# ORION: A Non-Server-Based Interactive SAS Report Builder

Authors: Bruce Nawrocki, Scott Proescholdbell and Shana Geary

**Background**

The North Carolina (NC) Injury and Violence Prevention Branch of the Division of Public Health is tasked with reporting injuries and drug overdoses gathered from statewide Death Certificates, Hospitalization Discharges and Emergency Department visits. The SAS code to determine injury categories (mechanisms and intents) can be tricky to write, and our staff – which consists of some people in the midst of their advanced university degree programs or perhaps an early career epidemiologist – may not be well-versed in SAS or our data table structure and its nuances. This led to lengthy responses to both internal and external data requests, as new staff worked through the proper SAS program logic and syntax, which potentially led to inconsistent or incorrect reports due to user error.

**Implementation**

We decided to build some standard SAS report templates, shielding staff from the complexities of the SAS logic involved. We also built a User Interface to gather the report-writer's requirements, which were then fed into the SAS code as parameters. The SAS code creates fully formatted reports in either PDF or Excel files, which are saved within specific sub-folders for that staff-member and reporting date. The reports contain all that is necessary (NC state logos, footnotes, data suppression rules) for both internal and external data requesters. We named the system ORION - **O**n-demand **R**eporting of **I**njury and **O**verdose in **N**orth Carolina.

**Results**

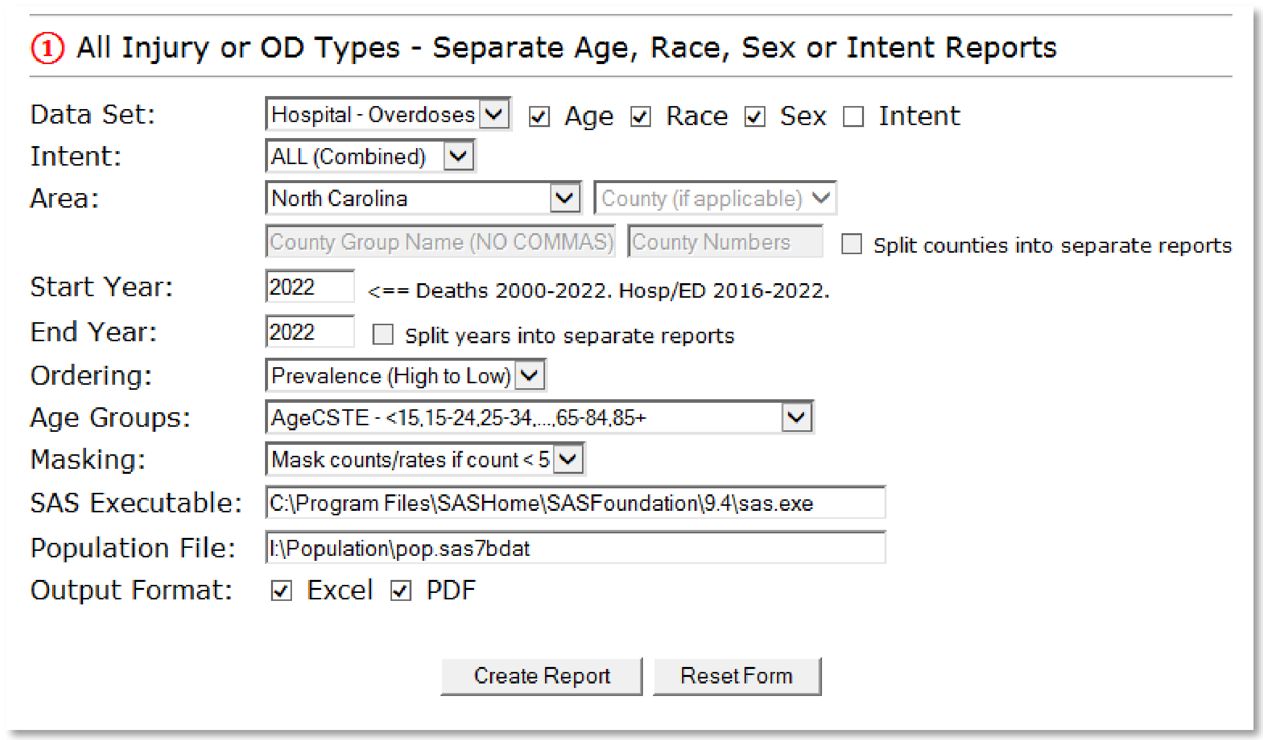
With ORION, we have greatly enhanced our core reporting process. We now provide more consistent reporting (and more quickly) to our data requestors and for internal data needs. New staff members require less training to become productive. Currently, ORION submits SAS batch programs that run on the staff's own local Windows computers, but we also have a prototype process that submits the SAS programs to run on a remote SAS Server instead.

