ABSTRACT
SAS/BP was written to ease the task of Support personnel running SAS Batch Jobs in a Production environment. It is also, from an end-user perspective, an indispensable tool for quickly retrieving SAS Outputs, viewing and analyzing Elapse times, Failures, Metrics, Browsing SAS Tables and Attributes, Volume trends etc. Using a simple Web interface, built with SAS/IntrNet, a quick glance at the Status window gives a complete and detailed view on the status of the Production SAS Batch Jobs.

INTRODUCTION
Facing the classic case of the ‘W5’?

W1 - What is the status of my Production Cycle?
W2 - Why did my Job(s) fail?
W3 - When did this happen?
W4 - Where can I find the SAS Logs, Lists, ODS, Programs, Scripts, Macro?
W5 - Who has met their completion commitment?

If you answered yes to at least one of these W5, then this paper is for you.

SYSTEM ARCHITECTURE
A QUICK GLANCE AT SAS/BP REALTIME STATUS WINDOW

Figure 1 - The detailed SAS/IntrNet Web interface displays the overall status of Production SAS Batch Jobs. More details are available where hyperlinks exist.

Let us focus on the 'DEMO Environment'. In this Environment we have a total of 1 Component (SUGI 30). The Component usually reflects the Business functions within your Application. They are 4 distinct Processes associated with this Business function (Sample Job 1, 2, 3, 4, for SUGI Demo). These are the actual SAS Batch Job names submitted by yourself, or, your Scheduler.

The 'Detailed Status' column shows various status codes based on the Completion code of your Process. Status can vary from, 'Active, Completed, Failed' and can be followed by a combination of 'Error, Warning' or 'Err/Warn'.

The 'Last Cycle ~ Completion' column is self explanatory. The optional 'Cycle Date' is available to quickly track your batch Jobs based on Cycles.

The 'Elaps ~ Avg' column displays the Elapse time for the last execution of the Process, followed by, the Average Elapse time based on historical data.

The 'Current View' column gives instant access to the Script, SAS Program, SAS Log/List and ODS for the Process.

The 'History View' column gives instant access to a Historical view of each Process.

The '%Met' column indicates if you have met your %Target commitments for the Processes and Environment.
A DETAILED VIEW OF THE STEPS IN A PROCESS

![Image of Detailed Status window]

**Figure 2 - The Detailed Status window shows an in-depth view for each step within a Process.**

If a Process is ‘active’, simply refresh your browser window and watch the steps progress… That's real-time!

More details are available where hyperlinks exist

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>CC</th>
<th>Rc</th>
<th>Err</th>
<th>Elapse</th>
<th>Engine</th>
<th>Library</th>
<th>Table</th>
<th>Rows</th>
<th>Var</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>This is the first Code. Step of the Job</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0:00:01</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Display a text message in the Remarks column.</td>
</tr>
<tr>
<td>2</td>
<td>This is a To Loop Code Step with 3 number of iterations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Set a subset of DATASET 1 to VAR A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Print Print of Dataset DATASET 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>First step executed since Bypass=Unspecified.</td>
</tr>
<tr>
<td>5</td>
<td>Copy a subset of DATASET 1 for Web Browser Display</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Copy DATASET 1 and Copy VAR A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Display the current Timestamp: 02/8/2001 18:22:22.344</td>
</tr>
<tr>
<td>7</td>
<td>Copy DATASET 2 to SAS/BP Share Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Copy a subset of DATASET 1 to an External File</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Print the current flow report before logging anything</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Append DATASET 1 to SASDATASET DATASET 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Process the external program listed in the observations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Output the results generated in the observations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Build an index on DATASET 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Calculate average values of hours. Sum Table in INCDATAS LIBRARY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Create new CONFIG Table in WORKSPACE Library</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Print of Member_name Sum Table in the WORKSPACE Library</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Note: The Cumulative Elapse time is 0:51:32 in which SAS/BP consumed an additional 0:01:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The ‘#' column indicates the Step sequence number in the Process.

The ‘Description’ column displays the step Description given by our Process. The hyperlink extracts an excerpt of the SASLOG for that step only.

The ‘CC RC ERR’ columns display the System Condition Code, Return Code and Error Code issued by SAS, SAS/BP, and/or the Operating System.

The ‘Elapse’ column displays the Step real Elapse time. Note that the additional Elapse time consumed by SAS/BP is excluded from this total. SAS/BP itself consumes on average 1~10 seconds per steps depending on the type of information captured.

