CUSTOM TASKS CHEAT SHEET

SAS® STUDIO

Custom Tasks are point-and-click interfaces you can create for generating and running SAS code. Tasks are made up of 6 sections:

REGISTRATION

Includes elements of the task such as Name, Description, PROCs used, and links to helpful information.

METADATA (MD)

Defines the data source, role objects, and controls you want in your task.

UI

Defines the layout of the user interface and order of objects you listed in the Metadata section. Only a top-down layout is supported.

DEPENDENCIES NOT REQUIRED

Specifies how certain options (or controls) rely on one another in order for the task to work properly.

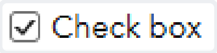
REQUIREMENTS NOT REQUIRED

Specifies conditions for the task to run. If the condition is true, SAS code can be generated.

CODE TEMPLATE

Written in Apache Velocity Template Language. The task triggers the SAS code, filled in with Velocity Macro Variables from the corresponding controls.

CHECKBOX


☑ Check box

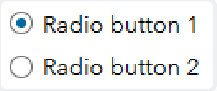
MD

```
<Option name="chkEXAMPLE" defaultValue="1"
inputType="checkbox">Check box</Option>
```

UI

```
<OptionItem option="chkEXAMPLE"/>
```

RADIO BUTTONS


● Radio button 1
○ Radio button 2

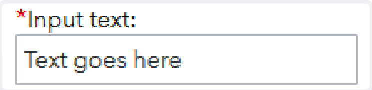
MD

```
<Option name="radioButton1"
variable="radioEXAMPLE" defaultValue="1"
inputType="radio">Radio button 1</Option>
<Option name="radioButton2"
variable="radioEXAMPLE"
inputType="radio">Radio button 2</Option>
```

UI

```
<OptionItem option="radioButton1"/>
<OptionItem option="radioButton2"/>
```

INPUT


*Input text:
Text goes here

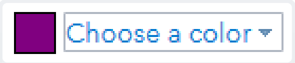
MD

```
<Option name="textEXAMPLE" defaultValue="Text
goes here" inputType="inputtext"
required="true" promptMessage="Enter text."
missingMessage="Missing text.">Input text:
</Option>
```

UI

```
<OptionItem option="textEXAMPLE"/>
```

COLOR SELECTOR



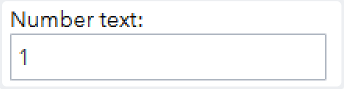
MD

```
<Option name="colorEXAMPLE"
defaultValue="purple" inputType="color">Choose
a color</Option>
```

UI

```
<OptionItem option="colorEXAMPLE"/>
```

NUMBER TEXT


Number text:
1

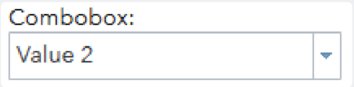
MD

```
<Option name="numberTextEXAMPLE"
defaultValue="1"
inputType="numbertext"
minValue="0"
maxValue="100"
promptMessage="Enter a number between 0 and
100."
missingMessage="Enter a number between 0 and
100."
rangeMessage="This number is out of range.
Enter a number between 0 and 100."
invalidMessage="Invalid value. Enter a
number between 0 and 100.">Number text:
</Option>
```

UI

```
<OptionItem option="numberTextEXAMPLE"/>
```

COMBO BOX


Combobox:
Value 2 ▾

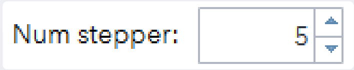
MD

```
<Option name="comboEXAMPLE"
defaultValue="value2"
inputType="combobox">Combobox:</Option>
<Option name="value1" inputType="string">Value
1</Option>
<Option name="value2" inputType="string">Value
2</Option>
```

UI

```
<OptionChoice option="comboEXAMPLE">
    <OptionItem option="value1"/>
<OptionItem option="value2"/>
</OptionChoice>
```

NUMSTEPPER


Num stepper: 5

MD

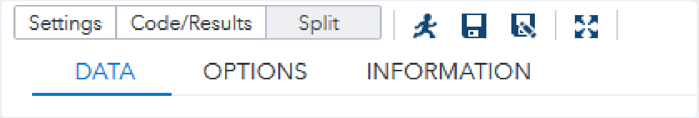
```
<Option name="basicStepperEXAMPLE"
defaultValue="5" inputType="numstepper">
Num stepper:</Option>
```

