



MidWest SAS® Users Group 2010

October 10-12  Milwaukee, Wisconsin

Application Development with SAS Stored Process

John Xu

1ST Consulting LLC, West Des Moines, Iowa





The Need for SAS Application

As SAS Programmer, we often

- Write SAS program with SAS Data step and PROCs to implement business logic.
 - Use SAS Macro for similar tasks or repeated processes.
 - Create SAS Macro Variables to create Input parameters.
-
- This works well for us SAS programmers. We can even set up daily/weekly/month production jobs.



The Need for SAS Application

We often do a great job implementing very complicated business logic, Creating a nice report, or perform extensive statistical analysis.

But end users are more demanding, they often

- Like a Graphic User Interface (GUI).
 - And the output in Excel file.
-



The Need for SAS Application

We often do a great job implementing very complicated business logic, Creating a nice report, or perform extensive statistical analysis.

But end users are more demanding, they often

- Like a Graphic User Interface (GUI).
 - And the output in Excel file.
-



Potential Problems Sharing Code

- Data may not be available: Your C: drive has data but not on others' PC.
 - Different version of data: One might be from this month, other from last month.
 - Different version of Program: You updated the program but others are still using the old code.
 - File Permission issue: Common on UNIX.
-
- User may not know SAS!



Potential Problems Sharing Code

- Data may not be available: Your C: drive has data but not on others' PC.
 - Different version of data: One might be from this month, other from last month.
 - Different version of Program: You updated the program but others are still using the old code.
 - File Permission issue: Common on UNIX.
-
- User may not know SAS!



Advantage of Server Application

- Same Data.
 - Same Program
 - Same result
 - Security
-



What is SAS Stored Process

SAS program stored on a server that can be accessed by a client application.

SAS has done a lot work to allow access from MS Office, Internet, and SAS software tool, such as Enterprise Guide.

For people who know database application, think about Stored Procedure for SQL application.



Advantage of Stored Process

- Code is on the server.
- Access Security
- Data Security
- Code/Logic Security
- Consistent among users
- Easier to support and maintain
- Support multiple client applications
- Support multiple users



Two Parts of the Presentation

- An Introduction to SAS Stored Process through an simple example
 - Practical Design and Implementation of an real world application
-



What Do You Need for SP work?

- Regular SAS is not enough
 - Need SAS Metadata server,
 - Typically, company will license SAS Business Intelligence Package
 - We assume the server is ready
-



Steps in SP Development

- Develop and test your SAS program
- Create SAS macro variables as parameters
- Use Enterprise Guide to Create and test Stored Process
- Configure client application to access Stored Process

Let's get started... Sorry I can only show screens



1 - Develop & Test SAS Program

Create data on server:

```
rsubmit;  
options ls=64;  
libname test '/users/apps/xujohn';
```

```
data test.SP;  
    name='John'; score=80; output;  
    name='Jack'; score=90; output;  
run;
```



1 - Develop & Test SAS Program

Create data on server:

```
rsubmit;  
options ls=64;  
libname test  
'/users/apps/xujohn';  
  
data test.SP;  
    name='John'; score=80;  
output;  
    name='Jack'; score=90;  
output;  
run;
```

SAS Output:

Score		
Obs	name	score
1	John	80
2	Jack	90



2 - Create SAS Macro Variables

(We now want to print the score for a selected name)

```
rsubmit;  
%let name_selected=John;  
  
libname test '/xxxxxx/xujohn';  
  
proc print data=test.SP;  
  where name="&name_selected";  
  title "Score for &name_selected";  
  var name score;  
run;
```

SAS Output:

Score for John

Obs	name	score
1	John	80



3 - Use EG to Create and Test SP

The screenshot shows the SAS Enterprise Guide interface for a project named 'Project_IASUG'. The 'Project Explorer' on the left lists the project structure: 'Project_IASUG' (containing 'Process Flow', 'Log', and 'HTML - IASUG_test'). The main window displays the 'HTML - IASUG_test' output, which includes the SAS logo and the title 'Score for John'. Below the title is a table with the following data:

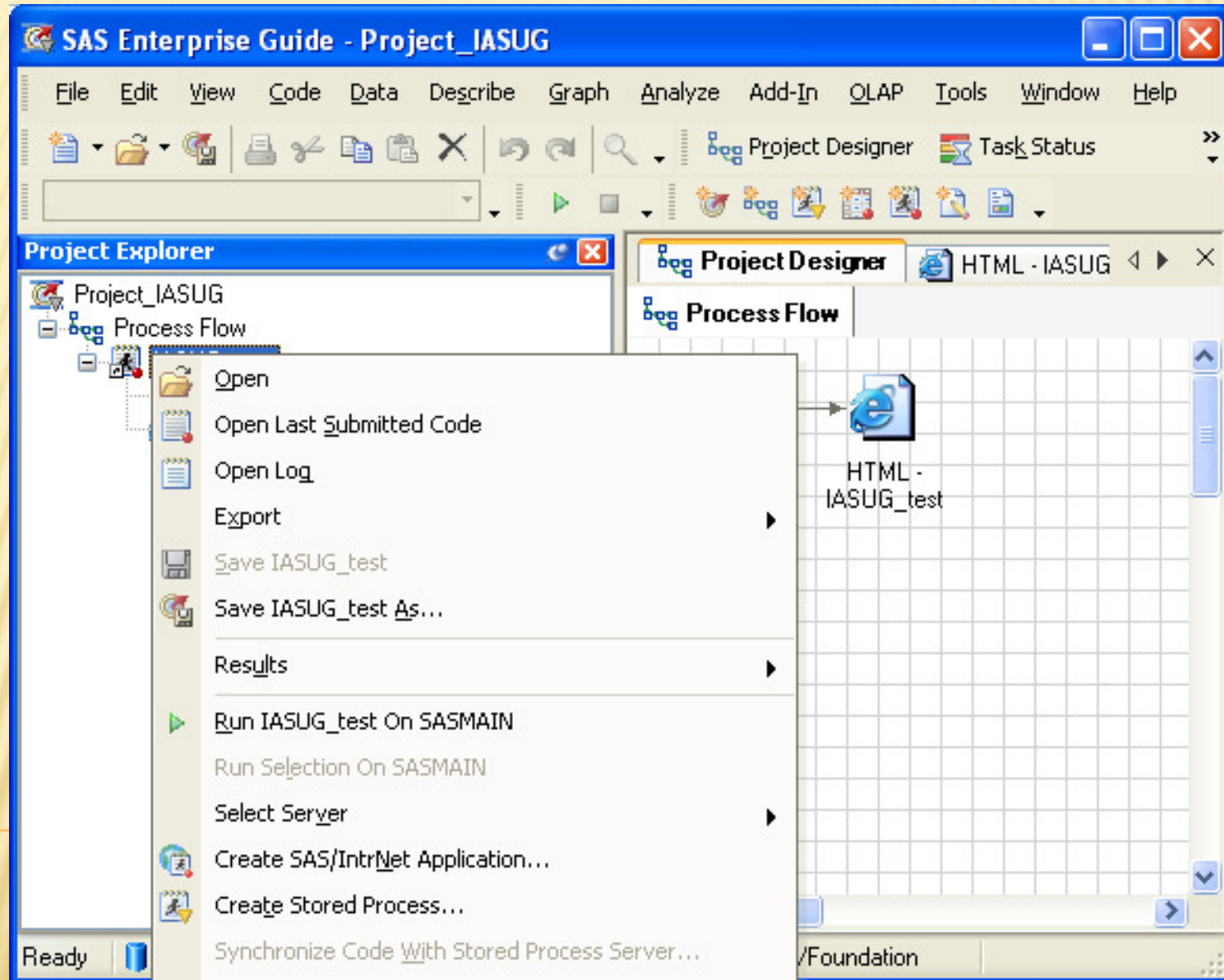
Obs	name	score
1	John	80

The status bar at the bottom indicates the user is 'xujohn as John Xu, connected to sasprod1.wellsfargo.com:8561/Foundation'.

A quick test. Just comment out 'rsubmit' and run SAS program in EG.



3 - Use EG to Create and Test SP





Right click to get popup menu and select Create Stored Process.



3 - Use EG to Create and Test SP

Step 1: General Information

Create New SAS Stored Process Wizard 



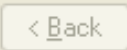
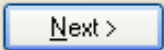

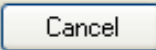
1 of 7 General Information 

Name:
Score Report

Description:
Print Score for a selected name.