The ‘Engine, Library, Table, Rows, Var’ columns display the contents of the Libraries used in the step. These fields are only populated when requested by the %STEPLOG Macro (explained later). The links allow you to view the information about the Library, Columns and even browse the Dataset.

The ‘Remarks’ column is a free format field. Any type of information that you wish to pass from your step into this field is possible. SAS/BP will override this field when warrant.

Finally, the link at the bottom allows the information displayed to be directly exported to an excel spreadsheet.
AN ALTERNATE VIEW OF THE STEPS IN A PROCESS

This view is an alternative to Figure #2 and shows how your program steps have evolved in time. When steps are altered in your Program (i.e. renamed, added, moved or removed), their historical placement are retained throughout time.

In this example, the step named ‘Build an index on Dataset_2’ was last run as Step#11 on Jan28. When the step named ‘Verify if the Append procedure loaded all the observation’ was added, the old Step#11 became Step #12. This kind of tracking is vital when we want to produce statistics by Steps. For example, the Elapse time for Step#11 after Jan28 may increase or decrease if the newly inserted step adds observations to the Dataset in step#12.

Remember that SAS/BP is a ‘Data Collection’ Application for your SAS Jobs. All the information about a process is retained in time. Reporting possibilities are endless when the data is captured in a consistent way.

The ‘Elapse and Trend’ columns have a link to Graphical displays (SAS/Graph shown in Figure 5A-5B).
THE HISTORICAL VIEW OF A PROCESS

Figure 4 - The Historical window displays historical executions of a Process. Further details are available where hyperlinks exist.

The ‘Cycle Date’ column indicates the Cycle Date associated to the Process. The concept of Cycle Date is an efficient way of tracking the entire Environment. For Data Quality issues, it can be interesting to generate reports based on Cycle Dates and further analyze critical paths in the Application.

The ‘PID#’ column indicates the Operating System Process Id number (Window, Unix, MVS).

The ‘MODE’ column indicates the type of execution submitted. Possible values are: Normal, Restart, or, Rerun.

The remaining columns are self explanatory and provide further links.
VIEWING SOME METRICS VIA SAS/GRAPH

Figure 5A – Graphical display of Elapse Time for a specific Job Step

Figure 5B – Graphical display of the Nb of Observations processed in a specific Job Step

Dynamic SAS/Graphs are generated via SAS/IntrNet and returned to your Web Browser. It’s that simple!
VIEWING VARIABLE ATTRIBUTES FOR A TABLE

Figure 6 - The Variable Attributes window displays information about attributes in a Table

This window is self explanatory. It is interesting to note that all this information is retained in SAS/BP for each execution of a Process. It then becomes possible to track down Variable Attributes that have been altered in time. SAS/BP could be viewed as a ‘Metadata’ of your SAS Batch Jobs.
WHAT IS REQUIRED TO RUN SAS/BP?

At the heart of the System is **SAS/Share**. This is the only additional requirement to your SAS Base System. **SAS/IntrNet** is an optional component should you want to View your Batch Jobs through a Web Interface. Alternatively, you can develop your own ASP or Java interface using an ODBC connection to SAS/Share.

HOW CAN I MAKE IT WORK FOR ME?

All you need is to do is add the SAS/BP Macro `%STEPLOG` after the Data step, or, Proc step that you want to track. You must also standardize your Application by defining an Environment, a Component and a Process. If you already have a well structured Production environment, then implementing SAS/BP will require minimal efforts. If you don’t, then do yourself a favor and get organized!

They are 3 basic Scripts required to submit your SAS Batch Job.

1. The Operating System script that Submits your SAS Job (i.e: .bat, cmd, .sh) You should already have that!
2. The ‘**RUN**’ Script. This script holds Libnames, Filenames and Macro variables that define your environment. It should not contain any SAS Programs or Procs.
3. The **`SAS Program`** that you want to run

A SAMPLE ‘**RUN**’ SCRIPT

```sas
```

Note that any Job or Step failure will automatically send a detailed Email notification to the Support Team. Optionally, you can request that a PAGE be sent via the Pager=Y parameter of the RUN Script.
A SAMPLE ‘SAS PROGRAM’

