

Automated Error Reporting for SAS Enterprise Guide Scheduled Projects

Bjorn Leiren
Optimization Analyst
Enbridge Pipelines Inc.
October 12, 2007



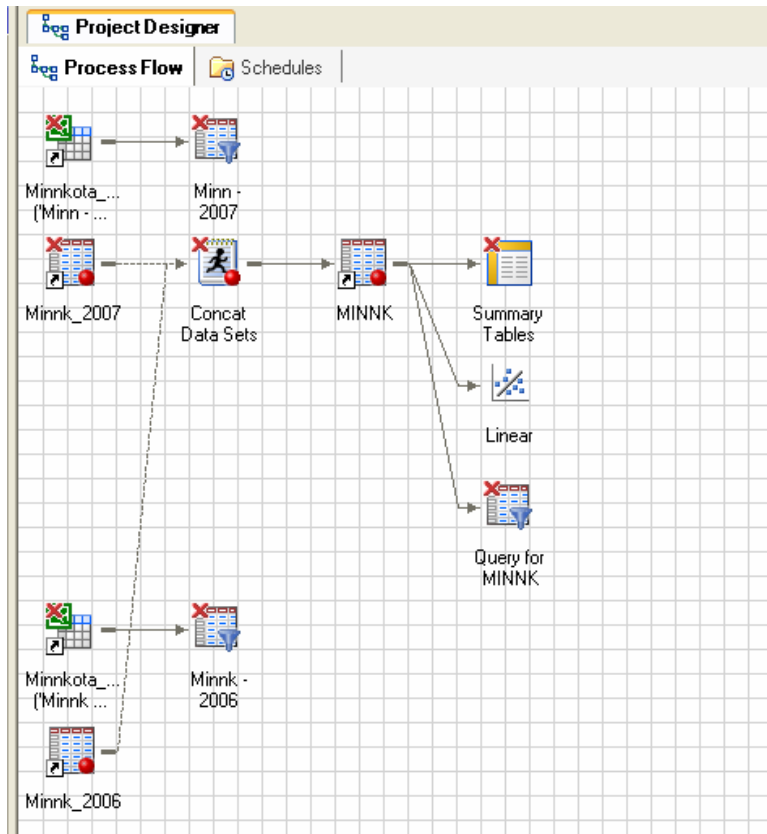
Overview

- Types of Scheduled Tasks:
 - Reports, Data Updates, Support for Business Processes, etc.

- Programs Can Fail
 - Run Manually - Check Process Flow & SAS Log
 - What about daily, weekly, or monthly tasks?

Identifying Errors Manually

□ Project Process Flow



□ Examine the SAS Log

■ IE: Query for Minnk

18 Data Minnk ;

19

20 Set Minnk_2006 (in = i1)

21 Minnk_2007 (in = i2) ;

ERROR: File WORK.MINNK_2006.DATA does not exist.

ERROR: File WORK.MINNK_2007.DATA does not exist.

22

23 By Year F3 ;

□ Problem:

Time Consuming!



Why Automated Error Reporting?

- Why Not?
- Failed Tasks result in lost time as Data and Reports are not ready for necessary functions
- Failed tasks that go unnoticed are particularly costly...

History

- Originally Conceived and Implemented at Federated Co-operatives Ltd. in Saskatoon

- Mainframe SAS
 - Built-in Error Reporting Mechanism

- Switch to Base SAS for PC run on a Server
 - Error Reporting Required
 - **Bad Idea:** End-of-Job Email Macro using &SYSERR
 - **New Idea:** Check SAS Log with a program

Base SAS vs. SAS Enterprise Guide

□ Base SAS

- Scheduled Tasks run via a .bat file
- SAS Log sent to common directory

"C:\Program Files Program Files\SAS\SAS 9.1\sas.exe"

-SYSIN "D:\Homedir\common\Ori170\Ori170xx.sas"

-LOG "D:\Homedir\common\OrSasLogFiles\Ori170xx.log"