ORION's user interface has evolved over time. It can now:

- Display a message that the SAS program is running, and determine when it has finished.
- Scan the SAS log, looking for occurrences of "Error" or "uninitialized". If found, it opens Notepad to display the log file.
- Save a person's most recent SAS code and SAS log, which can be useful for debugging purposes.
- Save each person's reports under the "Output" folder on our network Windows drive, within sub-folders named with that person's unique network userID, then saved into sub-folders named with date the report was run.

## Our User Interface

When someone runs the ORION.HTA file (by double-clicking it from Windows Explorer), this screen appears:

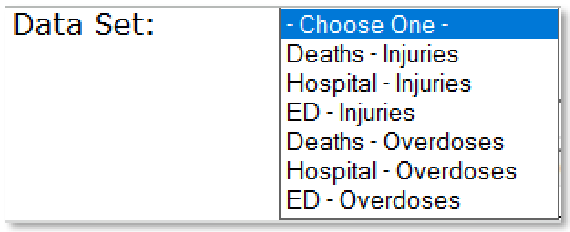


① All Injury or OD Types - Separate Age, Race, Sex or Intent Reports

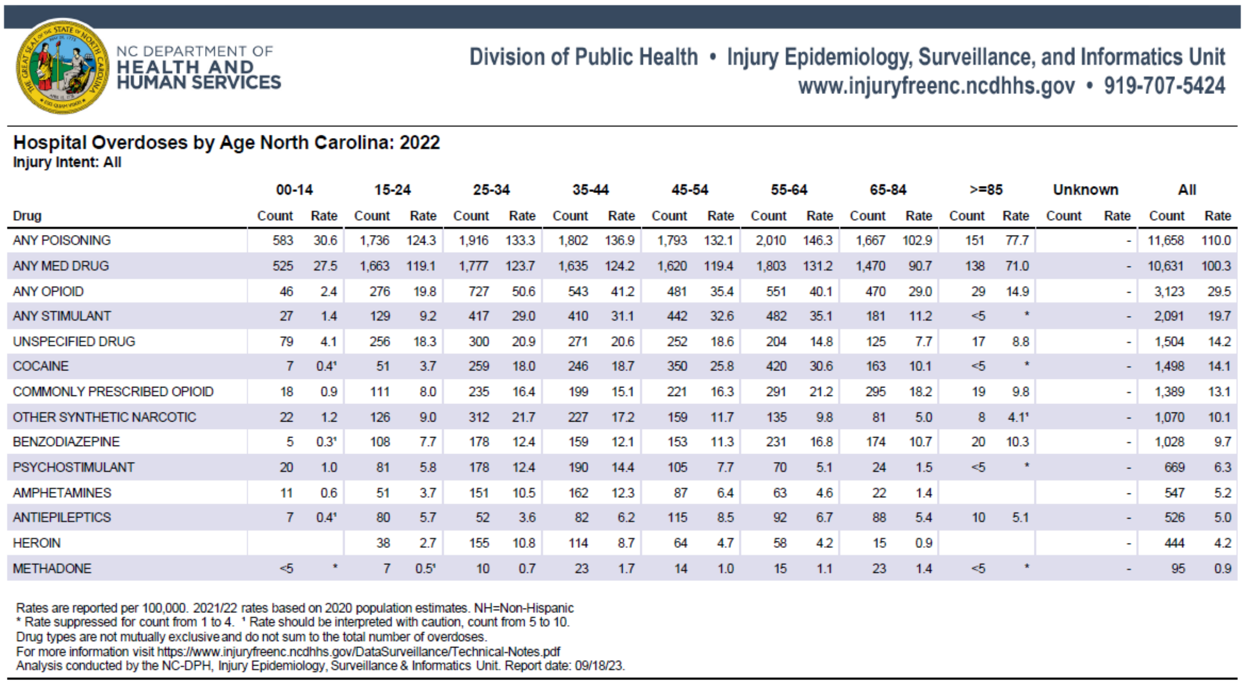
| | |
|---|---|
| Data Set: | Hospital - Overdoses ☑ Age ☑ Race ☑ Sex ☐ Intent |
| Intent: | ALL (Combined) |
| Area: | North Carolina; County (if applicable) |
| | County Group Name (NO COMMAS); County Numbers; ☐ Split counties into separate reports |
| Start Year: | 2022 <== Deaths 2000-2022. Hosp/ED 2016-2022. |
| End Year: | 2022 ☐ Split years into separate reports |
| Ordering: | Prevalence (High to Low) |
| Age Groups: | AgeCSTE - <15,15-24,25-34,...,65-84,85+ |
| Masking: | Mask counts/rates if count < 5 |
| SAS Executable: | C:\Program Files\SASHome\SASFoundation\9.4\sas.exe |
| Population File: | I:\Population\pop.sas7bdat |
| Output Format: | ☑ Excel ☑ PDF |

