Paper 270-27

Maximizing Productivity by Automating Frequent Tasks

Robert Ellsworth, Ellsworth Stewart Consulting Inc., Markham, On., Canada

Abstract

When working in a multi-tier environment, reduced productivity can result from the repetitive and time-consuming details required to submit, view, schedule, and access jobs and files. In an effort to improve productivity, the details, steps, and work flow were built into a SAS® system that handles everyday tasks. The result is a system that supports a one, two, and three tiered environment, and enables users to produce results even though they may have little practical knowledge of the environments they are accessing.
This presentation will introduce our approach to the issues encountered in multiple environments. We will then provide our solutions to many of the issues, both conceptually and at the 'SAS-code' level.
Some of the tools we will focus on are:

- interface for batch submission of SAS jobs, which
  - frees the workstation to perform other tasks
  - stops loss of work due to workstation failure.
- interface for SAS job scheduling, which can improve system performance since jobs can execute in off peak hours.
- interface for file access, to
  - retrieve output and logs on any of the tiers with one click of the mouse
  - edit files on any tier without tier-specific knowledge (No more need to learn the cryptic "vi" commands)
- Other interesting tools enable space management, file comparison, and password changes.

Introduction

There are many advantages of a Multi-Tier Environment. We must consider them when automating frequent tasks. The advantages are:

- Powerful processors on other tiers available to the programmer.
- Centralized Data Repository.
- Workstation free to do other work.
- Power of workstation presentation tools.
- Resource sharing.

In automating frequent tasks we are trying to overcome the disadvantages of a Multi-Tier Environment. The disadvantages we are going to focus on are:

- Data transfers between tiers.
- Executing programs on the workstation ties up the workstation from doing other work.
- Multiple user id's & passwords.
- Loss of work due to workstation/connection failure.
- Resource sharing.
- Managing files on many tiers. (e.g. vi,..)

Automation needs to maximize the advantages and minimize the disadvantages. Automation should free workstation to do presentation and editing tasks (i.e. User interaction). We need to effectively manage files in a common interface regardless of operating system. Passwords need to be used to provide security to the data but not impede work. To do this we need to streamline frequent tasks.

We will look at the following frequent tasks:

- Logging onto multiple servers.
- Password changes.
- Submitting programs.
- File Management.
- Moving files to and from other tiers.
- Executing OS commands on middle tier.
- Output comparison.

Logging On and Password Changes

The scripting language and macro variables of SAS allow the user to log on to different servers with a common interface. The scripting language can also be used to change passwords on different platforms. Easier password changes means fewer forgotten passwords.

Submitting Programs

Running a SAS program interactively consumes resources on the workstation and increases the chance that a query will fail due to connection or PC failure. This implies you should not run your long running programs interactively.

We can provide an interface to run SAS programs in the batch environment to free your workstation to do other work in SAS or other products. The batch submission interface needs to make it easy to view the log and output of programs, to edit programs and to execute programs.

Programming Tips for Batch Submission

Periodic programs should run without coding changes. Passwords required in the program to access the database make this impossible as they change on a regular basis. A Password file in your home directory can be used to stop code changes because of password change.

Password File

```
%let password=user=robert password=test01;
```

Program

```
%include '~/pswd';
proc sql;
  connect to db2(database =db &password);
```

Another reason for coding changes is dates. Macro variables in the SQL can make the program work each month without code changes.

```
proc sql;
  create table rob.max_dt as select * from
connection to db2(
    select  max(a1.me_dt) from load_cntrl a1) as
a(dt);
data _null_;
  set rob.max_dt;
  if "&sysparm" ^= "" then dt = "&sysparm"d;
  call symput('dt_me',"'"||put(dt,mmddyy10.)||"'");
run;
```

```
proc sql;
  create table rob.counts as Select * from connection
to DB2(
    Select count(*) from ACCT_MTH A2
     where A2.ME_DT = &dt_me);
```

Structuring programs to be restartable by not over writing data sets reduces runtime for restart in the case of job failures.

```
proc sql;
  connect to db2(database=d3ca &password);
  create table rob.counts as Select * from
connection to DB2(
    Select PROC_CTR,
           BILLING_CYLE as cycle,
           VISA_PROD_CD as prod_cd,
           LANG_CD ,
           sum(case when DT_PREV_STMT >= &dt_1st
then 1 else 0 end) as stmt_cnt,
           count(*) as cnt
      from ACCT_MTH
      group by PROC_CTR, BILLING_CYLE,
VISA_PROD_CD, LANG_CD
) as t1( PROC_CTR, cycle, prod_cd, LANG_CD,
stmt_cnt, cnt);
proc summary data=rob.counts;
   by prod_cd proc_ctr lang_cd;
   var stmt_cnt cnt;
   output out=total sum=;
data rob.rpt;
  length cycle $5;
  set rob.counts total;
  by prod_cd;
  if cycle = '' then cycle = 'Total';
run;
```

In a multi-tier environment the database server has more computing power than the other tiers. Using the power of the database server can reduce run times.

Using structures like case can allow you to do the summing with the database server.

```
select * from connection to db2(
  select count(*) as cnt,
         sum(balance) as balance,
         case when interest > 1 then 'REVL' else
'SPND' end
    from visa_acct
   group by case when interest > 1 then 'REVL'
else 'SPND' end);
```

When working with subsets of the data using key tables can improve performance by sub setting the data in the server without doing table scans. Key tables can also be used to take advantage of common code.

```
%let grp = 'revolvers';
Proc sql;
select * from connection to db2(
   select count(*), sum(balance)
     from visa_acct v,
          key_table k
    where v.acct_no = k.acct_no
          and k.grp = &grp);
```

Sometimes large tables need to be joined in a way that is not efficient in the database engine. Use of views to avoid temporary datasets make some impossible queries possible

```
Proc sql;
create view daily as select * from connection to db2(
   select acct_no, dt_ext, balance
     from daily
          order by acct_no dt_ext
 Create view monthly as select * from connection to
db2(
   select acct_no, me_dt as dt_ext, balance
     from daily
          order by acct_no me_dt;
 data result;
   set monthly (in=m) daily (in=d);
   by acct_no dt_ext
```

Smart Browser

A directory browser frees user from having to know different operating systems. By making the browser aware of file types, viewing and editing data is much easier. Editing files from other tiers in the familiar workstation environment makes for fewer errors and faster changes.

Move Utility

A point and click interface to the upload/download procedure makes file movement fast and easy. With a common interface file movement can be more automated.

Output Comparisons

Testing is facilitated by output comparison utilities where parameters can be set to tune the comparison.

OS Command Window

User doesn't need to logon again to the target system to execute commands that can only be done with OS command.

Monitoring tools

Using SAS programs to process output from OS commands means the user can monitor disk space and other resources in a point and click environment.

Conclusion

A well planned interface to your data warehouse can significantly improve productivity.

Contact Information

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

Author Name:	Robert Ellsworth
Company:	Ellsworth Stewart Consulting Inc.
Address:	83 Bryant Rd
City state ZIP:	Markham, On. Canada L3P5Y8
Work Phone:	(416) 414-1172
Fax:	(905) 471-3298
Email:	rob@ellsworthstewart.com
Web:	www.ellsworthstewart.com