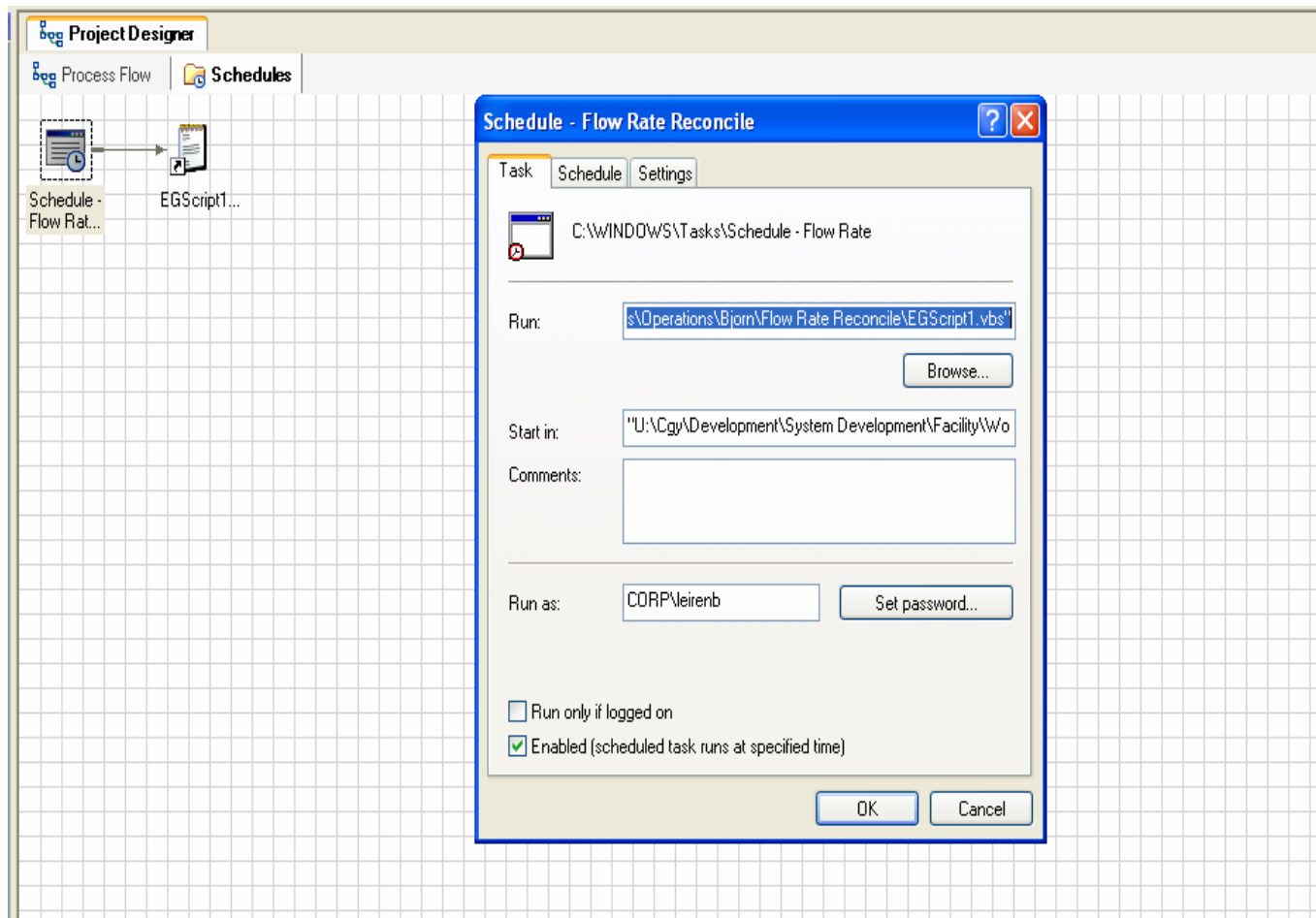
□ SAS Enterprise Guide

- .vbs File
- No SAS Log for project

Adapting Concept for SAS EG

- Obtaining an equivalent to a Project SAS Log
 - Each object within a SAS EG Process Flow stores a log property separately
 - Scheduled SAS EG Projects create a Visual Basic Scripting (.vbs) file
 - Using the same language we compile each log element into a single file

Scheduling a Project



Option Explicit

Dim app

Call dowork

'shut down the app

If not (app Is Nothing) Then

 app.Quit

 Set app = Nothing

End If

Sub dowork()

 On Error Resume Next

 '-----

 ' Start up Enterprise Guide using the project name

 '-----

 Dim prjName

 Dim prjObject

 prjName = "U:\Cgy\Development\System
 Development\Facility\Working Files\Operations\Bjorn\Flow
 Rate Reconcile\Flow Rate Reconcile.egp" 'Project Name