They can choose from one of our multiple core injury datasets – Deaths, Hospitalizations or Emergency Department Visits, statewide or for specific counties, by year. Output can be sent to Excel and/or PDF files. There's a second report parameter screen (not shown) that optionally combines the core injury datasets into one report.

In the above screen-shot, a person has selected Data Set = "Hospital – Overdoses", but they can choose from these other Data Sets:



After selecting the choices above, three Hospital Overdose reports are created – by Age-Group, Race and Sex – both in Excel and PDF formats, since both checkboxes have checkmarks. An example of the Age-Group PDF report that gets created is:



NC DEPARTMENT OF HEALTH AND HUMAN SERVICES

Division of Public Health • Injury Epidemiology, Surveillance, and Informatics Unit
www.injuryfreenc.ncdhhs.gov • 919-707-5424

**Hospital Overdoses by Age North Carolina: 2022**
**Injury Intent: All**

| | 00-14 | | 15-24 | | 25-34 | | 35-44 | | 45-54 | | 55-64 | | 65-84 | | >=85 | | Unknown | | All | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Drug | Count | Rate | Count | Rate | Count | Rate | Count | Rate | Count | Rate | Count | Rate | Count | Rate | Count | Rate | Count | Rate | Count | Rate |
| ANY POISONING | 583 | 30.6 | 1,736 | 124.3 | 1,916 | 133.3 | 1,802 | 136.9 | 1,793 | 132.1 | 2,010 | 146.3 | 1,667 | 102.9 | 151 | 77.7 | | - | 11,658 | 110.0 |
| ANY MED DRUG | 525 | 27.5 | 1,663 | 119.1 | 1,777 | 123.7 | 1,635 | 124.2 | 1,620 | 119.4 | 1,803 | 131.2 | 1,470 | 90.7 | 138 | 71.0 | | - | 10,631 | 100.3 |
| ANY OPIOID | 46 | 2.4 | 276 | 19.8 | 727 | 50.6 | 543 | 41.2 | 481 | 35.4 | 551 | 40.1 | 470 | 29.0 | 29 | 14.9 | | - | 3,123 | 29.5 |
| ANY STIMULANT | 27 | 1.4 | 129 | 9.2 | 417 | 29.0 | 410 | 31.1 | 442 | 32.6 | 482 | 35.1 | 181 | 11.2 | <5 | * | | - | 2,091 | 19.7 |
| UNSPECIFIED DRUG | 79 | 4.1 | 256 | 18.3 | 300 | 20.9 | 271 | 20.6 | 252 | 18.6 | 204 | 14.8 | 125 | 7.7 | 17 | 8.8 | | - | 1,504 | 14.2 |
| COCAINE | 7 | 0.4[1] | 51 | 3.7 | 259 | 18.0 | 246 | 18.7 | 350 | 25.8 | 420 | 30.6 | 163 | 10.1 | <5 | * | | - | 1,498 | 14.1 |
| COMMONLY PRESCRIBED OPIOID | 18 | 0.9 | 111 | 8.0 | 235 | 16.4 | 199 | 15.1 | 221 | 16.3 | 291 | 21.2 | 295 | 18.2 | 19 | 9.8 | | - | 1,389 | 13.1 |
| OTHER SYNTHETIC NARCOTIC | 22 | 1.2 | 126 | 9.0 | 312 | 21.7 | 227 | 17.2 | 159 | 11.7 | 135 | 9.8 | 81 | 5.0 | 8 | 4.1[1] | | - | 1,070 | 10.1 |
| BENZODIAZEPINE | 5 | 0.3[1] | 108 | 7.7 | 178 | 12.4 | 159 | 12.1 | 153 | 11.3 | 231 | 16.8 | 174 | 10.7 | 20 | 10.3 | | - | 1,028 | 9.7 |
| PSYCHOSTIMULANT | 20 | 1.0 | 81 | 5.8 | 178 | 12.4 | 190 | 14.4 | 105 | 7.7 | 70 | 5.1 | 24 | 1.5 | <5 | * | | - | 669 | 6.3 |
| AMPHETAMINES | 11 | 0.6 | 51 | 3.7 | 151 | 10.5 | 162 | 12.3 | 87 | 6.4 | 63 | 4.6 | 22 | 1.4 | | | | - | 547 | 5.2 |
| ANTIEPILEPTICS | 7 | 0.4[1] | 80 | 5.7 | 52 | 3.6 | 82 | 6.2 | 115 | 8.5 | 92 | 6.7 | 88 | 5.4 | 10 | 5.1 | | - | 526 | 5.0 |
| HEROIN | | | 38 | 2.7 | 155 | 10.8 | 114 | 8.7 | 64 | 4.7 | 58 | 4.2 | 15 | 0.9 | | | | - | 444 | 4.2 |
| METHADONE | <5 | * | 7 | 0.5[1] | 10 | 0.7 | 23 | 1.7 | 14 | 1.0 | 15 | 1.1 | 23 | 1.4 | <5 | * | | - | 95 | 0.9 |