Keywords (comma separated):
Report, Score

[More \(F1\)...](#)



3 - Use EG to Create and Test SP

Step 2: SAS Code (Comment out %let lines)

Create New SAS Stored Process Wizard

2 of 7 SAS Code

SAS

```
*rsubmit;  
  
*%let name_selected=John;  
  
libname test '/users/apps/crm_blm/xujohn';  
  
proc print data=test.SP;  
  where name="&name_selected";  
  title "Score for &name_selected";  
  var name score;  
run;
```

Replace with code ▾ Include code for ▾ Clear code Reset code

More (F1)...


< Back ▾ Next > Finish Cancel



3 - Use EG to Create and Test SP

Step 3: Metadata Location

Create New SAS Stored Process Wizard

3 of 7 Metadata Location 

Location of the stored process on the metadata server


[More \(F1\)...](#)



3 - Use EG to Create and Test SP

Step 4: Metadata Location

Create New SAS Stored Process Wizard

4 of 7 Execution Environment 

Execution Environment



Source file name:
Score_Report.sas

Execution Server:
SASMain - Logical Stored Process Server

Source file path on execution server:
/users/apps/crm_blm/model6/horizon/code

[Modify...](#)

Moves to the next step in the wizard. [More \(F1\)...](#)

  [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

[illegible]



3 - Use EG to Create and Test SP

Step 5-2: Parameters (The key step to create user interface)

Automatically Add Parameters from SAS Code

General | Constraints

User prompt: name_selected

SAS variable name: name_selected

Description:

Data type: String

Default Value:

Location:

Options:

- Boolean
- Color
- Date
- File Name
- Float
- Integer
- Multi-line Text
- String**

☒ Modifiable ☒ Visible

Skip this parameter | Rescan code | Add | Close | Cancel



3 - Use EG to Create and Test SP

Step 5-3: Parameters (The key step to create user interface)

The screenshot shows a dialog box titled "Automatically Add Parameters from SAS Code" with a close button (X) in the top right corner. The dialog has two tabs: "General" and "Constraints", with "Constraints" currently selected. The "List Values" section contains a table with three columns: "Display as", "Resolves to", and an empty column. The table has three rows: "John Xu" resolving to "John", "Jack Smith" resolving to "Jack", and a row with an asterisk (*) in the "Display as" column. Below the table are "Delete" and "Load values from..." buttons. The "List Options" section has two radio buttons: "Single selection" (selected) and "Allow multiple selections". Under "Single selection", there is a checkbox for "Allow additional values". Under "Allow multiple selections", there is a section for "User selections allowed" with checkboxes for "Minimum" and "Maximum", each followed by a text input field. Below this is a section for "String length" with checkboxes for "Minimum" and "Maximum", each followed by a text input field. At the bottom of the dialog are five buttons: "Skip this parameter", "Rescan code", "Add", "Close", and "Cancel".

	Display as	Resolves to
	John Xu	John
	Jack Smith	Jack
*		

List Options

☒ Single selection
☐ Allow additional values

☐ Allow multiple selections

User selections allowed

☐ Minimum

☐ Maximum

String length


☐ Minimum

☐ Maximum

Buttons: Delete, Load values from..., Skip this parameter, Rescan code, Add, Close, Cancel



Create New SAS Stored Process Wizard

5 of 7 Parameters 

Group or parameter name	Data type	Options	Description
General	n/a	n/a	
name_selected	String	EMRV	Pick the name from dropdown box

Add Edit Delete Up Down

Displays the groups and parameters for the stored process.

[More \(F1\)...](#)

< Back Next > Finish Cancel



3 - Use EG to Create and Test SP

Step 6: Output Options and Input Streams (default)

Create New SAS Stored Process Wizard

6 of 7 Output Options and Input Streams

Output Options:

☐ None

☒ Streaming output

☐ Create HTML user interface

☐ Transient package of files

☐ Permanent

File system

Input Streams:

Name	Multi-read	Description
------	------------	-------------

Add... Edit... Delete...

Input streams are an alternative way of providing data to a stored process at run time. Input streams usually consist of XML data that originates in a remote application, such as a Web service. [More \(F1\)...](#)

< Back Next > Finish Cancel



3 - Use EG to Create and Test SP

Step 7: Summary (default)

Create New SAS Stored Process Wizard

7 of 7 Summary

Stored Process

Name: Score Report

Stored process execution server:

Name: SASMain - Logical Stored Process Server
Type: LogicalServer Server
Version: 9.1

SAS source code storage:

Location: Source code repository on SASMAIN
Location description:
SAS code filename: Score_Report.sas

☒ Run stored process when finished

[Copy to clipboard](#)

Review the information in the text box. To change a value, click Back.