 Set app = CreateObject("SASEGObjectModel.Application.4")

 If Checkerror("CreateObject") = True Then

 Exit Sub

 End If

 '-----

 ' open the project

 '-----

 Set prjObject = app.Open(prjName,"")

 If Checkerror("app.Open") = True Then

 Exit Sub

 End If

 '-----

 ' run the project

 '-----

 prjObject.run

 If Checkerror("Project.run") = True Then

 Exit Sub

 End If

 '-----

 ' Save the new project

 '-----

 prjObject.Save

 If Checkerror("Project.Save") = True Then

 Exit Sub

 End If

 '-----

 ' Close the project

 '-----

 prjObject.Close

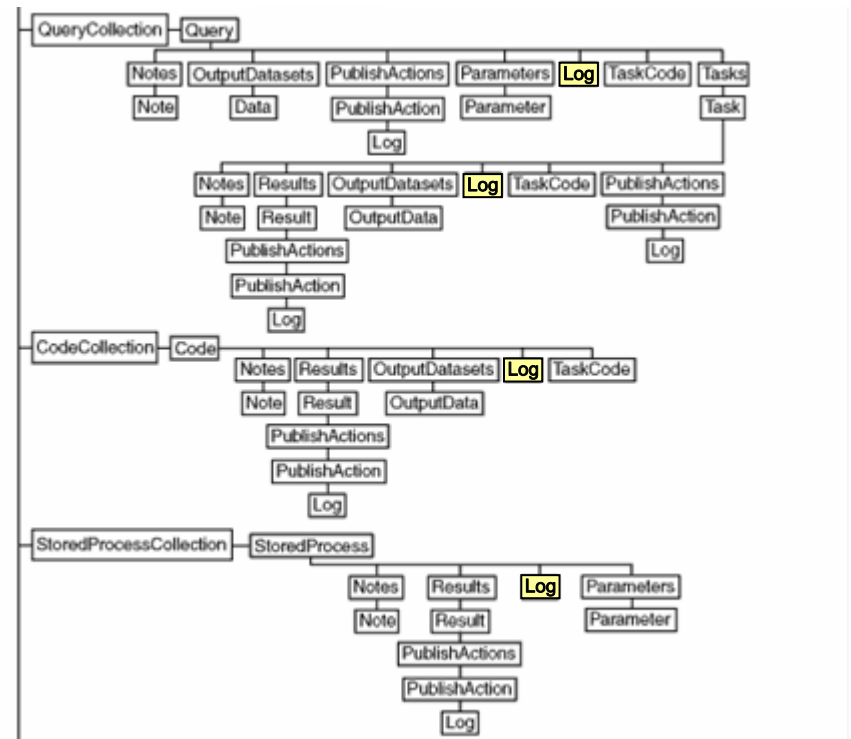
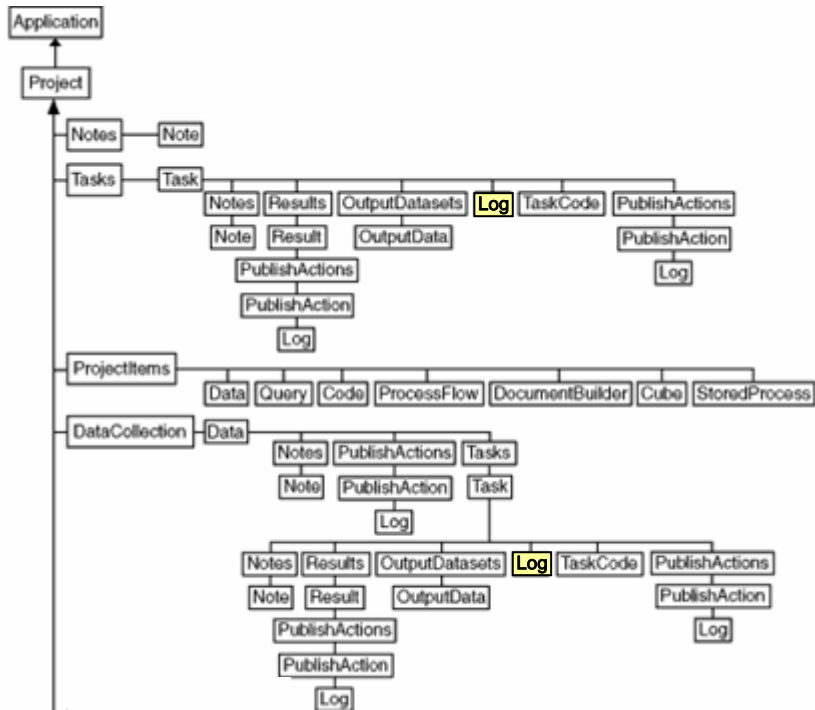
 If Checkerror("Project.Close") = True Then