Rates are reported per 100,000. 2021/22 rates based on 2020 population estimates. NH=Non-Hispanic
* Rate suppressed for count from 1 to 4. [1] Rate should be interpreted with caution, count from 5 to 10.
Drug types are not mutually exclusive and do not sum to the total number of overdoses.
For more information visit https://www.injuryfreenc.ncdhhs.gov/DataSurveillance/Technical-Notes.pdf
Analysis conducted by the NC-DPH, Injury Epidemiology, Surveillance & Informatics Unit. Report date: 09/18/23.

**Details on the HTA file**
Let's go into more detail regarding these topics:

- The main HTA file
- How does it gather parameters to pass to SAS?
- How does it call SAS?
- How does it know who you are?
- Where does it write output?
- What is in MAIN.SAS?
- Can it call SAS Server? Yes!

**The main HTA file**

The main program file is **ORION.HTA**. The HTA file extension, specific to Microsoft Windows, is a text file format that may contain HTML, VBScript and/or JavaScript code. HTA files run with enhanced privileges, and thus have access to information such as user's ID.

You can create and edit your HTA file with Notepad or your favorite text editor. The SciTE editor formats different sections of the code, and what was used for this process. The code was written using Windows 7, and it still works OK in Windows 11. Again, to start the HTA application, just double-click the HTA file within File Explorer.

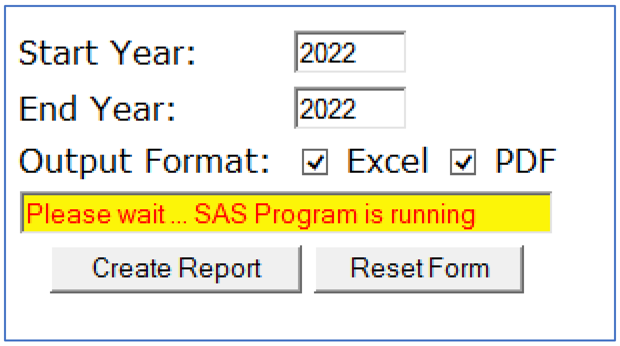
**How does it gather parameters to pass to SAS?**

Think of your HTA file as describing an HTML form and the fields within that form – which might include text boxes, combo-boxes, checkboxes, etc.

