Paper SE15

Track Your SAS Jobs with Tracker

Phil Busby, Live Data Systems, Inc., Apex, NC

ABSTRACT

Tracker is a little SAS subsystem you can stick into programs you want to track. It writes an observation for the start of the job, and another when the job ends, so you can see the results of all of the jobs you care about.

INTRODUCTION

Tracker is the kind of tool you will find useful in a small shop, where you are in charge of production systems as well as new program development. It gives the SAS programmer a quick way to verify correct execution of important jobs, as well as an audit trail of all activity for all tracked jobs. Tracker was written for SAS 8.2 running Enterprise Guide 2.0 under Windows 2000.

WHAT IT DOES

This utility lets you see results of your important SAS job executions. Did a production job abend? With three mouseclicks you can look at the output record count for the job in Tracker. Tracker also can tell you who ran the job and when. Tracker also shows jobs that are started but not ended yet. You can tell by the start time usually whether or not the job is really running or it must have abended. But Tracker does not know how long a job “should” take to run before the running job’s duration suggests that it has abended.

HOW YOU GET IT

At the start of a job you want to track, drop in this line of code:

```
  %jobbegin(jobname=MYJOB);
```

This invokes the following macro jobbegin, which records the start of the job.

```
 /* Code exported from SAS Enterprise Guide
 DATE: Wednesday, March 05, 2003      TIME: 11:54:45 AM
 PROJECT: Project
 PROJECT PATH: S:\somepath\someseg.seg   ------------- */
 /* JOBBEGIN - Macro to Capture JOB START statistics.   */
 /* PFB  05mar2003  Truncate datetime to nearest second. */
 %macro jobbegin(jobname=);
   %global jobbegdt obcount;;
   /* First time only, to create data set: */
   /******************
   libname tracklib;
   libname tracklib
       "e:\somepath\sasuser\production\tracker" server=shr11;
   data tracklib.joblog;
      length job   $ 8;
      length user  $ 12;
      length jobbegdt 8;
      length jobenddt 8;
      length folder $ 60;
      length jobcount 8;
      format jobbegdt jobenddt datetime.;
   run;
   libname tracklib;
   **************************/
   data work.joblog;
     length job   $ 8;
     length user  $ 12;
     length jobbegdt 8;
     length jobenddt 8;
     length folder $ 60;
     format jobbegdt jobenddt datetime.;
     length jobcount 8;
     job  = "&jobname";
     user = "&_EGUSERID";   /* sas/eg 2.0 */
     if substr(user,1,1)='&' then
       user = "&sysjobid";
     jobbegdt= int(datetime());
     call symput('jobbegdt', trim(left(put(jobbegdt,12.))));
     jobenddt = .;
     folder   = ' ';
     jobcount = .;
     %let jobcount=;
   run;
   libname tracklib;
   libname tracklib "e:\somepath\production\tracker"
server=shr11;
   proc append base=tracklib.joblog data=work.joblog;
   run;
   libname tracklib;
 %mend jobbegin;
```

PICK YOUR OWN UNIQUE JOB IDS FOR TRACKER.

Tracker shows a job name for every job being tracked. Best to keep duplicate job names off this list. Tracker also shows the user ID. On EG 1.2 this was not available; Process ID was substituted.

When the tracked job ends, the following macro invocation logs the entry:

```
  %JOBEND(JOBNAME=myjob,
       FOLDER=mypath,
       JOBCOUNT=&jobcount);
```

The path is passed to the tracker so it will be easy to find the actual output data set from the job if it needs to be seen.

