Paper 131
A Quick Bite on SAS® Studio Custom Tasks

Michael Kola
SalientCRGT
Michael Kola is a SAS Programmer at SalientCRGT where he has been for 2 Years. Michael uses SAS® in his daily responsibilities of SAS Support. He has used SAS for more than 10 years.
INTRODUCTION

SNAPSHOT of the Custom Task Built For this Presentation
Results Window of the Custom Task

<table>
<thead>
<tr>
<th>Obs</th>
<th>Origin</th>
<th>Make</th>
<th>Model</th>
<th>Type</th>
<th>DriveTrain</th>
<th>MSRP</th>
<th>Invoice</th>
</tr>
</thead>
<tbody>
<tr>
<td>374</td>
<td>Asia</td>
<td>Toyota</td>
<td>Prius 4dr (gas/electric)</td>
<td>Hybrid</td>
<td>Front</td>
<td>$20,510</td>
<td>$19,926</td>
</tr>
<tr>
<td>375</td>
<td>Asia</td>
<td>Toyota</td>
<td>Sequoia SR5</td>
<td>SUV</td>
<td>All</td>
<td>$35,695</td>
<td>$31,827</td>
</tr>
<tr>
<td>376</td>
<td>Asia</td>
<td>Toyota</td>
<td>4Runner SR5 V6</td>
<td>SUV</td>
<td>All</td>
<td>$27,710</td>
<td>$24,801</td>
</tr>
<tr>
<td>377</td>
<td>Asia</td>
<td>Toyota</td>
<td>Highlander V6</td>
<td>SUV</td>
<td>All</td>
<td>$28,900</td>
<td>$24,915</td>
</tr>
<tr>
<td>378</td>
<td>Asia</td>
<td>Toyota</td>
<td>Land Cruiser</td>
<td>SUV</td>
<td>All</td>
<td>$34,680</td>
<td>$34,460</td>
</tr>
<tr>
<td>379</td>
<td>Asia</td>
<td>Toyota</td>
<td>RAV4</td>
<td>Sedan</td>
<td>All</td>
<td>$20,200</td>
<td>$18,553</td>
</tr>
<tr>
<td>380</td>
<td>Asia</td>
<td>Toyota</td>
<td>Corolla CE 4dr</td>
<td>Sedan</td>
<td>Front</td>
<td>$14,065</td>
<td>$13,065</td>
</tr>
<tr>
<td>381</td>
<td>Asia</td>
<td>Toyota</td>
<td>Corolla S 4dr</td>
<td>Sedan</td>
<td>All</td>
<td>$15,030</td>
<td>$13,650</td>
</tr>
<tr>
<td>382</td>
<td>Asia</td>
<td>Toyota</td>
<td>Corolla LE 4dr</td>
<td>Sedan</td>
<td>All</td>
<td>$15,250</td>
<td>$13,599</td>
</tr>
<tr>
<td>383</td>
<td>Asia</td>
<td>Toyota</td>
<td>Echo 2dr manual</td>
<td>Sedan</td>
<td>Front</td>
<td>$10,700</td>
<td>$10,144</td>
</tr>
<tr>
<td>384</td>
<td>Asia</td>
<td>Toyota</td>
<td>Echo 2dr auto</td>
<td>Sedan</td>
<td>Front</td>
<td>$11,990</td>
<td>$10,896</td>
</tr>
<tr>
<td>385</td>
<td>Asia</td>
<td>Toyota</td>
<td>Echo 4dr</td>
<td>Sedan</td>
<td>All</td>
<td>$11,200</td>
<td>$10,642</td>
</tr>
<tr>
<td>386</td>
<td>Asia</td>
<td>Toyota</td>
<td>Camry LE 4dr</td>
<td>Sedan</td>
<td>All</td>
<td>$19,500</td>
<td>$17,550</td>
</tr>
<tr>
<td>387</td>
<td>Asia</td>
<td>Toyota</td>
<td>Camry LE V6 4dr</td>
<td>Sedan</td>
<td>All</td>
<td>$22,770</td>
<td>$20,325</td>
</tr>
<tr>
<td>388</td>
<td>Asia</td>
<td>Toyota</td>
<td>Camry Solara SE 2dr</td>
<td>Sedan</td>
<td>All</td>
<td>$21,635</td>
<td>$17,722</td>
</tr>
<tr>
<td>389</td>
<td>Asia</td>
<td>Toyota</td>
<td>Camry Solara SE V6 2dr</td>
<td>Sedan</td>
<td>All</td>
<td>$21,985</td>
<td>$18,819</td>
</tr>
<tr>
<td>390</td>
<td>Asia</td>
<td>Toyota</td>
<td>Avalon XL 4dr</td>
<td>Sedan</td>
<td>All</td>
<td>$26,980</td>
<td>$23,603</td>
</tr>
<tr>
<td>391</td>
<td>Asia</td>
<td>Toyota</td>
<td>Avalon XLS V6 4dr</td>
<td>Sedan</td>
<td>All</td>
<td>$28,920</td>
<td>$23,125</td>
</tr>
<tr>
<td>392</td>
<td>Asia</td>
<td>Toyota</td>
<td>Avalon XLS V6 4dr</td>
<td>Sedan</td>
<td>All</td>
<td>$28,920</td>
<td>$23,125</td>
</tr>
<tr>
<td>393</td>
<td>Asia</td>
<td>Toyota</td>
<td>Avalon XLS 4dr</td>
<td>Sedan</td>
<td>All</td>
<td>$28,920</td>
<td>$23,125</td>
</tr>
<tr>
<td>394</td>
<td>Asia</td>
<td>Toyota</td>
<td>Sienna CE</td>
<td>Sedan</td>
<td>All</td>
<td>$23,450</td>
<td>$21,198</td>
</tr>
<tr>
<td>395</td>
<td>Asia</td>
<td>Toyota</td>
<td>Sienna XLE Limited</td>
<td>Sedan</td>
<td>All</td>
<td>$22,600</td>
<td>$22,600</td>
</tr>
<tr>
<td>396</td>
<td>Asia</td>
<td>Toyota</td>
<td>MR2 Spyder convertible 2dr</td>
<td>Rear</td>
<td>Rear</td>
<td>$24,465</td>
<td>$19,072</td>
</tr>
<tr>
<td>397</td>
<td>Asia</td>
<td>Toyota</td>
<td>Tacoma</td>
<td>Truck</td>
<td>Rear</td>
<td>$11,800</td>
<td>$11,870</td>
</tr>
<tr>
<td>398</td>
<td>Asia</td>
<td>Toyota</td>
<td>Tundra Regular Cab V6</td>
<td>Truck</td>
<td>All</td>
<td>$15,495</td>
<td>$14,075</td>
</tr>
<tr>
<td>399</td>
<td>Asia</td>
<td>Toyota</td>
<td>Tundra Access Cab V6 SR5</td>
<td>Truck</td>
<td>All</td>
<td>$25,950</td>
<td>$23,020</td>
</tr>
<tr>
<td>400</td>
<td>Asia</td>
<td>Toyota</td>
<td>Tundra Regular Cab V6</td>
<td>Truck</td>
<td>All</td>
<td>$15,495</td>
<td>$14,075</td>
</tr>
</tbody>
</table>