The SAMPLE.HTA file (see Appendix 1) creates:

- Two text boxes for year range
- Two checkboxes to select output format
- A progress message text box
- Two buttons: Create Report and Reset Form

When you run the HTA file, a form appears, looking like this:



| | |
|---|---|
| Start Year: | 2022 |
| End Year: | 2022 |
| Output Format: | ☑ Excel ☑ PDF |

Please wait ... SAS Program is running

In the HTA file, you can add client script (such as VBScript or JavaScript).
For example, we might want to write client script to hide the "Please wait" text box when the app first loads, because the SAS program hasn't started running yet. In SAMPLE.HTA, the VBScript subroutine **Sub** Window_onLoad does that by setting the text box's visibility attribute to "hidden".

You probably will also want to add verification code. Perhaps you want to set the minimum and maximum values for the Years, or you don't want to allow Start Year to be after End Year. You might also want to ensure that at least one Output Format checkbox is checked. For simplicity reasons, verification code is not included in SAMPLE.HTA.

**How does it call SAS? How does it know who you are?**

You may already know that you can run Windows PC SAS in batch from a Windows (MS-DOS) Command line, using a command like this:

```
path\sas.exe -sysin sas-program-name -log output-log-filename
        -print results-filename -nologo -rsasuser -sysparm input-parameters
```

In the HTA file, you can assign a subroutine (in VBScript) or function (in JavaScript) to the "Create Report" button's onClick event, which could gather the bolded parameter values (above) and then pass them to the sas.exe command to run. In SAMPLE.HTA, the "Create Report" button's onClick event is set to run the subroutine **Sub Submit**.

When someone presses the "Create Report" button, the VBScript does several things:

- Runs some optional verification subroutines (not shown)
- Stores user-selected choices into "Output" variable
- Shows pop-up box to allow user to select OK or Cancel before proceeding
- Stores full SAS.EXE command into "RunLine" variable, which it passes to the Windows "shell" for execution
- Unhides the "Please wait … SAS program running" textbox

When the SAS program has finished running, the VBScript:

- Opens SAS LOG file, searching for "Error" or "uninitialized" text
  - If found, it opens Notepad and shows LOG file to user.
  - If not found, shows "Report run successfully" message, and shows location of report

**How does it know who you are?**

You could ask the user to enter their network ID into a text box, then use that. But HTA files can use a Windows backdoor approach, by running this VBScript:

```
Set oNetwork = CreateObject( "WScript.Network" )
Dim userID
userID = oNetwork.UserName
```

**Where does it write output?**

You could write it anywhere. In our case, we wanted each user's reports to be sent to an ~~Ouptut~~output folder, then a subfolder based their network (Windows) ID, and within that, a subfolder with the report's date. So, the folder structure might appear as this:

```
Output
        UserID1
                2023-09-19
                2023-09-15
        UserID2
                2023-08-15
                2023-08-14
```

**What is in MAIN.SAS?**
In ORION, the MAIN.SAS code includes all our LIBNAME statements, pointing to directories where SAS data is stored. It sets up a macro pointing to the location for output reports, using the SYSUSERID automatic SAS macro variable:

```
%let OutputDir = I:\ORION\Output\&sysuserid\%SYSFUNC(date(), YYMMDD10.);
```

Our ORION SAS code then calls several macros in sequence which gather the parameters passed in on SAS.EXE command line, and run customized SAS code to create PROC REPORT output which is saved to &OutputDir. Each macro is stored in its own .SAS file within a Macro subfolder, but you could also reference a SAS autocall macro library.

The SAS macros in our ORION system can reference any passed-in parameters as if they were SAS macro variables, such as dataset type, date range, county selections, output filetypes, etc.

In the sample MAIN.SAS code, we can reference &StartYear, &EndYear, &chkExcel and &chkPDF values. See Appendix 2 for a complete sample version of MAIN.SAS code.

