Using Eventus® for WRDS through PC SAS/Connect Remote Submission

Arnold R. Cowan
Cowan Research LC and Iowa State University

This version: October 2011

Wharton Research Data Services (WRDS) offers a web query for Eventus®. The web query is user-friendly but does not support all native Eventus features. WRDS users can access all Eventus features by running programs from a personal computer through SAS for Windows or through a WRDS Unix server command line ("shell") session. This document explains the SAS for Windows method, known as remote submission.

Remote submission does not require installing Eventus (nor the CRSP™ database if you plan to use that data source) on your PC. You must have a WRDS account assigned to you by the organization where you are a current student, faculty member or other employee, and the organization must subscribe to Eventus for WRDS. You must have SAS for Windows installed on the computer you plan to use, including the SAS/Connect component, which is a typical part of universities' SAS licenses.

This document assumes a little familiarity with SAS for Windows, including how to use the Editor window and run a program (put it in the Editor and click the icon that looks like a person running). A beginning knowledge of SAS data step syntax would be useful for modifying the examples to fit your needs. You should have a basic knowledge of the file structure of your WRDS account (please see WRDS documentation).

Running a WRDS remote session from PC SAS

A *remote session* is a process in which SAS on a PC sends specific program code lines to the WRDS server, causes SAS (with Eventus enabled) on WRDS to execute the code, and downloads the results. The statements below run a remote session. The signon statement prompts you to sign on to WRDS. The results appear in the PC SAS output window and log messages appear in the Log window. You can run remote sessions repeatedly, and can run other SAS code before or after a remote session.

```
options nomautosource noimplmac; run;
%let wrds = wrds.wharton.upenn.edu 4016;
options comamid=TCP remote=WRDS;
signon username=_prompt_;
rsubmit inheritlib=(work=pcwork);
/* Insert your Eventus code and SAS code to run on WRDS here */
endrsubmit;
signoff;
```

Uploading and downloading files

To allow WRDS SAS and Eventus to use a file on the PC, or to allow PC SAS to use a WRDS file, the file must be physically transferred between the two locations, which requires specific attention in the program code being remotely submitted. This section covers the available methods for transferring files within SAS and Eventus programs.

The statements above include the option `inheritlib=(work=pcwork)` on the `rsubmit` statement. To explain the need for and effect of this statement, it is useful to review how SAS temporary data sets work. If you have used SAS in either Windows or WRDS, you may have created, changed or used SAS data sets with code like this:

```
data name;
 /* Some statements in the data step to read or      */
 /* manipulate data and produce an output data set.  */
proc something data=name; /* Any SAS proc to use the data */
```

Data set *name* above is created in the work library, which is a temporary storage location, and could equivalently be referenced in SAS code as work.*name*. PC SAS and WRDS SAS each has its own work library and cannot access data sets in the other's by using a one-part name or a two-part name starting with work. The inheritlib option allows WRDS SAS and Eventus to use a PC SAS work-library data set by referencing it as pcwork.*name*. Data sets referenced this way in the remote session are automatically uploaded on use or downloaded on creation. You can also use the inheritlib option to allow WRDS to use data sets in PC SAS user-defined libraries (a SAS term for shortcuts to PC folders); please see the SAS documentation of the rsubmit statement for details. The `inheritlib=(work=pcwork)` option appears in the examples later in this article.

You also can use Proc Download and Proc Upload to transfer SAS data sets and non-SAS files. These procedures are not needed for SAS data sets if you use the inhertilib method, but are always required for non-SAS files such as text files. To copy a non-SAS file, first use filename statements (see SAS documentation for more details) to create SAS file shortcuts on both the PC and WRDS. Filename statements pointing to PC files must run in PC SAS, while filename statements pointing to WRDS files must run in the remote session. A filename statement can point to an existing file or a file that will be created later, such as a copy to be created by downloading or uploading. Use the infile and outfile options of the Download and Upload procedures to point to the files through their shortcuts. The syntax is:

```
filename pcfile 'C:\Folder\file.etc'; /* Change C:\Folder as needed */
/* insert lines from %let wrds … through rsubmit; from p. 1 here */
filename wrdsfile 'Path/file.etc'; /* Change Path to a valid path */
proc download infile=wrdsfile outfile=pcfile;
   /* or */
proc upload infile=pcfile outfile=wrdsfile;
endrsubmit;
signoff;
```

where pcfile and wrdsfile can be changed to other allowed SAS shortcut names if desired. In Proc Download, the file to which *wrdsfile* points must already exist, and *pcfile* must point to the downloaded file to be created. The converse holds for Proc Upload. Other SAS and Eventus statements to execute on WRDS can appear between rsubmit and endrsubmit, and other SAS statements to execute on the PC can appear before and after the remote session code.

