard'),8.);
1167
1168    end;
1169
1170    else cutdate=today();
1171
1172    pos      = nameditem(l_define,'STATISTIC DATE VARIABLES');
1173
1174    if pos then do; /* If date variables have been registered then check for */
1175                    /* date cut-off processing                                */
1176
1177       l_across  = getnitemi(l_define,'ACROSS');
1178       l_down    = getnitemi(l_define,'DOWN');
1179       lastmnth  = 'N' ;
1180
1181       _self_.GetDateVariables(year_var,qtr_var,mnth_var);
1182
1183
1184
1185       /* Get cut-off at last month indicator */
1186       lastmnth  = getnitemc(l_define,'UP TO LAST MONTH ONLY',1,1,'N');
1187
1188       /* Get last month cut-off offset if it exists - else it is 0 */
1189       lmoffset  = getnitemn(l_define,'MONTH CUT-OFF OFFSET',1,1,0);
```



boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.

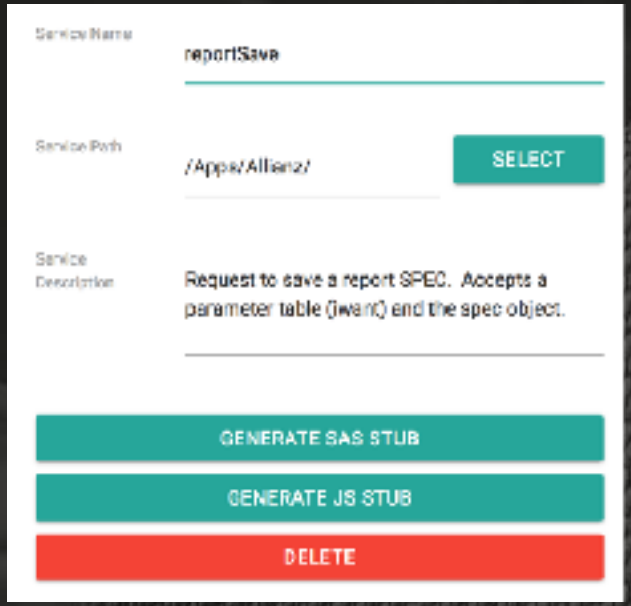
Easy AF – API

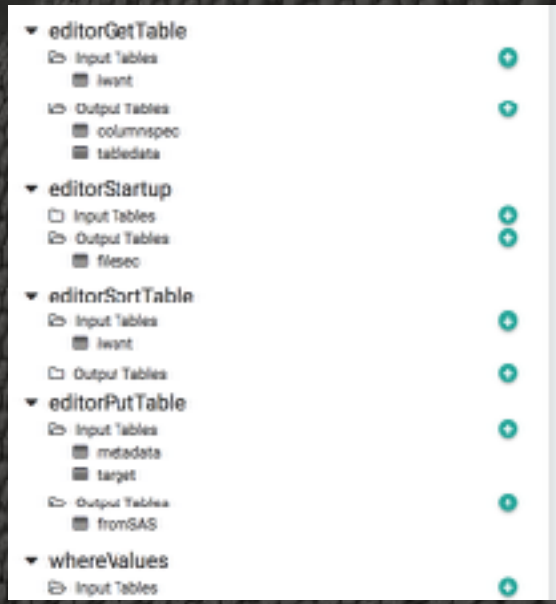
- STP location
- Inputs / Outputs
- Permissions

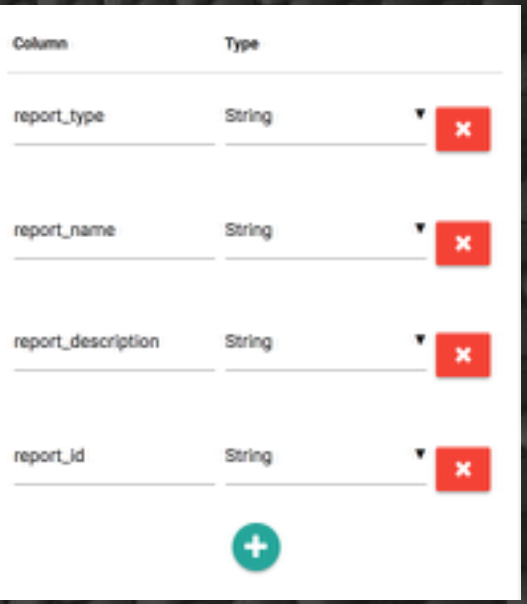
boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.







	report_type	report_name	report_description	report_id
1	GROUP	af.dholap.somelist	This is my extra well good report of lore	a;lksdfjp434309asdf
2	PRIVATE	af.dholap.somelist	My private report	gfsdgsvfertd5seggg%
3	PUBLIC	af.dholap.somelist	Public Facing - call this in usual way (not via reportSelect)	af.dholap.somelist
4				

boemska

sas Alliance Silver Member

Copyright © SAS Institute Inc. All rights reserved.

Generated SAS

```
1  /*--------reportSave--------*/
2
3  * load h54s from location variable ;
4  %include "%H54SLOC.";
5
6  * parse the input tables ;
7  %hfsGetDataset(imart,work.imart);    %hfsGetDataset(spec,work.spec);
8  %hfsErrorCheck;
9
10 /* SAS Code Goes Here */
11
12 data report_list;
13     report_type = 'GROUP';
14     report_name = 'af.drolap.samelist';
15     report_description = 'This is my extra well good report of lore';
16     report_id = 'a;lksdfjp434309asdf';
17     output;
18     * end row for loop;
19     report_type = 'PRIVATE';
20     report_name = 'af.drolap.samelist';
21     report_description = 'My private report';
22     report_id = 'gfsdgsvfsrtd5seggg%';
23     output;
24     * end row for loop;
25     report_type = 'PUBLIC';
```

reportSave.sas

CLOSE SAVE

Copyright © SAS Institute Inc. All rights reserved.

Generated JavaScript

```
var adapter = new h54s({/* options */});

var tables = new h54s.Tables([
  {
    'report_type': "GROUP/PRIVATE',
    'report_name': "af.sholap.somelist",
    'report_description': "This is my extra well good report of lore",
    'group_report_parms': "CLSGROUP.REPORT5'
  }
], 'iwant');

tables.add([], 'spec');

adapter.call('undefined/Apps/Allianz/reportSave', tables, function(err, res) {
  if(err) {
    alert(err.message);
    return;
  }

  console.log(res.report_list);

});
```

reportSave.js

CLOSE SAVE

Copyright © SAS Institute Inc. All rights reserved.

SLIDES REMOVED

Copyright © SAS Institute Inc. All rights reserved.

SAS® FORUM
UNITED KINGDOM 2017

www.sas.com/uk/events

Copyright © SAS Institute Inc. All rights reserved.