**Can it call SAS Server?**
Yes, it can. See Appendix 3 for more information.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the primary author at:

Bruce Nawrocki
North Carolina Division of Public Health, Injury and Violence Prevention Branch
5505 Six Forks Rd, Raleigh NC 27613
E-mail: bruce.nawrocki@dhhs.nc.gov

**Appendix 1 – The SAMPLE.HTA code**

The code in this HTA file makes some assumptions about the location of **Main.sas** (in **I:\ORION**) and **SAS.exe** ( **C:\Program Files\SASHome\SASFoundation\9.4\sas.exe** ). It also assumes you have already created an empty **I:\ORION\UserPrograms** folder.

```
<html>
<script language="VBScript">
Dim BadYear, BadCheckbox
Dim SASexe, SASProg, SASProgDir, Output

' Get userID
Set oNetwork = CreateObject( "WScript.Network" )
Dim userID
userID = oNetwork.UserName

'SAS program directory – where SAS program (ExecutionFile.sas) and its LOG and LST files are stored
SASProgDir = "I:\ORION\UserPrograms\" & userID & "\"
' Use File System Object to create new folder to store SAS program if necessary. Copy standard
ExecutionFile.sas file into this folder for processing
Set oFSO = CreateObject("Scripting.FileSystemObject")
If Not oFSO.FolderExists(SASProgDir) Then
  oFSO.CreateFolder SASProgDir
End If
oFSO.CopyFile "I:\ORION\Main.sas", SASProgDir, true 'Overwrite if existing
SASProg = SASProgDir & "Main.sas"

Sub Window_onLoad
  window.resizeTo 400,250
  window.moveTo 350,50
  document.getElementbyID("txtProgress").style.visibility = "hidden"
End Sub

Sub Reset
   Location.Reload(True)
End Sub

Sub Submit
  'Call Verification_Sub(s) which return BadYear and BadCheckbox as "Good" or "Bad"
  If  BadYear="Bad" or BadCheckbox = "Bad" Then
    Exit Sub
  End If

  Output = "Report=1," & _
    "StartYear=" & document.getElementbyId("StartYear").value & "," & _
     "EndYear=" & document.getElementbyId("EndYear").value & ","
  If document.getElementByID("chkExcel").checked = True Then
    Output = Output & "," & "chkExcel=1"
  End if
  If document.getElementByID("chkPDF").checked = True Then
    Output = Output & "," & "chkPDF=1"
  End if

  SASExe = "C:\Program Files\SASHome\SASFoundation\9.4\sas.exe"

  Call SASBatch
End Sub
```

```
Sub SASBatch
  Set FSO = CreateObject("Scripting.FileSystemObject")
  If not FSO.FileExists(SASProg) Then
    Msgbox "ERROR. The execution file is missing: " & vbcrlf & SASProg & "."
    Exit Sub
  End If
  If not FSO.FileExists(SASExe) Then
    Msgbox "ERROR. Location of SAS executable is mistyped or missing: " & vbcrlf & SASExe & "."
    Exit Sub
  End If

  Dim Msg, Response
  Msg = "If you want to run this SAS program press OK, or press Cancel to return."
  Response = MsgBox(Msg, vbOKCancel)
  If Response = vbCancel Then
    Exit Sub
  End If

  document.getElementbyID("txtProgress").style.visibility = "visible"

  BaseFileName = FSO.GetParentFolderName(SASProg) & "\" & FSO.GetBaseName(SASProg)
  LogFile = BaseFileName & ".log"
  ListFile = BaseFileName & ".lst"

  RunLine =   DQ & """" & SASExe & """"   _
            & DQ & " -sysin " _
            & DQ & """" & SASProg & """"  _
            & DQ & " -log   " _
            & DQ & """" & LogFile & """"  _
            & DQ & " -print " _
            & DQ & """" & ListFile & """" _
            & DQ & " -nologo " _
            & " -rsasuser " _
            & " -sysparm " & DQ & """" & Output & """" & DQ

  Dim oShell
  Set oShell = CreateObject("WScript.Shell")
  Call oShell.Run(RunLine, SHOWMINIMIZED, True)

  Dim LogFile, Contents
  Const FORREADING = 1
  Set LogFilePath = FSO.OpenTextFile(LogFile, FORREADING)
  Contents = LogFilePath.ReadAll

  'Hide progress textbox after program has run
  document.getElementbyID("txtProgress").style.visibility = "hidden"

  Set Rgx = New RegExp
  With Rgx
    .Pattern = "(error:|uninitialized)(?! your system is scheduled to expire on)"
    .Pattern = "(error:|uninitialized)(?! (the .{4,15} product with which|your system is scheduled))"
    .Pattern = "(\n(error:)|uninitialized|remerg)(?! (the .{4,15} product with which|your system is
scheduled))"
    .Global = False
    .IgnoreCase = True
  End With

  If Rgx.Test(Contents) Then
```

```
      MsgBox FSO.GetFileName(SASProg) & " ran with 'Error' or 'Uninitialized' value --check your log
file!", vbOKOnly + vbCritical, "Warning"
      oShell.Run("notepad " & DQ & LogFile & DQ )
   Else
      MsgBox "Report ran successfully!" & vbcrlf & vbcrlf & "It is saved to a subfolder in:" & vbcrlf &
vbcrlf & "I:\ORION\Output" & vbcrlf & vbcrlf & "The subfolder is your userID, then today's date, as in
'YYYY-MM-DD'"
   End If

End Sub
</script>

<body STYLE="font:14pt verdana;color:black">
<table>
<tr><td>Start Year:</td>
<td><input type="text" name="StartYear" size=5 value="1980"
onChange="ChangeEndYear"></td></tr>

<tr><td> End Year:</td>
<td><input type="text" name="EndYear" size=5 value="1994"
onChange="ChangeChkSplitYear"></td></tr>

<tr><td>Output Format: </td>
<td><input type="checkbox" id="chkExcel" name="chkExcel" checked>
<label for="chkExcel">Excel</label>
<input type="checkbox" id="chkPDF" name="chkPDF" checked>
<label for="chkPDF">PDF</label> </td></tr>

<tr><td colspan=2 align="center"><input type="text" name="txtProgress" size=40
style="background-color:#FCF508;color:#FF0000;"value="Please wait ... SAS Program is
running"></td></tr>

<tr><td colspan=2 align="center"><input type="button" value="Create Report" onClick="Submit">
<input type="button" value="Reset Form" onClick="Reset" > </td></tr>
</table>
</body></html>
```