To copy a SAS data set when not using the inheritlib method, use the data= and out= options of Proc Download and Proc Upload with the *libref.datasetname* style of reference. The *libref*, or SAS library name, is a shortcut that points to a folder on the PC or a directory on the WRDS server. Use the libname statement to define a library name. The *datasetname* is the name of a SAS data set within the folder. SAS manages the physical file or files that make up a given data set. The general syntax for transferring data sets is:

```
libname pclib 'C:\Folder\Subfolder';
/* insert lines from %let wrds … through rsubmit; from p. 1 here  */
filename wrdsfile 'Path/file.etc'; /* Change Path to a valid path */
proc download data=wrdslib.mydsname out=pclib.mydsname;
 /* or */
proc upload data=pclib.mydsname out=wrdslib.mydsname;
/* Other SAS and Eventus statements to run on WRDS go here.       */
endrsubmit;
signoff;
```

where pclib and wrdslib can be changed to other allowed SAS library names as needed. When using Proc Download, data set *mydsname* must already exist in *wrdslib*, and the Windows folder pointed to by *pclib* must already exist. The procedure creates the PC copy of the data set in *pclib*. For Proc Upload, *mydsname* must already exist in *pclib* and the Unix directory or subdirectory pointed to by *wrdslib* must already exist in your WRDS account.

To download a WRDS work library data set using proc download, omit the libname statement, libref and dot (.) from the SAS data set reference. The syntax is:

```
proc download data=mydsname out=mydsname;
```

where mydsname is the name of a data set that exists, at that point in the WRDS SAS program, in the work library. The WRDS original and PC copy data sets can have different names. To upload a work data set from the PC to WRDS, either use Proc Upload in a similar manner, or simply refer to the PC SAS work library data set in the WRDS SAS code as *pcwork.datasetname* if the inheritlib option discussed above was used.

If you have a work data set on the PC (whether created on the PC or downloaded from a remote session) that you do not want to delete when you close SAS, define another library name if you don't already have one defined. In the Explorer pane of PC SAS, you can then cut and paste the data set from Work to the other library in the familiar Windows manner.

Examples

The program file and any input files for each of the following examples are attached to this PDF.

Among other things, the examples include the use of Excel files through Proc Import and the use of CSV files (possibly saved by Excel) in data steps. To import your own Excel files, we encourage you to consider a newer method not covered in this article. Please see the topic "Import Excel data more easily and accurately" in the Eventus Questions, Tips and Examples forum on support.eventstudy.com.

Basic example

This program uploads the user's text-format request file, runs an event study using CRSP, saves CARs to a work data set in PC SAS, and displays the data set.

```
/* Define shortcut to request file on PC. */
filename pcreqst 'C:\Folder\request4.txt'; /* Change C:\Folder as needed */
options ps=60 ls=129; /* Page and line size on PC side */

/* Sign on to WRDS */
options nomautosource noimplmac; run;
%let wrds = wrds.wharton.upenn.edu 4016;
options comamid=TCP remote=WRDS;
signon username=_prompt_;
/* (user responds to the user name/password popup here) */
rsubmit inheritlib=(work=pcwork);

options ps=60 ls=129; /* Page and line size on WRDS side */

/* Define shortcut to request file on WRDS Unix server.  */
/* ~/ means use my home directory; change if desired.    */
/* Make WRDS shortcut be request as expected by Eventus. */
filename request '~/tmp_request4.txt';
proc upload infile=pcreqst outfile=request; run;

/* Run the event study and write CAR to data set demo_CARs.*/
Eventus;
 Title 'Eventus for WRDS Run by PC SAS-WRDS Remote Submission';
 Request CusiPerm AutoDate;
 Windows (-10,3) (-5,+5) (-30,+30); /* CARs produced for these */
 Evtstudy OutWin=pcwork.demo_CARs; /* Data set saved on the PC */

/* Delete file from WRDS and end remote session.  */
x 'rm tmp_request4.txt';
endrsubmit; signoff;

/* Print the data set to the PC SAS output window. */
/* SAS offers several methods of viewing and sum-. */
/* marizing data sets.                             */
proc print data = demo_CARs;
 id cusip;
run;

/* End of basic example.                           */
```