Of course, you must load the macro variable value for &JOBCOUNT, so that whatever output record count you want to track is passed to the output observation.

```
   /* ---------------------------------------
      Code exported from SAS Enterprise Guide
      DATE: Tuesday, February 18, 2003
   TIME: 04:20:25 PM
      PROJECT: someproj_graphs
      PROJECT PATH:
   S:\somepath\Production\graphs.seg
   --------------------------------------- */
   /* Tracker - JOBEND macro to capture job
   statistics at completion. */
   %macro jobend(jobname=, folder=, jobcount=);
     %global jobbegdt;
     data work.joblog;
       length job   $ 8;
       length user  $ 12;
       length jobbegdt 8;
       length jobenddt 8;
       length folder $ 60;
       length jobcount 8;
       format jobbegdt jobenddt datetime.;
       job  = "&jobname";
       user = "&_EGUSERID";
       if substr(user,1,1)='&' then
          user = "&sysjobid";
       jobbegdt= input(symget('jobbegdt'),12.);
       jobenddt = datetime();
       folder= "&folder";
       jobcount = &jobcount;
     run;
     libname tracklib;
     libname tracklib
   "e:\somepath\production\tracker"
   server=shr11;
    proc append base=tracklib.joblog
   data=work.joblog;
    run;
    libname tracklib;
%mend jobend;
```

SEE THE RESULTS

Here is the code to run when you invoke Tracker for reporting in Enterprise Guide:

```
  /* ----------------------------------------
  Code exported from SAS Enterprise Guide
  DATE: Wednesday, March 26, 2003 ME:  11:02:27
AM
  PROJECT: Tracker
  PROJECT PATH: S:\Production\Tracker\Tracker.seg
---------------------------------------- */
/* Change SORTBY to the ordering you want.

/* HISTORY   PFB     05mar2003   Initial coding

*%let sortby=descending jobbegdt;
%let sortby=job descending jobbegdt;

%macro tracker_jobs;
  data sasdata.alljobs;
    set trkdata.joblog;
    jobbegdt=int(jobbegdt);   /* truncate fraction*/
  run;
  libname trkdata;  /* Free this libref at once. */
                /* All production jobs need this library. */
  proc sort data=sasdata.alljobs;
    by job jobbegdt jobenddt;
  run;
  data sasdata.jobpair;
    set sasdata.alljobs;
    by job jobbegdt;
    if last.jobbegdt then do;
      if jobenddt > 0 then
duration=intck('second',jobbegdt,jobenddt);
      output;
      put job= user= jobbegdt= jobenddt= jobcount=
duration=;
    end;
    format duration time8.;
  run;
  proc sort data=sasdata.jobpair out=sasdata.paired;
    by &sortby;
  run;
  proc print data=sasdata.paired;
    var job jobcount duration jobbegdt jobenddt folder
user;
  run;
%mend tracker_jobs;
filename logstore
"e:\somepath\sasuser\Production\tracker\tracker.log";
  proc printto log=logstore new;
  run;
  data _null_;
    format dt datetime.;
    dt = datetime();
    put dt;
```

```
   run;
   options fmtsearch=(work) nobyline nodate nonumber;
   ods path work.templat(update) sashelp.tmplmst(read);
   ods listing close;
   ods show;
   options
SASAUTOS=("e:\somepath\sasuser\Production\Standard_
macros",sasautos)
SOURCE NOTES MPRINT MLOGIC
MAUTOSOURCE MRECALL;
   libname sasdata;
   libname sasdata
"e:\somepath\sasuser\Production\tracker\data"
server=shr11;
   libname trkdata;
   libname trkdata
"e:\somepath\sasuser\Production\tracker" server=shr11;

   proc delete data=sasdata.tracker_done;
   run;

   filename odsout;
   filename odsout
"e:\somepath\sasuser\Production\tracker\Output";

   %tracker_jobs;

   proc printto;
   run;
   data sasdata.tracker_done;
      x=1;
   run;
```

If you run with the other sort option (by job ID and execution time) you can see how each job compares to previous executions of the same program. In this way you can see instantly the way the record counts are changing over time for a particular job. Also quite useful is to watch the duration for a job change over time. A big drop in duration, with equivalent record counts, sometimes means that you have made some performance improvement; the Tracker report is often a reassuring confirmation

CONCLUSION

The presented code runs Tracker for you to watch a custom list of your jobs. The Tracker data set is simple to work with and gives you a fast report of recent job activity.

CONTACT INFORMATION

Contact the author at:
Phil Busby
Live Data Systems, Inc.
342 Calico Drive
Apex, NC 27523
Work Phone: 919/303-8902
Email: fransioli@hotmail.com
Web: www/cdbaby.com/fransioli2