The intent of this sample program is to show the use of Macro `%STEPLOG`.
The program logic and/or errors are irrelevant.

```sas
Data _Null_;  
Call Symput('DISPLAY_RMKS','Display a text message in the Remarks column');  
Run;  
%STEPLOG(Step=This is the first Data Step of the Job, Parm=&DISPLAY_RMKS);  

Data DATASET_1 (Drop = I Label='Label info for Dataset_1');  
Attrib  
  Var_A    Length=8. Format=8. Informat=8. Label = "This is the Label description for Var_A"  
  Var_B    Length=8. Format=8. Informat=5. Label = "This is the Label description for Var_B"  
  Var_C    Length=8. Format=8. Informat=3. Label = "This is the Label description for Var_C"  
  Var_D    Length=8. Format=8. Informat=8. Label = "This is the Label description for Var_D"  
  Var_E    Length=8. Format=DATETIME19.2 Informat=DATETIME19.2 Label = "This is the Label description for Var_E"  
  Var_F    Length=8. Format=ZD8. Informat=ZD8. Label = "This is the Label description for Var_F" ;  
Do I = 1 to 1933519 ;  
  Var_A = I + I ;  
  Var_B = I * I ;  
  Var_C = Var_B - I  ;  
  Var_D = Var_B * I  ;  
  Var_E = TODAY() + I ;  
  Var_F = I ;  
  Output;  
End;  
Run;  
%STEPLOG(Step=This is a Do Loop Data Step with X number of iterations, Libref=WORK, Mem=DATASET_1, Col=Y, NOBS=OBS_DATASET_1);  

Proc Sort Data = DATASET_1 (Obs=247101)  
  Out = DATASET_2 (Label='Label of Sorted Dataset 2'  
    Index=(Var_A Index_1=(Var_A Var_B) / Unique Var_C / Nomiss)  
    Compress=Yes);  
By Var_A ;  
Run;  
%STEPLOG(Step=Sort a subset of DATASET_1 by VAR_A, Libref=WORK, Mem=DATASET_2, Col=Y, NOBS=OBS_DATASET_2);  

ODS HTML BODY=ODSOUT ;  
Proc Print Data = DATASET_2(Obs = 501) LabelW N Width = Min;  
  Title "Print 501 of &OBS_DATASET_2 Observations from Dataset_2";  
Run;  
ODS HTML CLOSE ;  
%STEPLOG(Step=Proc Print of Dataset DATASET_2, Bypass=Warning);  

Data SASDATA.DATASET_3 (Drop=Var_A Var_B Var_E);  
  Set DATASET_1(Obs=16150) ;  
Run;  
%STEPLOG(Step=Copy a subset of DATASET_1 for Web Browser  
  Display, Libref=SASDATA, Mem=DATASET_3, Col=Y);  

/* Author: Bernard Nicoll, CGI, Jan 28th, 2005  
   Environment: DEMO  
   Component: SUGI 30  
   Process: Sample Job 1 for SUGI Demo  
   Modified by Description  
*/  
*-------------------------------------------------------------------------*/
/*                                                                         */
/*      Author: Bernard Nicoll, CGI, Jan 28th, 2005                        */
/*                                                                         */
/* Environment: DEMO                                                       */
/* Component: SUGI 30                                                      */
/* Process: Sample Job 1 for SUGI Demo                                    */
/*                                                                         */
*-------------------------------------------------------------------------*/
```

Data _Null_;  
Call Symput('DISPLAY_RMKS','Display a text message in the Remarks column');  
Run;  
%STEPLOG(Step=This is the first Data Step of the Job, Parm=&DISPLAY_RMKS);  

Data DATASET_1 (Drop = I Label='Label info for Dataset_1');  
Attrib  
  Var_A    Length=8. Format=8. Informat=8. Label = "This is the Label description for Var_A"  
  Var_B    Length=8. Format=8. Informat=5. Label = "This is the Label description for Var_B"  
  Var_C    Length=8. Format=8. Informat=3. Label = "This is the Label description for Var_C"  
  Var_D    Length=8. Format=8. Informat=8. Label = "This is the Label description for Var_D"  
  Var_E    Length=8. Format=DATETIME19.2 Informat=DATETIME19.2 Label = "This is the Label description for Var_E"  
  Var_F    Length=8. Format=ZD8. Informat=ZD8. Label = "This is the Label description for Var_F" ;  
Do I = 1 to 1933519 ;  
  Var_A = I + I ;  
  Var_B = I * I ;  
  Var_C = Var_B - I  ;  
  Var_D = Var_B * I  ;  
  Var_E = TODAY() + I ;  
  Var_F = I ;  
  Output;  
End;  
Run;  
%STEPLOG(Step=This is a Do Loop Data Step with X number of iterations, Libref=WORK, Mem=DATASET_1, Col=Y, NOBS=OBS_DATASET_1);  