Fama-French calendar-time portfolio regression example

This example shows how to run the calendar-time portfolio regression (CTPR) method using the Fama-French three-factor model. The program on the following page illustrates these techniques, most of which are not specific to the CTPR method:

- The use of Proc Import in PC SAS to import an Excel spreadsheet, SEO.xls, containing CUSIPs and event dates, into a SAS data set to be used as the request file.
- The Monthly option of the Eventus statement to run a monthly-return event study.
- The FFF specification to point to the Fama-French factors data set. On WRDS as of this writing, the factors data sets that are in the format Eventus expects are ff.factors_daily and ff.factors_monthly. These data sets are maintained by WRDS staff and questions about them should be directed to WRDS support.
- The use of a request file that is a SAS data set, instead of a text file as in the previous example. When the request file is a SAS data set, the InSAS=option of the Request statement must point to it, and no filename statement is used.
- The Shift1=+1 option of the Request statement moves the event date one month later (for a monthly event study), and the EvtStudy statement options Pre=0 Post=23 define the event period as starting with the shifted event month and ending 23 months later. In this example, the dates in the request file are seasoned equity offering dates. Thus, each security will be a member of the calendar-time portfolio for the 24 months following the offering date.
- The FamaFrench option invokes the three-factor model. The Momentum option (not shown) can be added to expand the model to four factors including momentum.
- The CTPR option selects the calendar-time portfolio regression method.
- The ValueWeightSample=Update option of the EvtStudy statement weights each security-event in the calendar-time portfolio by its market capitalization at the end of the previous month.
- The NoNames option omits the listing of the requested security events and the months of data found for each in order to shorten the displayed output.

```
/* Import the CUSIPs and dates from Excel into a SAS data set. */
PROC IMPORT
 OUT=WORK.SEO_Offer_Request(rename=(issue_date=EventDate))
 DATAFILE="C:\Folder\SEOs.xls" /* Change C:\Folder as needed */
 DBMS=XLS REPLACE;
 SHEET="SEOs";
RUN;

options ps=60 ls=132; /* Sets page lines and line length */
/* This affects PC SAS. If desired, change here and*/
/* make matching change in remote session code.    */

/* Sign on to WRDS */

options nomautosource noimplmac; run;
%let wrds = wrds.wharton.upenn.edu 4016;
options comamid=TCP remote=WRDS;
signon username=_prompt_;
/* (user responds to the user name/password popup here) */
rsubmit inheritlib=(work=pcwork);

options ps=60 ls=132;

/* Run the calendar-time event study. */
Eventus Monthly FFF=ff.Factors_monthly;
 title 'Seasoned Equity Offerings: Offer Date';
 title2 'A Sample for Demonstration Purposes Only';
 Request insas=pcwork.SEO_Offer_Request CusiPerm Shift1=+1;
 Evtstudy Pre=0 Post=23 FamaFrench CTPR
   ValueWeightSample=Update NoNames;

endrsubmit;
signoff;

/* End of Fama-French calendar-time   */
/* portfolio regression example.      */
```

Non-CRSP event study example

This example shows how to set up and run a non-CRSP event study using remote submission. You can use data from any source, so some details here may not fit your data. For example, the security key in the example program is a six-character alphanumeric string, and therefore is read using a SAS informat of $6. You can use any data type, length and name for the security key as long as the request file, security returns file and IssueKey and IssueFmt options are all consistent.