Filter 1:
- Variable 1:
  - Origin
  - Make
  - Model

Filter 2:
- Variable 3:
  - Make
  - Value:
    - Toyota

Dataverse:
SASHELP.CARS

Procedure:
Select from the List: Print
Two Ways to Create Custom Tasks

There are two ways to create custom tasks. One way to create is by expanding the “Tasks and Utilities” and right clicking on the Tasks and selecting “New Task”
Another way to create is selecting the existing SAS task and adding to My Tasks. This can be done by right clicking the existing SAS task for example “List Data” and selecting “Add to My Task” This will add the existing task to My Task and now the user will have the ability to edit this task.
The REGISTRATION element represents a collection of metadata for the task. This element is required in order to know the type of task. In the below code the registration element contains child elements NAME, DESCRIPTION, GUID, PROCEDURE, VERSION and the hyper LINK the values of these elements are self-explanatory. Since this task is built for the sole purpose of this paper the child elements describe the current task type in the REGISTRATION.
<Registration>
  <Name>A Quick Bite on SAS Studio Custom Tasks</Name>
  <Description>Custom Built Task</Description>
  <GUID>726B2B12-4542-4200-96F4-960C645D767F</GUID>
  <Procedures>MEAN FREQ PRINT</Procedures>
  <Version>3.5</Version>
  <Links>
    <Link href="http://support.sas.com/software/products/sasstudio/index.html#s1=2">SAS Studio Documentation</Link>
  </Links>
</Registration>
When the above code is executed following User Interface is displayed under INFORMATION tab:

- **Name:** A Quick Bite on SAS Studio Custom Tasks
- **Description:** Custom Built Task
- **Category:** None
- **Procedures:** MEAN FREQ PRINT
- **Version:** 3.5

**RESOURCES**

- [SAS Studio Documentation](#)
The following code is for the METADATA element. This element contains child elements which are DATASOURCES and OPTIONS. The DATASOURCES contains one child element which is DATASOURCE. This element specifies the information about the dataset for the task. THE DATASOURCE also contains a child element which is Roles which identifies the variables that must be assigned in order to run the task and groups the individual role assignments that are needed for a task. The Role tag, which is the only child of the roles element, describes one type of role assignment for the task.
<Metadata>

<DataSources>

<DataSource name="dataset">

<Roles>

<Role maxVars="0" minVars="0" name="var1" order="true" type="A">Variable 1:</Role>
<Role maxVars="1" minVars="0" name="var2" order="true" type="A">Variable 2:</Role>
<Role maxVars="1" minVars="0" name="var3" order="true" type="A">Variable 3:</Role>

</Roles>

</DataSource>

</DataSources>
When you run this code, you get the data and roles section as follows:
The DATASOURCE element helps display the DATASET and the Roles are Variable1, Variable2 and Variable3:
When you run this code, you get the data and roles section as follows. The DATASOURCE element helps display the DATASET and the Roles are Variable1, Variable2 and Variable3:
The OPTIONS element identifies the options that are required in order to run the task. The option tag, which is the only child of the options element, describes the assigned option.

```xml
<Options>
  <Option inputType="string" name="datatab">DATA</Option>
  <Option inputType="string" name="datagrp">DATASOURCE</Option>
  <Option name="PROCS" inputType="string">PROC GROUP</Option>
  <Option defaultValue="Print" inputType="combobox"
       name="List" width="264px">Select from the List:</Option>
  <Option inputType="string" name="Print">Print</Option>
  <Option inputType="string" name="Freq">Freq</Option>
  <Option inputType="string" name="Means">Means</Option>
  <Option inputType="string" name="filter1">FILTER 1</Option>
  <Option inputType="string" name="distinct" name="values1" source="var2" max="200">Value:</Option>
  <Option inputType="string" name="filter2">FILTER 2</Option>
  <Option inputType="string" name="distinct" name="values2" source="var3">Value2:</Option>
</Options>
```
OPTIONS

The output for the above options will display the following screen:
OPTIONS

Data set name (Please name your output data set):

SASGF18

Rows to list:

All rows

Rows (n):

10
<Option inputType="string" name="statistics">STATISTICS</Option>
<Option inputType="string" name="GROUPCHECK">Means STATISTICS</Option>
<Option inputType="string" indent="1" name="labelCHECK">Select the statistics you would like included in the output dataset.</Option>
<Option defaultValue="0" indent="1" inputType="checkbox" name="chkMEAN">Mean</Option>
<Option defaultValue="0" indent="1" inputType="checkbox" name="chkMIN">Minimum</Option>
<Option defaultValue="0" indent="1" inputType="checkbox" name="chkMAX">Maximum</Option>
<Option defaultValue="0" indent="1" inputType="checkbox" name="chkSTD">Standard Deviation</Option>
<Option defaultValue="0" indent="1" inputType="checkbox" name="chkLCLM">LCLM</Option>
<Option defaultValue="0" indent="1" inputType="checkbox" name="chkUCLM">UCLM</Option>
<Option defaultValue="0" indent="1" inputType="checkbox" name="chkP25">25th Percentile</Option>
<Option defaultValue="0" indent="1" inputType="checkbox" name="chkP75">75th Percentile</Option>
The UI Element contains CONTAINER child element and the CONTAINER contains GROUP Element as child and the GROUP contains the DATAITEM, OPTIONCHOICE and ROLEITEM as child elements. The OPTIONCHOICE contains OPTIONITEM as child.

```xml
<UI>
  <Container option="datatab">
    <Group open="true" option="datagrp">
      <DataItem data="dataset"/>
    </Group>
    <Group open="true" option="PROC$">
      <OptionChoice option="List">
        <OptionItem option="Print"/>
        <OptionItem option="Freq"/>
        <OptionItem option="Means"/>
      </OptionChoice>
    </Group>
    <Group open="true" option="filter1">
      <RoleItem role="var1"/>
      <RoleItem role="var2"/>
      <OptionItem option="values1"/>
    </Group>
    <Group open="true" option="filter2">
      <RoleItem role="var3"/>
      <OptionItem option="values2"/>
    </Group>
  </Container>
</UI>
```
The following is the UI for the DATA container:
<Dependency condition="$('Tr-Tr').size() > 0">
  <Target action="enable" conditionResult="true" option="var3"/>
</Dependency>

