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

Arnold R. Cowan

Cowan Research LC and Iowa State University

This version: April 2011

Wharton Research Data Services (WRDS) offers a web query for Eventus® software. The web query is user-friendly but does not support all native Eventus features. WRDS users can access the full Eventus feature set 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 (or 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 institution at which you are a current student, faculty member or staff member, and the institution must subscribe to Eventus for access through 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 site licenses.

This document assumes just a little familiarity with the SAS for Windows environment, including how to use the Editor window and how to run code (put it in the Editor and click the icon that looks like a person running). To adapt an Eventus (or SAS) program to the remote submission method, you need to know how to run a WRDS remote session and how to upload and download files.

Running a WRDS remote session from PC SAS

A *remote session* is a process in which SAS on a PC 1) sends a specific set of SAS or Eventus code to the WRDS server, 2) causes SAS (with Eventus enabled) on WRDS to execute the code and 3) downloads the results. The statements shown on the following page run a remote session. The signon statement pops up a request for your WRDS user name and password, which you must provide. The results appear in the PC SAS Output window and log messages appear in the Log window. You can run remote sessions as many times as needed, and can run other SAS code before or after a remote session, in the same PC SAS Editor window or a different one.

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

Uploading and downloading files

In general, SAS on the WRDS server cannot directly read a file on your PC or vice-versa. PC files and WRDS server files are logically distinct and must be physically transferred between the two locations when needed.

The statements above include the option *inheritlib=(work=pcwork)* on the rsubmit statement. This refers to the *work library* in PC SAS. If you have used SAS in either Windows or WRDS, you may be used to creating or changing, and using, SAS data sets with code like this:

```
data name;
 /* some statements in the data step to read or manipulate data */
proc something data=name;
```

The data set *name* is created in the work library, which is a temporary storage location, and could equivalently be referenced in SAS code by explicitly using the two-part name *work.name*. PC SAS and WRDS SAS each has its own separate work library and does not "see" data sets in the other's work library. The inheritlib option allows WRDS SAS to use a PC SAS work library data set that your code explicitly references as *pcwork.name*. This is done in the final example of this article. You can also use the inheritlib option to allow WRDS SAS to use data sets in user-defined libraries (which are really SAS shortcuts to ordinary folders on the PC). Please see the SAS documentation of the rsubmit statement for more details.

More generally, you can use Proc Download and Proc Upload to copy SAS data sets and non-SAS files between WRDS and the PC. To copy a text file or other 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 WRDS SAS. Use the infile and outfile options of the procedures to point to the files through their shortcuts. The syntax is:

```
filename pcfile 'C:\Folder\file.etc';
/* before the filename wrdsfile statement, insert the */
/* code to start a remote session, ending with rsubmit; */
filename wrdsfile '~\directory\file.etc';
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, use the *libref.datasetname* style of reference and the data= and out= options of the Proc Download and Proc Upload statements. 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 SAS data sets is

```
libname pclib 'C:\Folder\Subfolder';
/* before the libname wrdslib statement, insert the */
/* code to start a remote session, ending with rsubmit; */
libname wrdslib '~\directory\subdirectory';
proc download data=wrdslib.mydsname out=pclib.mydsname;
/* or */
proc upload data=pclib.mydsname out=wrdslib.mydsname;
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. 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 download a WRDS work library data set using proc download, omit the libname statement, libref and . 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 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

Basic example

This example illustrates the basics of working with SAS and Eventus for WRDS and SAS for Windows together. The program uploads the user's request file in text format to WRDS, runs an Eventus event study, extracts window CARs to a SAS data set, downloads the SAS data set, and "prints" it to the PC SAS Output window.

```
/* Define shortcut to request file on PC. */
filename pcreqst 'C:\Documents\file.txt';
/* 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;
options ps=50;
/* Define shortcut to request file on WRDS Unix server.  */
/* ~/ means use my home directory.                        */
/* Make WRDS shortcut be request as expected by Eventus. */
filename request '~/demorequest.txt';
proc upload infile=pcreqst outfile=request; run;
/* Run the event study and write CAR for selected windows   */
/* to temp SAS data set democar. Remove CusiPerm if the     */
/* request file contains PERMNOs.                           */
Eventus;
 Request CusiPerm AutoDate;
 Windows (-10,3) (-5,+5) (-30,+30);
 Evtstudy OutWin=work.democar;
/* Copy democar from WRDS SAS work library to PC work library */
 Proc Download data=democar out=democar; run;
/* End remote session. */
endrsubmit;
signoff;
/* Print the data set to the PC SAS output window. */
proc print data=democar;
 id cusip;
run;
```

Fama-French calendar-time portfolio regression example

This example shows how to run the calendar-time portfolio regression method using the Fama-French three-factor model. The example illustrates the following techniques.

- The use of Proc Import in PC SAS to import an Excel spreadsheet (containing CUSIPs and event dates) into a SAS data set. You can download the spreadsheet for the example from http://www.eventstudy.com/DocLatest/SEOs.xls. If you are running 64-bit PC SAS, change the word EXCEL to XLS in Proc Import.[1]
- The Monthly option of the Eventus statement to run a monthly-return event study.
- The FFF option 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; 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. In the absence of the IRATS or TwoStep options, the FamaFrench option also invokes the calendar-time portfolio method. The Momentum option (not shown) can be added to expand the model to four factors including momentum.
- The ValueWeightSample=Update option of the EvtStudy statement weights each security-event in the calendar-time portfolio by market capitalization.
- The NoNames option suppresses the listing of the requested security events and the number of months of data found for each, for convenience when viewing the output.

[1] Proc Import is one of a few options for importing Excel files into SAS. Please see SAS documentation for other methods.

```
PROC IMPORT
 OUT=WORK.SEO_Offer_Request(rename=(issue_date=EventDate))
 DATAFILE=”C:\Documents\Eventus\Sample Programs and Data\SEOs.xls”
 DBMS=EXCEL REPLACE; /* Change EXCEL to XLS if using 64-bit SAS */
 SHEET="SEOs";
RUN;
/* 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=50;
/* 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;
```

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 Owen Tidwell for comments on an earlier version of this article.