UI

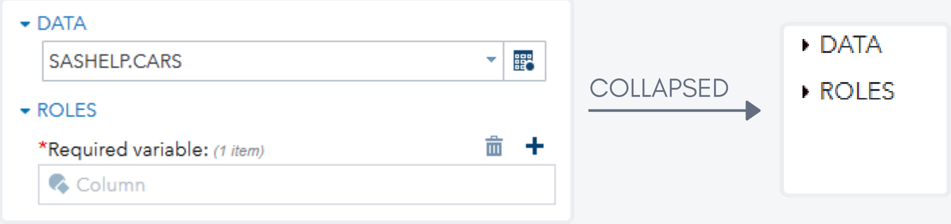
```
<OptionItem option="basicSTEPPEREXAMPLE"/>
```

TABS, GROUPS, & DATA CONTROLS

It is typical for most tasks to have DATA and OPTIONS tabs. The INFORMATION tab is created automatically and contains the info from the REGISTRATION section of the task.



Groups are optional, collapsible sections of controls.



Data controls are data set selectors and role (variable) selectors.

MD

```
<DataSources>
    <DataSource name="DATASOURCE">
        <Roles>
            <Role type="A" maxVars="1" order="true" minVars="1"
                 name="VAR">Required variable:</Role>
        </Roles>
     </DataSource>
</DataSources>
<Options>
    <Option name="DATATAB" inputType="string">DATA</Option>
    <Option name="DATAGROUP" inputType="string">DATA</Option>
    <Option name="ROLESGROUP" inputType="string">ROLES</Option>
</Options>
```

UI

```
<Container option="DATATAB">
    <Group option="DATAGROUP" open="true">
        <DataItem data="DATASOURCE"/>
    </Group>
    <Group option="ROLESGROUP" open="true">
        <RoleItem role="VAR"/>
    </Group>
</Container>
```

Notes: For a variable selector that only allows character variables, use `type="C"`, and for numeric use `type="N"`. For a variable selector that has no maximum number, use `maxVars="0"`. For a variable selector that isn't required, use `minVars="0"`.

CODE TEMPLATE

To access the value of a control, use a $ followed by the NAME of the control.

```
data new;
    set $DATASOURCE;
run;
```

HELPFUL APACHE VELOCITY CODE

CODE	DESCRIPTION	EXAMPLE

#if	Allows for conditional execution of SAS code based on the value of a control	`#if( $chkSORT == 1 )`
		`  proc sort data=$DATASOURCE;`
		`    by var;`
		`  run;`
		`#end`

#foreach	Allows for cycling through a list of items	`#foreach( $item in $VAR )`
		`  proc means data=$DATASOURCE;`
		`    var $item;`
		`  run;`
		`#end`

size()	Checks that a list is not empty before executing code assuming a list	`#if( $BYVAR.size() > 0)`
		`  #foreach( $item in $BYVAR )`
		`    proc sort data=$DATASOURCE;`
		`      by $item;`
		`  #end`
		`    run;`
		`#end`

length()	Checks to see if a string is empty before executing code assuming a string	`#if( $text.length() > 0 )`
		`  title "$text";`
		`#end`

getLibrary()	Automatically specifies a library using the library used in a data source	`%let`
		`SASLIB = $DATA.getLibrary();`
		`data &SASLIB.new;`
		`run;`

HELPFUL LINKS

SGF Paper: "Developing Your Own SAS® Studio Custom Tasks for Advanced Analytics"

SAS® Studio 3.8 Developer's Guide to Writing Custom Tasks

Custom Task Tuesday article series on SAS Communities

FREE e-Learning on SAS® Studio Custom Tasks

SGF Paper: "Teach Them to Fish—How to Use Tasks in SAS® Studio to Enable CoWorkers to Run Your Reports Themselves"

Custom Task Tuesday GitHub Page

Follow author of #CustomTaskTuesday @OliviaJWright on Twitter

Apache Velocity Website with Resources