<Dependency condition="$('rows2List').equalsIgnoreCase('firstrows')">
  <Target action="enable" conditionResult="true" option="NVALUE"/>
</Dependency>

<Dependency condition="$('rows2List').equalsIgnoreCase('all')">
  <Target action="disable" conditionResult="false" option="NVALUE"/>
</Dependency>
When the above code is executed, all the dependency conditions are applied. The variables Variable1 and Variable2 are disabled when Freq is selected.
TASK ELEMENTS

REQUIREMENTS

This element acts like the DEPENDENDENCY element. This element specifies a list of condition that must be met in order for the task to run. If the condition is true, SAS code can be generated. If the condition is false, no code is generated. When defining a requirement, you can specify the message to display when the requirement is not met.

In the below code the condition is ($var1.size() &gt; 0) this means if the value of the var1 is great than 0 then the condition is met and if the value is not great then 0 i.e.; no value is entered the condition is not met and the message will be displayed.

```xml
<Requirements>
  <Requirement condition="$var1.size() &gt; 0">
    <Message>At least one variable must be assigned to the Variable1.</Message>
  </Requirement>
</Requirements>
```

Once the above code is executed the following screen is displayed. If you notice the highlighted areas the message is displayed “At least one variable must be assigned to the Variable1” Because the Variable1 does not have any columns selected.
*/
* Code cannot be generated because the following requirements are not met:
* At least one variable must be assigned to the Variable1.
*/
The code template creates the string output of the task. For most tasks, this output is SAS code. The code Template element contains a CDATA block of the Apache Velocity scripting language. The string output is produced using this scripting language.

Here is the code that I’m using for PROC PRINT and PROC FREQ. I am using the IF statement in the velocity code to choose the PRINT/FREQ/MEANS. The $List is the velocity variable which is created in the options element which is the combo box. Following is the code.

```html
<Option defaultValue='Print' inputType='combobox' name='List' width='264px'>Select from the List:</Option>
```

And the UI has the option to choose from PRINT, FREQ and MEANS as it is shown in the below screen.
In the following Velocity code if the “IF STATEMENT” has PRINT. It will execute PROC PRINT part of the Velocity Code

```velocity
<CodeTemplate>
  <![CDATA[

#if($List.equals("Print"))
  Proc Print data=$dataset #if($ROWS2LIST.equals("firstnrows"))(obs=$NVALUE) #end
    #if($ROWS2LIST.equals("all"))
    #end;
  #if( $var1.size() > 0 )
    var #foreach( $item in $var1 ) $item#end;
  #end
  #if( $var2.size() > 0 )
    Where #foreach( $item in $var2 ) $item#end = '$values1'
  #end
  #if( $var3.size() > 0 )
    AND #foreach( $item in $var3 ) $item#end = '$values2'
  #end;

run;
#end
```
The following screen shows the SAS code on the left which is displayed when the Print is selected in the PROC Group and code for FREQ is displayed when the FREQ is selected and same for the Means.
The Custom Task is created and now the task can be saved as CTM or CTK files and be shared with the users by downloading from SAS Studio and could be emailed or place in a share drive. The user will have to upload the files to SAS Studio in order to execute and produce the results.

The upload and download could be done by right clicking on the file and selecting upload/download and follow the directions to select the path where the file need to be saved.
CONCLUSION

All the elements which are used in the custom task building must be put together in order for the Task to generate the application/UI Screen. Executing the task elements individually may not produce the desired results. The REGISTRATION is used for identity and the DATASOURCES and ROLES are used for SAS Data Set and variable selection. UI is used to display User Interface part of the task. CODE Template is used for writing the Velocity Code.

The entire CTM code is run to produce the CTK output which is the User Interface. Once the UI is displayed, the Data Set, variables etc. are selected. After the data selection the SAS code is generated on the code window. Then the SAS code is submitted by hitting this button and the result is generated in the Results window.
Your comments and questions are valued and encouraged. Contact the author at:

Michael Kola
SalientCRGT
Michael.Kola@salientcrgt.com
Kola_mail@yahoo.com

@Michael_Kola