The example program also illustrates the use of the Median option (to display median CARs) and the StdCSect option (to select the standardized cross-sectional test). These options and many others can be used in CRSP and non-CRSP event studies.

```
/* Create the request file as a SAS data set. This */
/* example reads the security codes and event      */
/* dates from lines pasted into the code, but you  */
/* can use any SAS method to create your data set. */
/* All dates in this program are read using SAS    */
/* informats that create SAS date variables.       */
data request;
 input SecurityCode:$6. EventDat:yymmdd8.;
datalines;
H6576H  20050304
H0814P  20050624
H1112V  20050916
H9538X  20051213
H9778J  20060401
H7916H  20060415
H9538X  20060530
H3112V  20070212
H1184W  20070413
H9155V  20080314
H9155V  20080314
run;

/* Create the market index returns data set. This  */
/* example uses a SAS data step to read a CSV file,*/
/* but you can use any SAS method to create the    */
/* data set. The code below also checks the dates  */
/* to see whether they fall between a particular   */
/* date and today; if not, they must not have been */
/* read correctly. Check the PC SAS Log for a mess-*/
/* age generated by the data step. It is best to   */
/* also view the SAS data set manually.            */
data MarketIndex; /* Change C:\Folder below as needed */
 infile 'C:\Folder\NonCRSPStudy-Index.csv' end=_lastobs_
          dlm=',' truncover firstobs=2;
 input CalDt:mmddyy10. VWRetD;
 format CalDt yymmddn8.;
 if not ('01JAN1925'D<CalDt<today()) then do;
  put 'ERROR: Check dates! Example: ' _n_= CalDt=yymmddn8.;
  stop;
 end;
 else if _lastobs_ then put 'Dates appear plausible.';
run;

/* Create the data set containing stock returns.   */
data SecurityReturns; /* Change C:\Folder below as needed */
 infile 'C:\Folder\NonCRSPStudy-Securities.csv' end=_lastobs_
          dlm=',' truncover firstobs=2;
 input SecurityCode:$6. CalDt:mmddyy10. Ret;
 format CalDt yymmddn8.;
 if not ('01JAN1925'D<CalDt<today()) then do;
  put 'ERROR: Check dates! Example: ' _n_= CalDt=yymmddn8.;
  stop;
 end;
 else if _lastobs_ then put 'Dates appear plausible.';
run;
```

```
/* Non-CRSP event study example continued          */

options ps=65 ls=140; /* Page lines & line length. */
/* This affects PC SAS. If desired, change here and*/
/* make similar changes in the remote session code.*/

/* Run Eventus on WRDS.                            */
options nomautosource noimplmac; run;
%let wrds = wrds.wharton.upenn.edu 4016;
options comamid=TCP remote=WRDS;
signon username=_prompt_;
/* (user responds to the user name/password popup here) */
rsubmit inheritlib=(work=pcwork);

options ps=65 ls=140;

Eventus SASNONCRSP
     ReturnDS=pcwork.SecurityReturns
     IndexDS=pcwork.MarketIndex;
Title 'Eventus Event Study Using Non-CRSP Data';
Request InSAS=pcwork.request AutoDate
     IssueKey=SecurityCode IssueFmt=$6.;
Windows (-30,-2) (-1,0) (-1,+1) (-2,+2) (-5,+5) (2,30);
EvtStudy Median StdCSect OutWin=pcwork.IndividualCARs;

endrsubmit;
signoff;

/* End of Non-CRSP event study example            */
```

Conclusion

This article presents a method for users at Eventus for WRDS subscribing organizations with SAS for Windows licenses to exploit the full power of Eventus, beyond the subset of capabilities supported by the WRDS web query pages, by submitting programs remotely. Please see the Eventus User's Guide for a full listing of Eventus statements and options. Some organizations also maintain a subscription to Eventus for Windows, which provides a non-WRDS alternative to the approach described here.

Questions are welcome through http://support.eventstudy.com. This site requires a one-time registration; please register an e-mail address at your Eventus-subscribing organization for help with Eventus usage.

The author is the creator of Eventus® software and president and CEO of Cowan Research LC, which is solely responsible for this document. Arnie also is Professor of Finance and Wells Fargo Professor in Finance at Iowa State University. Thanks to Eugene Kang Soon Lee and Owen Tidwell for comments on earlier versions of this article.