 Exit Sub

 End If

End Sub

SAS EG Object Hierarchy



Option Explicit

Dim app

Sub Include(sInstFile)

Dim oFSO, f, s

Set oFSO =

CreateObject("Scripting.FileSystemObject")

Set f = oFSO.OpenTextFile(sInstFile)

s = f.ReadAll

f.Close

ExecuteGlobal s

End Sub

Include ("U:\Cgy\Development\System
Development\Facility\Working
Files\Operations\Bjorn\FacManSASLog\
SasLog.vbs")

Call dowork

1. Create Sub "Include" to execute code in external file
2. .vbs code in External File creates Sub "SASLogSave"
3. Call SASLogSave before closing project
4. Pass Two Arguments to the Sub: Project Object & Path for saving SAS Log file

'-----

' Save the new project

'-----

prjObject.Save

If Checkerror("Project.Save") = True Then

Exit Sub

End If

Call SASLogSave (prjObject,
"U:\Cgy\Development\System
Development\Facility\Working
Files\Operations\Bjorn\FacManSASLog\
" & prjObject.name & ".log")

'-----

' Close the project

'-----

prjObject.Close

If Checkerror("Project.Close") = True Then

Exit Sub

End If

SASLogSave Sub-Routine (.vbs)

Sub SASLogSave(ByVal ProjectObj, LogPath)

```
Dim filesystem, filetxt
Dim objProjectItems
Dim d, n, DC
Dim ObjCollection
Dim DataCollection
Dim DataCollectionCount

Const ForReading = 1, ForWriting = 2, ForAppending = 8
Set objProjectItems = ProjectObj.ProjectItems
Set filesystem = CreateObject("Scripting.FileSystemObject")

Set filetxt = filesystem.CreateTextFile(LogPath, True)
filetxt.WriteLine ("!!! Project Log:" & ProjectObj.Path & " $$$ Project
Name:" & ProjectObj.Name & " ### Project DateTime:" & Date
& " " & Time )
filetxt.Close

n = objProjectItems.Count

Set ObjCollection = ProjectObj.Tasks
Call WriteLogForEach(ObjCollection, LogPath , ForAppending )
```

Set DataCollection = ProjectObj.DataCollection

```
DataCollectionCount = DataCollection.Count

DC = 0
Do While DC < DataCollectionCount
Set ObjCollection = DataCollection(DC).Tasks
Call WriteLogForEach(ObjCollection, LogPath , ForAppending )
DC = DC + 1
Loop
```

Set ObjCollection = ProjectObj.QueryCollection

```
Call WriteLogForEach(ObjCollection, LogPath , ForAppending )
```

Set ObjCollection = ProjectObj.CodeCollection

```
Call WriteLogForEach(ObjCollection, LogPath , ForAppending )
```

Set ObjCollection = ProjectObj.StoredProcessCollection

```
Call WriteLogForEach(ObjCollection, LogPath , ForAppending )
```

End Sub

WriteLogForEach Sub-Routine (.vbs)

```
Sub WriteLogForEach(ObjColltn, LogPathX ,  
    WriteTypeX )  
    Dim filetxt , FileSysX  
    Set fileSysX = CreateObject("Scripting.FileSystemObject")  
  
    Dim ObjCollectionCount  
    Dim objLog  
    Dim ObjectName  
    Dim ObjectType  
    Dim ObjLogText  
  
    ObjCollectionCount = ObjColltn.Count  
  
For Each ObjectD In ObjColltn  
  
    Set objLog = ObjectD.log
```

If not (objLog is nothing) then

```
    ObjectName = ObjectD.name  
    ObjectType = ObjectD.type  
    ObjLogText = objLog.text
```

```
    Set filetxt = fileSysX.OpenTextFile(LogPathX,  
        WriteTypeX, True)
```

```
    filetxt.WriteLine ("*** Object Log: " & ObjectName & "  
        xxx  ObjectType=" & ObjectType)
```

```
    filetxt.Write (ObjLogText)
```

