Abstract
By using Base SAS and SAS macro language, we have created a contract payment system for the reimbursement of clinical sites based on their Case Report Form completion. With only a few SAS programs, the system automatically assigns payment values, generates summary report, and outputs letters to be mailed.

This paper discusses the issues surrounding the creation of the contract payment system. SAS code associated with key functions is presented.

Introduction
In multi-center clinical trials, reimbursement of clinical investigators for their participation in the trial can be a thorny issue. On one hand, it is essential for the sponsor to cover the clinical site’s costs of implementing the trial, as well as reimburse the participating subjects. On the other hand, it is critical for the sponsor to collect reliable data from the sites. Therefore, it is desirable to create a payment structure that reimburses the sites in a timely manner, while at the same time encouraging them to collect clinical data in real time. Set payments can be made up front, or periodically throughout the trial, but there is no guarantee that the sites will collect quality data. Conversely, holding payment until the very end of the trial is not a solution, as many clinical sites depend on the payments to cover the costs of carrying out the trial. It also may discourage the sites to provide timely data.

One of the solutions is to implement a ‘pay-for-performance’ model, where sites are paid based on the number of subjects they see. They can be given an initial payment to cover set-up costs, but once the trial is underway, they are paid only based on their progress.

How do we quantify a site’s progress? It is unreasonable to expect that the sponsor’s Clinical Research Associates (CRAs) can closely monitor a site’s progress, especially when often each CRA is responsible for many sites and hundreds of subjects throughout the country. Furthermore, how do we assign a dollar value to what would ultimately be a subjective measure?

One simple answer: use the data. We can tell a site’s progress by looking at the amount of data from each subject at that site. Specifically, the number of CRFs that have been data entered in the computer provides an accurate glimpse of progress of the trial at the site, which translates how much the site should receive in payment.

By flagging essential CRFs as ‘required’, we can set up, for each visit, a final per-subject visit status of ‘Completed’ or ‘Not completed’. For a visit to be considered ‘Completed’ all required CRFs must completed with no outstanding queries. A visit is not considered ‘completed’ until all queries have been resolved for all required CRFs. Once a subject visit is ‘completed’, the site can be paid at the end of the quarter at an agreed-upon rate for each subject visit. The total payment to the site would be the product of all the completed visits times the per-subject visit rate. For example, if a site is paid $50 for each completed subject visit, and they complete 50 subject-visits, then they would be paid $2500.

Methodology
This is most easily visualized by looking at the data on a subject-visit level. In many clinical trials, recruited subjects are required to complete several visits prior to their successful completion of the trial.

In our example, the clinical trial is a vaccine trial. Subjects receive vaccinations over the course of 36 months for a total of eight visits. Additionally, they are required to return two weeks after each vaccine visit to measure reactogenicity at the injection site. This makes
for a total of 16 visits. Each visit has its own set of CRFs, and some require more than others. For example, the two-week reactogenicity evaluation visits have only three required forms, while the vaccine administration visits have many more. However, not all CRFs are required.

Regardless of the number of CRFs that are completed at each visit, by flagging the critical CRFs as ‘required’, you can assign a final visit status of ‘Completed’ or ‘Not completed’.

A flow chart illustrates how the system processes the information. Detailed descriptions are followed.

Prior to study initiation
In order to fully utilize the benefits of a Contract Payment System, it is essential to agree upon a pay rate for each visit completed. This can be standardized across all clinical sites, or can vary depending on site enrollment.

Additionally, it is necessary to obtain site information such as payment contact and address, principal investigator information and address, agreed-upon pay rate per visit, and any other information that could be stored as macro variable information (to be used in the form letters that accompany the payment to each site).

Finally, write a standardized form letter that will go to each of the sites accompanying the payment. For the parts of the letter that will differ among the sites (number of subjects and visits, payment rates, number of completed CRFs, etc.), insert macro variables (which will eventually be created and resolved in the programming below). A section of it might look like this:

The enclosed information indicates a total of &nsub subjects enrolled and &ncrf completed subject visits (including screening enrollments) for your center. At a reimbursement rate of &rate per subject visit, and including the xx% withhold, your site is due reimbursement of &totalpay. Since total reimbursement due is &moreless than sponsor's accumulative payments of &prepay, sponsor's check in the amount of &thispay will be sent to the payment address listed in our Clinical Trial Agreement.

The macro variables (preceded by the ampersand ‘&’) will be different for each site, depending on its progress.

At each quarter
At the end of each quarter, the Contract Payment System will generate reimbursement reports and letters. There are several steps involved here.

First, extract cumulative (most up-to-date) CRF data into SAS datasets on the last day of the quarter. This can be set up to run automatically (for example, at the end of each quarter: on March 31, June 30, September 30, and December 31).

Once the most recent data has been extracted, a SAS program takes each dataset (corresponding to each CRF) and merges it by subject and visit. Data sets with multiple records per visit need to be transposed by subject ID and visit ID so that there is a one-to-one relationship among all data
sets. Remember, the data itself is of no importance compared to the existence of data for each subject and visit.

