Macros In SAS Enterprise Guide

TASS-I, Sept 15, 2011
Rupinder Dhillon
Some ways to use Macros in EG

- Using Macros in code
- Automatic macro variables available to your EG session
- Macros disguised as Prompts
- Using %syslast
- Using the Autoexec flow in your projects
Some ways to use Macros in EG

- Using Macros in code
- Macros disguised as Prompts
- **Automatic macro variables available to your EG session**
- Using `%syslast`
- Using the Autoexec flow in your projects
What is a Macro Variable?

- In a nutshell: text substitution
- Once a macro variable is defined and given a value, ‘&’ is used to resolve the macro variable; it will replace the macro variable with the corresponding value stored in the macro processor
  - We define a macro variable called Rundate and give it a value
    - Call symput (‘rundate’, put(today(), date9.)) <- 15Sep2011
  - Wherever we see a &Rundate., SAS replaces it with 15Sep2011
Most common use of Macros in EG: Prompts

- Prompts allow you to take input from a user and then store the value of that input in macro variables.
- These macro variables can then be used throughout your project.
Automatic Macro Variables

- Your EG session will generate some useful automatic macro variables
- These include:
  - `GLOBAL _CLIENTUSERID 'rdhillon'`
  - `GLOBAL _CLIENTAPP 'SAS Enterprise Guide'`
  - `GLOBAL _CLIENTUSERNAME 'RupinderDhillon'`
  - `GLOBAL _CLIENTTASKLABEL 'Monthly Shoes Report'`
  - `GLOBAL _CLIENTPROJECTNAME 'Monthly Shoes Report.egp'`
Automatic Macro Variables - How can these be used?

Customizing your output in code or in your tasks

```sas
data test;
set sashelp.shoes;
run;

footnote1 h=2 jrc "Produced on &RunDate., by &_ClientUserName."

PROC GCHART DATA=WORK.Test
    VBAR Region / SUMVAR=Sales
    CLIPREF FRAME TYPE=SUM
    COUTLINE=BLACK
    RAXIS=AXIS1 MAXIS=AXIS2
; 
RUN; QUIT:
```
Automatic Macro Variables - How can these be used?

Debugging your multi step Projects – using %put.

```sas
%put Beginning of &_ClientTaskLabel.;

data test;
set sashelp.morshoes;
run;

footnote1 h=2 j=c "Produced on &RunDate., by &_ClientUserName.";

PROC GCHART DATA=WORK.Test
  VBAR Region / SUMVAR=Sales
  CLIPREF
  FRAME TYPE=SUM
  COUTLINE=BLACK
  RAXIS=AXIS1
  MAXIS=AXIS2
; 
RUN; QUIT;

%put End of &_ClientTaskLabel.;
```

```
<?xml version="1.0" encoding="utf-8"?>

<root>
  <monthly-shoes-report>
    <options-accessible>OPTIONS ACCESSIBLE;</options-accessible>
    <put-beginning-of &_client-task-label>put Beginning of &_ClientTaskLabel.;</put-beginning-of>
    <data test>
    set sashelp.morshoes;
    </data>
    <run>
    </run>
    <error>
    ERROR: File SASHELP.MORESHOES.DATA does not exist.
    </error>
    <run>
    </run>
    <note>
    NOTE: Compression was disabled for data set WORK.TEST because compr
    NOTE: The SAS System stopped processing this step because of errors
    WARNING: The data set WORK.TEST may be incomplete. When this step
    WARNING: Data set WORK.TEST was not replaced because this step was
    NOTE: DATA statement used (Total process time):
    real time 0.00 seconds
    cpu time 0.00 seconds
    </note>
  </monthly-shoes-report>
</root>
```
Automatic Macro Variables - How can these be used?

Building in Auditing and Controls

```sql
/* Update Audit Table */

proc sql;
insert into AuditTable
    set ProcessName = "&_CLIENTPROJECTNAME"
    , RunDate = "&_Rundate."
    , RunBy = "&_ClientUserName."
    , Status = 'Complete';
quit;
```

![Audit Table Example]

<table>
<thead>
<tr>
<th>ProcessName</th>
<th>RunDate</th>
<th>RunBy</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monthly Shoes Report</td>
<td>16SEP2011</td>
<td>rdhillon</td>
<td>Complete</td>
</tr>
<tr>
<td>Monthly Shoes Report</td>
<td>31AUG2011</td>
<td>rdhillon</td>
<td>Complete</td>
</tr>
</tbody>
</table>
Automatic Macro Variables

- Others include:
  - GLOBAL _CLIENTUSERID 'rdhillon'
  - GLOBAL _SASHOSTNAME ‘local'
  - GLOBAL _CLIENTAPP 'SAS Enterprise Guide'
  - GLOBAL _CLIENTUSERNAME 'RupinderDhillon'
  - GLOBAL _CLIENTPROJECTPATH 'C:\Users\rdhillon\Desktop\Monthly Shoes Report.egp'
  - GLOBAL _SASPROGRAMFILE
  - GLOBAL _CLIENTTASKLABEL 'Monthly Shoes Report'
  - GLOBAL _SASSERVERNAME ‘Local'
  - GLOBAL _CLIENTPROJECTNAME 'Monthly Shoes Report.egp'
- Check out all your macro variables with %put _all_;
Handy Trick with &syslast

- &syslast automatically stores the last dataset that you processed
- You can override what dataset name is stored in &syslast using links
Using Autoexec with Macros

When you include a flow called Autoexec in your project, it will run automatically whenever your project is opened. You can use this to initiate macro variables that you can use throughout your Project.

Set things like:

- Reporting Dates
- Report Titles
- Segments
- Or specify your autocall macro library location
Using Autoexec with Macros

/* Initiate reporting macros */

/* Set Reporting Date */

data _Null_
       ;
Extract_Start = intnx('month', today(), -1, 'b');
Extract_End = intnx('month', today(), -1, 'e');
call symput('ExtractStart', put(Extract_Start, date9.));
call symput('ExtractEnd', put(Extract_End, date9.));
call symput('RunDate', put(today(), date9.));
run;
Using Autoexec with Macros

This project contains a process flow labeled "Autoexec". SAS Enterprise Guide can run this automatically.

Do you want to run the "Autoexec" flow?

Yes  No
Using Autoexec with Macros

Select to bypass the prompt
Enhance your projects; Get to know the basics

-SAS Macros for Beginners; http://www2.sas.com/proceedings/sugi29/243-29.pdf
-SAS Macro Variables and Simple Macro programs; http://www2.sas.com/proceedings/sugi30/130-30.pdf
- SAS Canada Community - http://sascanada.ning.com/

Contact Me: rupinder@dhillonconsulting.com