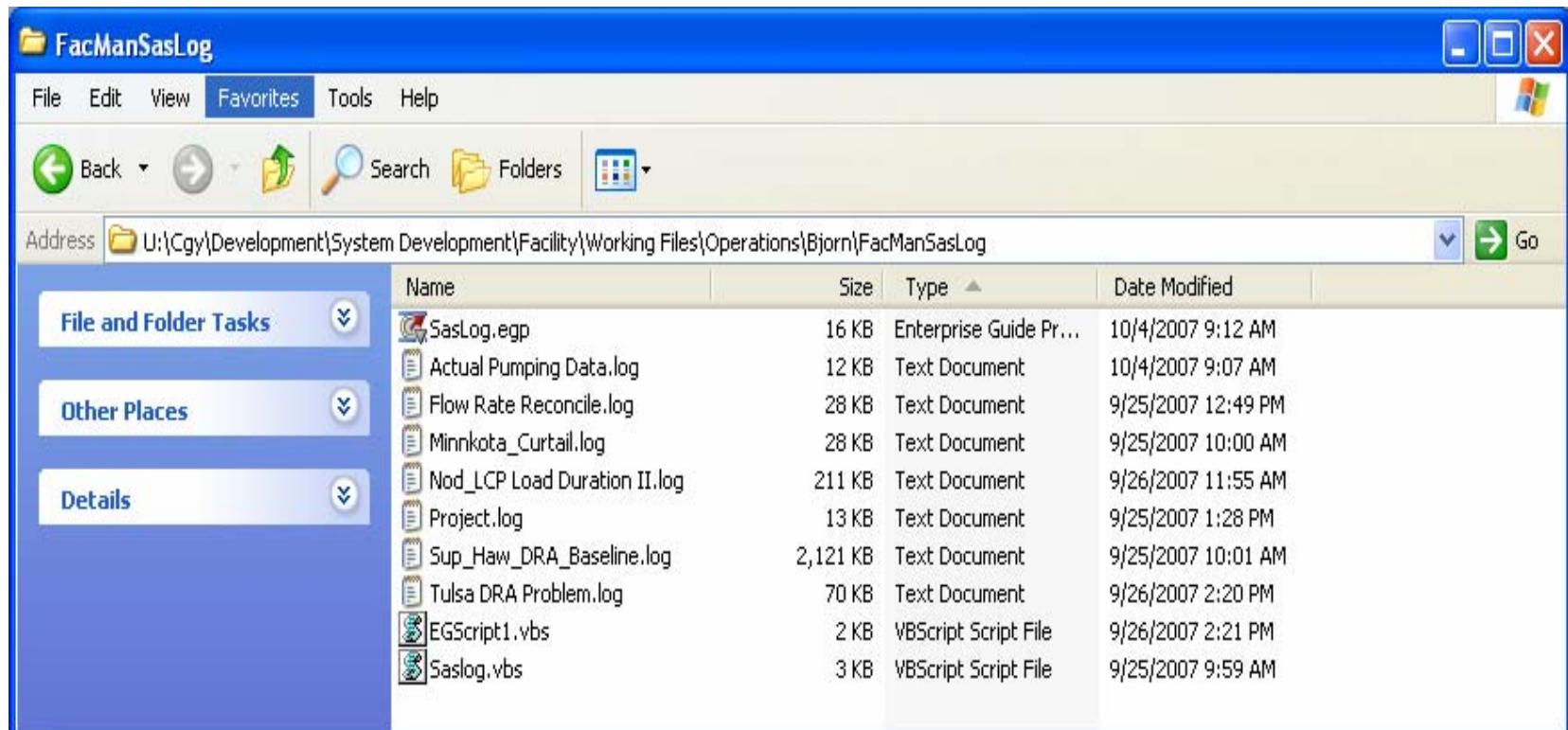
```
    filetxt.Close
```

```
    End If
```

```
Next
```

```
End Sub
```

Log Files in a Common Directory



The screenshot shows a Windows Explorer window titled "FacManSasLog". The address bar displays the path: U:\Cgy\Development\System Development\Facility\Working Files\Operations\Bjorn\FacManSasLog. The main pane shows a list of files with columns for Name, Size, Type, and Date Modified. The files listed are:

Name	Size	Type	Date Modified
SasLog.egp	16 KB	Enterprise Guide Pr...	10/4/2007 9:12 AM
Actual Pumping Data.log	12 KB	Text Document	10/4/2007 9:07 AM
Flow Rate Reconcile.log	28 KB	Text Document	9/25/2007 12:49 PM
Minnkota_Curtail.log	28 KB	Text Document	9/25/2007 10:00 AM
Nod_LCP Load Duration II.log	211 KB	Text Document	9/26/2007 11:55 AM
Project.log	13 KB	Text Document	9/25/2007 1:28 PM
Sup_Haw_DRA_Baseline.log	2,121 KB	Text Document	9/25/2007 10:01 AM
Tulsa DRA Problem.log	70 KB	Text Document	9/26/2007 2:20 PM
EGScript1.vbs	2 KB	VBScript Script File	9/26/2007 2:21 PM
Saslog.vbs	3 KB	VBScript Script File	9/25/2007 9:59 AM



What's Next?

- Using a SAS EG Project we iterate through each log file in the common directory
- Each line is inspected for key words known to indicate Errors
- Email generated with a daily Error Report
- Implemented with a single SAS Code step in the Project

Reading The Log Files

```
%Let LogDir = \\Cgas107\common\cgy\Development\ .... \FacManSasLog ;  
filename exdir %unquote(%str('%&LogDir%'));
```

```
%macro readdir ;
```

```
  %let dirid = %sysfunc(DOPEN(exdir));  
  %put dirid =&dirid;  
  %let dirent = %sysfunc(DNUM(&dirid));  
  %put dirent = &DIRENT;
```

```
%do i = 1 %to &dirent;
```

```
  %let dirread = %sysfunc(DREAD(&dirid,&i));  
  %put dirread = &dirread;
```

```
Data LogContents ;
```

```
  Length FileNameX $50. ;  
  Infile "&LogDir\&dirread" ;  
  If Index ("&dirread", ".log") > 0 then do ;
```

```
    Input ;
```

```
    InputRecordX = _infile_ ;
```

```
    FileNameX = "&dirread" ;
```

```
    Output ;
```

```
  End;
```

```
Run ;
```

```
%if &i=1 %then %do;
```

```
  data AllLogContents ;  
  set LogContents;  
  run;
```

```
%end;
```

```
%else %do;
```

```
  proc append base=AllLogContents data=LogContents;  
  run;
```

```
%end;
```

```
%end;
```

```
  %let rc = %sysfunc(DCLOSE(&dirid));  
  %put rc = &rc;
```

```
%mend;
```

Finding SAS Errors in Log Files

- We have Obtained a Dataset with one observation for each line in the Log Files
- How do we extract SAS Errors For Reporting ?

```
If ( Trim ( Substr ( Uppcase ( InputRecordX ) , 1 , 5 ) ) = "ERROR" ) Then Do ;  
    ErrIndex = ErrIndex + 1 ;  
    ErrMsg = Trim ( Substr( InputRecordX , 7 , ( Length ( InputRecordX ) - 7 ) ) )  
    Call Symput ( 'SubjErr ' , " !!! Error Detected !!! " ) ;  
    If ErrIndex < 6 Then Output ;  
End ;
```



Other Features:

- The Visual Basic Scripts insert fields into the Log Files identifying:
 - Project Name
 - Object Name & Type (Task, Query, etc.)
 - Run Date & Time
(Code Omitted)

- This information is extracted from the text and included in the Error Report

- SAS Error <message> statement

Other Features (Cont'd):

□ Object Type is Returned as an Integer

Proc Format ;

Value \$ ObjType

'0' = 'Log'

'1' = 'Code'

'2' = 'Data'

'3' = 'Query'

'4' = 'ProcessFlow'

'5' = 'DocBuilder'

'6' = 'Note'

'7' = 'Result'

'8' = 'Task'

'9' = 'TaskCode'

'10' = 'QueryParameter'

'11' = 'Output Data'

'12' = 'Stored Process'

'13' = 'Stored Process Parameter'

'14' = 'PublishAction'

'15' = 'Cube'

'16' = 'CubeSlice'

'17' = 'CubeDetail' ;

Run ;