Based on the existence of each CRF at each corresponding visit, assign a status of ‘1’ indicating that the CRF is present. Ultimately, a dataset containing some identifying variables (subject ID, site ID, visit ID) and one ‘completion status’ variable for each CRF is created. The next step is to assign the completion status of an entire visit by using the SUM function. For example, for the vaccine visit, where there are six required CRFs, the SAS code would look like this:

```sas
if sum(of form1 - form6)=6 then statusv1=1;
```

For each visit, a completion status is assigned based on the number of required CRFs, so it would look something like this:

```sas
if sum(of form1 - form6)=6 then statusv1=1;
if sum(form7,form8,form9)=3 then statusv2=1;
... /* more SAS code*/
```

Now we can see which visits are considered complete for each subject, and from there we can, using the SUM function again, get the total number of completed visits for each subject, like this:

```sas
allstatus=sum(of statusv1-statusv8);
```

The next step is to assign the total number of completed visits by site, rather than by subject ID. This can be done easily with a PROC MEANS, which looks something like this:

```sas
proc means noprint;
  var statusv1-statusv8 allstatus;
  output out=sum sum=sitev1-sitev8 siteall;
run;
```

With this information, we are now able to multiply the number of completed visits with the agreed-upon pay rate for each site. For example, a site that will be paid $50 per visit and has completed 50 total visits will be paid $2500.

Finally, we have obtained cumulative payment amount by site, from the CRF and the contact information we gathered at the start of the study.

### Quarterly Update

Each quarter, the cumulative payment amount is calculated and stored in permanent SAS datasets for future use. The total amount of money that should be paid to the clinical site is for the entire length of the study up to this quarter. By subtracting sum of money paid for all previous quarters from those in this quarter, we can derive the amount of reimbursement for this quarter. In the above example, the cumulative amount due for the quarter is $2500. However, the cumulative amount due to the site was $2000 last quarter. We need to subtract $2000 from $2500 in order to get the money that the site earned this quarter.

### Reporting the dollar information

We now have all the information to generate a quarterly report for each site. We include both a site-level and study level summary of the payments to date, which provides the site with the exact number of completed visits for each subject-visit, the total payment to date, and the payment for this quarter. This is accomplished by PROC REPORT.

```sas
proc report data=qsum nowd spacing=2 split='' missing headline headskip ls=132;
  Column("--" Category thisqtr lastqtr change);
  Define category / display ' ' width=12;
  ... /*more define statements*/
  By siteid;
run;
```

We can construct the form letter that will go to the sites. The information is as follows:

All contact information (from above)
Pay rate (for each site, if different)
Total number of subjects
Total number of completed visits
Total number of completed visits for the previous quarter
Total monies due to the site (all quarters)
Total monies due to the site this quarter

With the above information we can create a form letter that will be personalized for each clinical site. We can do this by using a macro which will loop through each site and, for each site, assign a macro variable which corresponds to each piece of the above information (using data _null_).

An example of how the macro looping and macro variable assignment will look follows:

```sas
%macro rpt;
%do i=1 %to &nosites;
  data _null_; /*Set in the payment data created above*/
  set paydata;
  if _n_=&i;
  /*Assign macro variables from the contact information*/
  call symput('pilast',compress(pilast));
  call symput('piaddr1',piaddr1);
  . . . /* More SAS code */
  run;

/*Read in the form letter that was created above*/
data letter;
set letter;
select;
/*Insert site contact information (recipient name, address, etc.)*/
  when (line=5) text="&sdate";
  when (line=10) text="&pifirst &pilast, &degree";
  . . . /* More SAS code */
/* Output the letter to a text file*/
filename outf "letter to site&i.txt";
proc printto print=outf new;
run;
proc print data=letter noobs;
var text;
run;
proc printto;
run;
%mend rpt;
```

The above code will, for each site, replace the macro variables from the form letter with the real data from the payment data set you constructed. The result will be a personalized form letter for each site with all the site’s correct contact information and most recent payment information.

**Final thoughts**
The above system can be tailored to your specific needs. You can make the form letter as detailed as you’d like. This requires more work up front, but you’ll find that once it’s created, it should run smoothly each quarter.

Additionally, we include a detailed subject-level listing which, for each subject, provides the site with the status of each visit (whether or not it is completed) and the total number of completed visits. The sites find this useful in tracking down visits (and specifically, CRFs) which, for whatever reason, were not sent to us.

**Conclusion**
With knowledge of Base SAS and the SAS Macro Language it is possible to create, implement, and maintain a Contract Payment System that can be used throughout the duration of a multi-site clinical trial. Furthermore, it is easily transferable among trials, with relatively few modifications.

**Acknowledgments**
Special thanks go to Joe Robinson, who initially proposed to automate the contract payment system to the sites. Thanks also to Valerie Pierce for her assistance in the ongoing maintenance of this system.

**Information About Authors**
Mark McLaughlin is an independent consultant. He is specialized in development of SAS applications in biotechnology and pharmaceutical industries. He has been using SAS since 1994. His contact information is:

666 Greenwich St. #608
New York, NY 10014
Phone: (212) 463-7228
Email: henry.dog@verizon.net
Elizabeth Li is a SAS certified professional™. She is a senior consultant at DIEM Computing Services, Inc., a contract research group specializing in statistical consulting and SAS program development for biotechnology and pharmaceutical companies. She has been using SAS since 1987 and has presented several papers in user groups at the regional level. Her contact information is:

38851 Litchfield Circle  
Fremont, CA 94536  
Phone: (510) 744-1304  
Fax: (510) 744-1412  
Email: elizabethli@diemcomputing.com  
www.diemcomputing.com

Trade Mark
SAS, SAS Certified Professional, and SAS Quality Partner are registered trademarks of SAS Institute Inc. in the USA and other countries.® indicates USA registration.