Proc Sort Data = DATASET_1 (Obs=247101)  
  Out = DATASET_2 (Label='Label of Sorted Dataset 2'  
    Index=(Var_A Index_1=(Var_A Var_B) / Unique Var_C / Nomiss)  
    Compress=Yes);  
By Var_A ;  
Run;  
%STEPLOG(Step=Sort a subset of DATASET_1 by VAR_A, Libref=WORK, Mem=DATASET_2, Col=Y, NOBS=OBS_DATASET_2);  

ODS HTML BODY=ODSOUT ;  
Proc Print Data = DATASET_2(Obs = 501) LabelW N Width = Min;  
  Title "Print 501 of &OBS_DATASET_2 Observations from Dataset_2";  
Run;  
ODS HTML CLOSE ;  
%STEPLOG(Step=Proc Print of Dataset DATASET_2, Bypass=Warning);  

Data SASDATA.DATASET_3 (Drop=Var_A Var_B Var_E);  
  Set DATASET_1(Obs=16150) ;  
Run;  
%STEPLOG(Step=Copy a subset of DATASET_1 for Web Browser  
  Display, Libref=SASDATA, Mem=DATASET_3, Col=Y);  

/* Author: Bernard Nicoll, CGI, Jan 28th, 2005  
   Environment: DEMO                                                       */
/* Component: SUGI 30                                                      */
/* Process: Sample Job 1 for SUGI Demo                                    */
/*                                                                         */
*/ **-------------------------------------------------------------------------*/
/*                                                                         */
/*      Author: Bernard Nicoll, CGI, Jan 28th, 2005                        */
/*                                                                         */
/* Environment: DEMO                                                       */
/* Component: SUGI 30                                                      */
/* Process: Sample Job 1 for SUGI Demo                                    */
/*                                                                         */
*-------------------------------------------------------------------------*/
```

Data _Null_;  
Call Symput('DISPLAY_RMKS','Display a text message in the Remarks column');  
Run;  
%STEPLOG(Step=This is the first Data Step of the Job, Parm=&DISPLAY_RMKS);  

Data DATASET_1 (Drop = I Label='Label info for Dataset_1');  
Attrib  
  Var_A    Length=8. Format=8. Informat=8. Label = "This is the Label description for Var_A"  
  Var_B    Length=8. Format=8. Informat=5. Label = "This is the Label description for Var_B"  
  Var_C    Length=8. Format=8. Informat=3. Label = "This is the Label description for Var_C"  
  Var_D    Length=8. Format=8. Informat=8. Label = "This is the Label description for Var_D"  
  Var_E    Length=8. Format=DATETIME19.2 Informat=DATETIME19.2 Label = "This is the Label description for Var_E"  
  Var_F    Length=8. Format=ZD8. Informat=ZD8. Label = "This is the Label description for Var_F" ;  
Do I = 1 to 1933519 ;  
  Var_A = I + I ;  
  Var_B = I * I ;  
  Var_C = Var_B - I  ;  
  Var_D = Var_B * I  ;  
  Var_E = TODAY() + I ;  
  Var_F = I ;  
  Output;  
End;  
Run;  
%STEPLOG(Step=This is a Do Loop Data Step with X number of iterations, Libref=WORK, Mem=DATASET_1, Col=Y, NOBS=OBS_DATASET_1);  

Proc Sort Data = DATASET_1 (Obs=247101)  
  Out = DATASET_2 (Label='Label of Sorted Dataset 2'  
    Index=(Var_A Index_1=(Var_A Var_B) / Unique Var_C / Nomiss)  
    Compress=Yes);  
By Var_A ;  
Run;  
%STEPLOG(Step=Sort a subset of DATASET_1 by VAR_A, Libref=WORK, Mem=DATASET_2, Col=Y, NOBS=OBS_DATASET_2);  

ODS HTML BODY=ODSOUT ;  
Proc Print Data = DATASET_2(Obs = 501) LabelW N Width = Min;  
  Title "Print 501 of &OBS_DATASET_2 Observations from Dataset_2";  
Run;  
ODS HTML CLOSE ;  
%STEPLOG(Step=Proc Print of Dataset DATASET_2, Bypass=Warning);  

Data SASDATA.DATASET_3 (Drop=Var_A Var_B Var_E);  
  Set DATASET_1(Obs=16150) ;  
Run;  
%STEPLOG(Step=Copy a subset of DATASET_1 for Web Browser  
  Display, Libref=SASDATA, Mem=DATASET_3, Col=Y);  

/* Author: Bernard Nicoll, CGI, Jan 28th, 2005  
   Environment: DEMO                                                       */
/* Component: SUGI 30                                                      */
/* Process: Sample Job 1 for SUGI Demo                                    */
/*                                                                         */
*/ **-------------------------------------------------------------------------*/
/*                                                                         */
/*      Author: Bernard Nicoll, CGI, Jan 28th, 2005                        */
/*                                                                         */
/* Environment: DEMO                                                       */
/* Component: SUGI 30                                                      */
/* Process: Sample Job 1 for SUGI Demo                                    */
/*                                                                         */
*-------------------------------------------------------------------------*/
```
Data SASDATA.DATASET_4 (Drop=Var_A);
  Set DATASET_1 ;
  Call Symput('TIMESTAMP',Put(DATETIME(),DATETIME20.3));
Run;