[More \(F1\)...](#)

[< Back](#) [Next >](#) [Finish](#) [Cancel](#)



4 - Stored Process in Enterprise Guide (UI)

Score Report [X]

General

name_selected John Xu [v]

[Empty list box with up/down arrows]

Run Cancel



4 - Stored Process in Enterprise Guide (UI)

Stored Process in Enterprise Guide (Output)

The screenshot displays the SAS Enterprise Guide interface for a project named 'Project_IASUG'. The 'Project Explorer' on the left shows a hierarchy: Project_IASUG > Process Flow > IASUG_test > HTML - IASUG_test > Score Report. The main window, titled 'HTML - Score Report', shows the SAS logo and 'Enterprise Guide' text. Below this, the heading 'Score for Jack' is displayed. A table with the following data is shown:

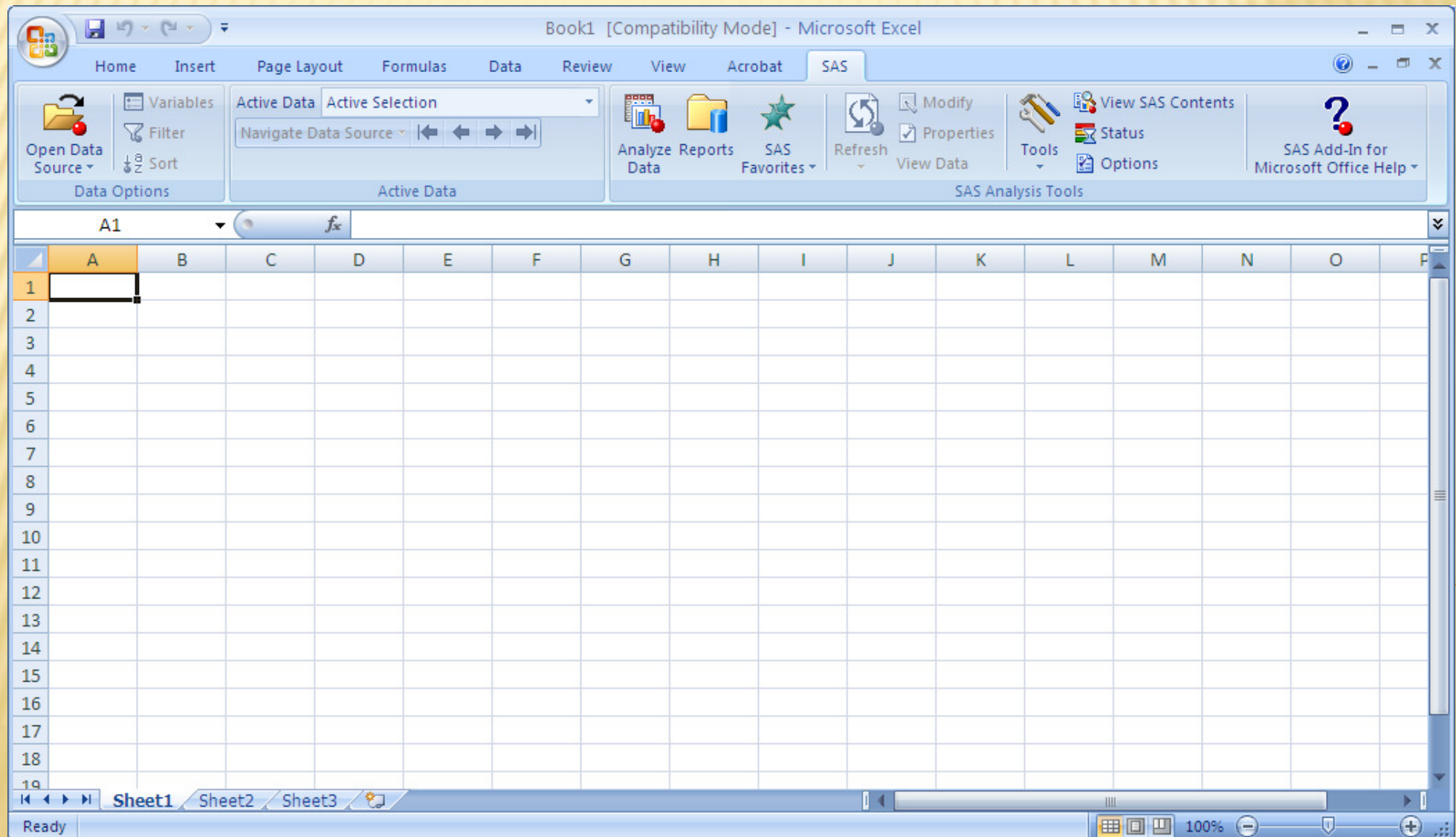
Obs	name	score
2	Jack	90

The status bar at the bottom indicates the user 'xujohn as John Xu' is connected to 'sasprod1.wellsfargo.com:8561/Foundation'.



4 – Use Stored Process in MS Excel

Stored Process in MS Excel (Requires SAS Add-in for MS Office)





4 – Use Stored Process in MS Excel

Stored Process in MS Excel (Requires SAS Add-in for MS Office)

Score Report

General

name_selected *

Pick the name from dropdown box (* denotes required parameter)



4 – Use Stored Process in MS Excel

Stored Process in MS Excel (Requires SAS Add-in for MS Office)

Choose Location

Specify the location for the results of "Score Report":

☒ Existing worksheet

Start at:
(Examples: \$A\$1 or A1)

☐ New worksheet

Named:

☐ New workbook

[Why can't I specify the location by clicking in a worksheet?](#)

OK Cancel

Book1 [Compatibility Mode]

Home Insert Page Layout Formulas Data Review View Macros SAS

Open Data Source Variables Filter Sort Data Options Active Data Active Selection Navigate Data Source

	A	B	C	D	E	F
1	Score for John					
2						
3	Obs	name	score			
4	1	John	80			
5						
6						

Sheet1 Sheet2 Sheet3

Ready 100%



Part II: Development Issues & Solution

Requirements:

- User level security
- Avoid duplicate requests
- Big job handling