## Appendix 2: SAS Code called from above SAMPLE.HTA file → MAIN.SAS

This code assumes you already have an **I:\ORION\Output** folder created

```
options mlogic symbolgen noquotelenmax;

%let Folder = I:\ORION;

** Set output location;
%let OutputLocation = I:\ORION\Output;
%let OutputDir = &OutputLocation.\&sysuserid\%SYSFUNC(date(), YYMMDD10.);

*** Where is the SAS Data Stored?;
* LIBNAME references go here;

*** Load the macros - if you have them as separate .sas files, or call autocall macro
library
*** For simplicity, all macros are written inside this code (below);
***%include "&Folder./Prod/Macro Files/*.sas";

*** Load formats or reference format library, if any;

%macro CheckandCreateDir(dir);
* Check if output directories exist;
  %put &dir;
  options noxwait;
  %local rc fileref;
  %let rc = %sysfunc(filename(fileref,&dir));
  %if %sysfunc(fexist(&fileref)) %then %put The directory "&dir" already exists;
  %else %do;
    %sysexec mkdir "&dir";
       %if &sysrc eq 0 %then %put The directory &dir has been created.;
       %else %put There was a problem while creating the directory &dir;
       %end;
%mend CheckandCreateDir;

%CheckandCreateDir(&OutputDir);


%macro GetSystemParameters;
*** Load Report Parameters From GUI;
       data _null_;
         length  sysparm express param value $ 20000;
         sysparm = symget('sysparm');
         do i=1 to 50 until(express = '');
           express = left(scan(sysparm, i, ','));
           param   = left(upcase(scan(express, 1, '=')));
           value   = left(scan(express, 2, '='));
           valid   = not verify(substr(param, 1, 1),
                                'ABCDEFGHIJKLMNOPQRSTUVWXYZ_')
             and     not verify(trim(param),
                                'ABCDEFGHIJKLMNOPQRSTUVWXYZ_0123456789')
             and     length(param) <=32;
           if valid then do;
             call symput(param, trim(left(value)));
             sss = trim(left(value));
             put param "=" sss;
             end;
           end;
       run;
%mend GetSystemParameters;

%GetSystemParameters;
```

```
%macro RunReport();
*** Create the PROC PRINT report;
       options nonumber nodate missing=' ';
       OPTIONS papersize=letter orientation=landscape LEFTMARGIN=0.1in
RIGHTMARGIN=0.1in TOPMARGIN=0.25in BOTTOMMARGIN=0.1in;
       %IF %symexist(chkPDF) %THEN %DO;
              ods pdf file="&OutputDir.\MyReport.pdf" notoc style=journal dpi=300;
       %END;
       %IF %symexist(chkExcel) %THEN %DO;
              ods excel file="&OutputDir.\MyReport.xlsx";
       %END;

       proc print data=sashelp.retail;
              title "Report for &StartYear to &EndYear";
              where year between &StartYear and &EndYear;
              var year date day month sales;
       run;

       ods excel close;
       ods pdf close;

%mend;

%RunReport();
```