%STEPLOG(Step=Copy DATASET_1 and Drop VAR_A,Libref=SASDATA,Mem=DATASET_4,Col=Y,Parm=Display the current Timestamp &TIMESTAMP);

Proc Copy Indd=WORK Outdd=SASBP Index=Yes ;
  Select DATASET_2 ;
Run;

%STEPLOG(Step=Copy DATASET_2 to SASBP Share Server,Libref=SASBP,Mem=DATASET_2,Col=Y);

Data _Null_; 
  Set DATASET_1(Obs=59177) End=The_End ;
  File DEMOFILE ;
  Put Var_A Var_B Var_C;
  If The_End Then Call Symput('NRECS',_N_);
Run;

%STEPLOG(Step=Copy a subset of DATASET_1 to an External File,Fileref=DEMOFILE,Ext_Recs=Y);
%STEPLOG(Step=Get the current Row count before we append anything,Libref=SASDATA,Mem=DATASET_2,NOBS=OBS_BEFORE_APPEND);

Proc Append Data = DATASET_1 Base = SASDATA.DATASET_2 FORCE ;
Run;

%STEPLOG(Step=Append DATASET_1 to SASDATA.DATASET_2,Libref=SASDATA,Mem=DATASET_2,NOBS=OBS_AFTER_APPEND);

%Compare_Nobs(Step=Verify if the Append procedure loaded all the observations, Input= &OBS_DATASET_1 + &OBS_BEFORE_APPEND, Output= &OBS_AFTER_APPEND );

Proc Datasets Library = SASDATA Nolist ;
  Modify DATASET_2 ;
  Index Delete _All_ ;
  Index Create Var_B ;
Quit;

%STEPLOG(Step=Build an index on DATASET_2,Libref=SASDATA,Mem=DATASET_2,Parm=Display the current System Date &SYSDATE);

%OS_Cmd(Cmd=copy c:\*.abc > test.txt,Bypass=Error);

Proc Datasets Library = SUGIODBC Nolist;
  Delete New_Sugi_Table;
Quit;

%STEPLOG(Step=Delete previous version of New_Sugi Table in SUGIODBC Library,Bypass=Warning);

Data SUGIODBC.New_Sugi_Table;
  Var1 = 'Just a dummy value';
Run;

%STEPLOG(Step=Create new ODBC Table in SUGIODBC Library,Libref=SUGIODBC,Mem=New_Sugi_Table);

Proc Print Data = SUGIODBC.New_Sugi_Table;
Run;

%STEPLOG(Step=Print of Member New_Sugi_Table in the ODBC Library);
USING THE %STEPLOG MACRO

The following parameters are available in macro %STEPLOG:

1. STEP=, This is the ONLY REQUIRED field, all others are optional
2. LIBREF= and MEM=, specify the Library Reference and Member name for which you wish to capture information from
3. Col=Y, to capture Variable Attributes from the Member specified by MEM=
4. BYPASS=ERROR/WARNING, to bypass a step that could has an Error or Warning
5. PARM=, is a free format field that holds additional remarks or comments about your step. You can pass Macro variables to this field.
6. NOBS=macro variable, name of a macro variable that will hold the number of Observations found in MEM=.
7. EXT_RECS=Y, if you wish to capture the number of records in an External file

CONCLUSION

SAS/BP consolidates your SAS Batch Jobs into a well structured Environment for ease of Support & Development.

In April 2004, CGI implemented SAS/BP in a Production environment on the Bell Canada SAS infrastructure. At time of publication for the SUGI 30 Proceedings, SAS/BP had proven to be extremely valuable and reliable for the Customer.

This paper was limited to the general highlights of SAS/BP. Other features and topics were not covered or shown.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Bernard Nicoll  
CGI  
1350, Blvd. Rene Levesque West  
Work Phone: 514-415-3000  
Email: Bernard.nicoll@cgi.com  
Web: www.cgi.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.