- Need for a job queue
- Limited the total number of the jobs.
- Allow High Priority job
- Allow Nightly job
- Support Development/Testing/UAT/Production.



Design: Application in 3 Parts

- (1) Stored process for application parameter creation. Collect user inputs, validate it and save it to a SAS dataset on server. A parameter Report is generated for user review.
 - (2) A Stored process to submit job or check job status.
 - (3) A KSH program on server that running on the background. It will wake up once per minute (Or at any desired time interval) to check if there is any new job in the queue.
-



Design: Development Life Cycle Support

- (1) Data is saved in the common directory.
- (2) Directories for Development, Testing, UAT, and Production are created to store corresponding SAS codes.
- (3) Stored Process GUI provides selection of Development, Testing, UAT, and Production by end user.
- (4) User access is controlled by a SAS data set on the server. If user does not have proper access right, it is defaulted to Production.



Sample KSH Code

```
while true
```

```
do
```

```
  dat=$(date +"%Y-%m-%d:%H:%M")
```

```
  hour=`date +"%H"`
```

```
  hour=$(( $hour - $night_start ))
```

```
  if [[ $hour -le 0 ]]; then
```

```
    hour=$(( $hour + 24 ))
```

```
  fi
```

```
  if [[ -r $lckfile ]]; then
```

```
    job_count=`ps -ef |grep "title XXXXXX__ui" |wc -l`
```

```
    job_count=$(( job_count-1 ))
```




Sample KSH code

```
if [[ $job_count -ge $maxjob ]]; then
    echo "There are $job_count XXXXXX jobs running. Server
    busy at $dat."
else
    job_id=`cat $lckfile`
    echo "Starting XXXXXX Job at $dat for job_id=$job_id  ...."
    sas $proj_dir/code/hz_main.sas -title XXXXXX__ui \
        -log $proj_dir/work/hz_main.log -set job_id $job_id &
fi
else
    echo "No Horizon job to run at $dat"
fi

sleep 60
done
```




Questions?

Contact Information:

John Xu

1ST Consulting LLC
4652 Valley View Ln
West Des Moines, IA 50265

johnxu@1st-consulting.com
(515) -778-4093

Thanks!