**Appendix 3: How to connect to a remote SAS Server**

First, in your .HTA script you would change the location of "sas.exe" on your server:

```
  SASExe = "/opt/sas/spre/home/SASFoundation/bin/sas_u8"
```

You must also modify the SASProgDir variable, so it points to location of remote SAS Server, and add a few other bits of information. The added complexity is due to you having to keep track of the Linux path and the references to Linux path from Windows:

```
'SAS program directory – where SAS program (ExecutionFile.sas) and its LOG and LST files are stored
SASProgDir = "\\server-name\ORION\Prod\UserPrograms\" & userID & "\"
SASServer = "server-name"
SASProgDirOnLinux = "/home/EADS/ORION/Prod/UserPrograms/" & userID & "/"
SASLogDir = "Y:\ORION\Prod\UserPrograms\" & userID & "\"

' Use File System Object to create new folder to store SAS program if necessary. Copy standard
ExecutionFile.sas file into this folder for processing
 Set oFSO = CreateObject("Scripting.FileSystemObject")
 If Not oFSO.FolderExists(SASProgDir) Then
   oFSO.CreateFolder SASProgDir
End If
oFSO.CopyFile "\\server-name\IVP\ORION\Prod\ExecutionFile.sas", SASProgDir, true 'True means
overwrite if already existing

 SASProg = SASProgDirOnLinux & "ExecutionFile.sas"
```

You must also change this section of the original code, since how we call SAS, which is now on remote Linux server) is different:

```
   BaseFileName = SASProgDirOnLinux & "ExecutionFile"
   LogFile = BaseFileName & ".log"
   ListFile = BaseFileName & ".lst"

   SSHExe = "cmd /K C:\Temp\ssh.exe"
   RunLine =  SSHExe & " -t " & userid & "@" & SASServer & " " _
              & DQ & """" & SASExe & """"   _
              & DQ & " -sysin " _
              & DQ & """" & SASProg & """"  _
              & DQ & " -log   " _
              & DQ & """" & LogFile & """"  _
              & DQ & " -print " _
              & DQ & """" & ListFile & """" _
              & " -rsasuser " _
              & " -sysparm " & DQ & """" & Output & """" & DQ
   Msgbox "On next screen, enter your NCID password to log into SAS Server. SAS program will run.
Window will remain open while SAS programs runs."

   Dim oShell
   Set oShell = CreateObject("WScript.Shell")
   Call oShell.Run(RunLine & "& exit", 1, True)

   Dim LogFile
   Dim Contents
   Const FORREADING = 1
```

```
    LocalLogFile = SASProgDir & "ExecutionFile.log"
    Set LogFilePath = FSO.OpenTextFile(LocalLogFile, FORREADING)
    Contents = LogFilePath.ReadAll
    Set Rgx = New RegExp
    With Rgx
      .Pattern = "(error:|uninitialized)(?! your system is scheduled to expire on)"
      .Pattern = "(error:|uninitialized)(?! (the .{4,15} product with which|your system is scheduled))"
      .Pattern = "(\n(error:)|uninitialized|remerg)(?! (the .{4,15} product with which|your system is
scheduled))"
      .Global = False
      .IgnoreCase = True
    End With

   'Hide progress textbox after program has run
   document.getElementbyID("txtProgress").style.visibility = "hidden"

    If Rgx.Test(Contents) Then
      MsgBox FSO.GetFileName(SASProg) & " ran with 'Error' or 'Uninitialized' value --check your log
file!", vbOKOnly + vbCritical, "Warning"
      oShell.Run("notepad " & DQ & LocalLogFile & DQ )
    Else
      MsgBox "Report ran successfully!" & vbcrlf & vbcrlf & "It is saved to a subfolder in:" & vbcrlf &
vbcrlf & "Y:\ORION\Output" & vbcrlf & vbcrlf & "The subfolder is your userID, then today's date, as in
'YYYY-MM-DD'"
    End If
```