Results

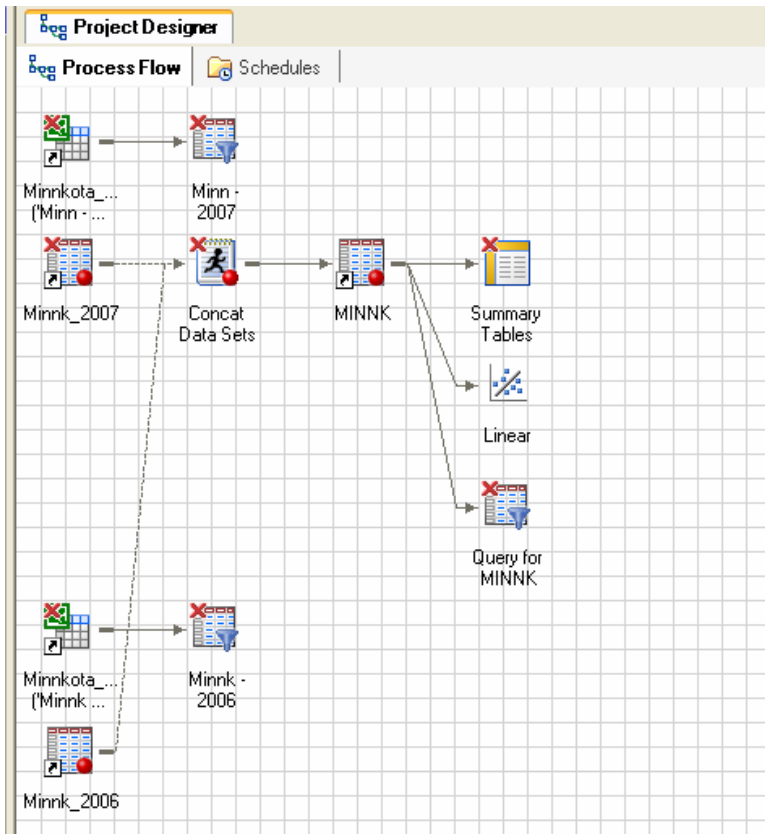
□ Program Generates a Daily Email with a list of Errors

FacMan SAS Server Error Report
Error Report Ran 09:43 on 09OCT07

Project Name	Object	Error Msg	Time	Date	Day
ACTUAL PUMPING DATA			9:07	04OCT2007	Thu
FLOW RATE RECONCILE			2:00	08OCT2007	Mon
MINNKOTA_CURTAIL			10:00	25SEP2007	Tue
001.1 Task	SUMMARY TABLE	The type of name (Year) is unknown			
003.1 Query	MINN - 2007	Unable to get SAS code. Error opening data "Minnkota_Curtail.xls			
004.1 Query	MINNK - 2006	Unable to get SAS code. Error opening data "Minnkota_Curtail.xls			
005.1 Query	QUERY FOR MIN	Column F3 could not be found in the table/view identified with t			
005.2 Query	QUERY FOR MIN	Column 'MK1-PWR#allowhp'n could not be found in the table/view i			
005.3 Query	QUERY FOR MIN	Column 'MK1-PWR#hp'n could not be found in the table/view identi			
005.4 Query	QUERY FOR MIN	Column F6 could not be found in the table/view identified with t			
005.5 Query	QUERY FOR MIN	Column True could not be found in the table/view identified with			
006.1 Code	CONCAT DATA S	File WORK.MINNK_2006.DATA does not exist			
006.2 Code	CONCAT DATA S	File WORK.MINNK_2007.DATA does not exist			
NOD_LCP LOAD DURATION II			11:55	26SEP2007	Wed
017.1 Task	LINE PLOT	Unable to get SAS code. The task could not be loaded due to an u			
PROJECT			13:28	25SEP2007	Tue
SUP_HAW_DRA_BASELINE			10:01	25SEP2007	Tue
TULSA DRA PROBLEM			14:20	26SEP2007	Wed

Investigate Errors

□ Project Process Flow



□ Examine the SAS Log

■ IE: Query for Minnk

18 Data Minnk ;

19

20 Set Minnk_2006 (in = i1)

21 Minnk_2007 (in = i2) ;

ERROR: File WORK.MINNK_2006.DATA does not exist.

ERROR: File WORK.MINNK_2007.DATA does not exist.

22

24 By Year F3 ;

Questions

