ABSTRACT
In this paper we present an application used to track serology samples for clinical trials. The system used SAS to interact with a Clinical data management system (Clintrial) and an Oracle database through the use of Base SAS, SAS Macro, SAS/AF, Open Data Base Connectivity (ODBC) and SAS Output Delivery System (ODS). Serology samples are collected from subjects at various time points during the conduct of clinical trials designed to evaluate investigational vaccines and immunoglobulins. Historically the serology inventory system was implemented using Microsoft Access. This required double entry of data in the Clintrial and MS Access databases, and required constant checking to assure they were identical. Furthermore, the Access database lacked an audit trail. Users who needed to access the systems were completely dependent on the programmers to write the necessary queries. By using SAS to interact with the clinical database management system, we were able to implement a system that resolved user access issues, data integrity, redundancy, and improved the efficiency of our process.

INTRODUCTION
Aliquots of sera for determination of antibody titers to investigational vaccines and immunoglobulins are collected from subjects at various time points during the conduct of a clinical trial. Keeping track of the vials and linking them to the antibody results are the primary objectives of the system.

Each vial is uniquely identified with a barcode representing the time the specimen is obtained, and a vial number. The barcode information and the number of vials collected are recorded on the case report form (CRF) and are entered in the clinical database. Insuring the accuracy of the inventory at the warehouse with the data in the clinical database is the key function of the system. In addition, the system generates a list of barcodes to be sent by the warehouse to the laboratory for testing and insures that only legitimate vial identification numbers are sent for testing. The system generates the necessary reports based on the user’s request to provide the status of the inventory. These detail vials that have been sent for testing, those that remain in inventory, and dates associated with current vial status.

THE OLD WAY
Previously, the serology inventory system was implemented using a Microsoft Access database. “Pull lists” were generated using queries that updated the Access database tables to identify the status of each vial; identification numbers, and dates of vials sent to the laboratory. The serology data had to be entered twice: once in the clinical database using the existing version of clintrial, and again in the MS Access database. Changes to the clinical database had to be duplicated in the Access database, which was not always done in a timely manner. This caused errors resulting from the data entry process, and required constant checking of the Access and the clinical databases to insure that the two contained the same serology inventory information. Furthermore, the MS Access database lacked an audit trail to keep track of changes to the database tables. Additionally, the study directors who needed the information from the MS Access data were completely dependent on the programmers to write the necessary queries to provide them with that information. This information was not always delivered in a timely manner, and resulted in substantial frustration.

The concerns mentioned above which reduced the trust in the quality of the data along with the inefficiency of generating the pull list and the dependence of the user on the programmers to generate pull lists influenced our decision to develop a better system.

THE SOLUTION
To resolve these issues, we designed the new system to perform all the functions necessary to insure the accuracy of the serology inventory information, and allow study directors to access data without requiring programmer intervention.

The main functions of the Serology Inventory System is to create the list of vial identification numbers (“pull list”) which accompanies the specimens when they are sent from the warehouse to the laboratory. The system also maintains the assay (testing) status of the vials so that the study directors can generate ad hoc reports of the inventory, and compares and monitors the accuracy of the serology information in the clinical database with the serology information in the warehouse database. The main operating characteristic of the system had to be its user friendliness and its ease of operation.

The serology clinical data, which is recorded on the CRF, is stored in Oracle tables through data entry using clintrial version 4.2. Using Microsoft ODBC for Oracle
we extract the clinical data for use by the SAS/AF front-end application. The pull list is generated based on the user input. From the main menu (Figure 1.1) the user will enter information to identify the study number (protocol number) from a pull down menu that is dynamically populated each time the system is started (Figure 1.2). Once the user selects a study, the treatment list is populated with the number of injections each subject received during the study (Figure 1.3). After the user selects the injection(s), the "visit day" list is dynamically populated with all possible associated visits on which blood was collected (Figure 1.4). The user selects the visit(s) of interest and a potential "pull list" is generated. After selecting the visit(s) (Figure 1.5) the cryotubes (Vials) list is populated with all possible associated cryotube numbers (Figure 1.6). The "pull list", is further defined by the user by selection of specific cryotube(s) from the available pool of cryotubes. Before generating the actual pull list, the user is provided with a list of subjects meeting the pull list defined by user input criteria (Figure 1.7). Thus the user is given the option of selecting a single, multiple or all subject to be included in the pull list.

Once the pull list is generated, the system will provide the user with a report containing a list of vial identification numbers, the injection number(s), the visit day(s), and a fake subject identification number, which meet the criteria selected by the user (Table 1.1). A fake subject number is provided so that the laboratory may test the samples for one subject at one time without breaking the study blind. An electronic copy of the list is generated, and sent to the warehouse and to the laboratory in MS Excel format. A text file is generated to populate the inventory table in the Oracle database (Table 1.2). The inventory table is used by the system to maintain the accuracy of the serology inventory, and to provide the user with information the user may request using interactive querying.

The user can verify at any time, which vial identification numbers should have been sent to the laboratory from the warehouse. The user is provided the option to query the database for all information about all vials sent to be tested. The user can identify all vial numbers that are available to be sent to the laboratory following a process similar to that used to generate the pull list. The user has the option of performing a set of queries independent from the programmers. To query the database, the user selects a query option from the main menu allowing selection of a specific protocol from a pull down menu (Figure 2.1). Once the user selects the protocol the user is provided access to the SAS query window (Figure 2.2). The user then can query for any combination of columns and values.

**FINAL THOUGHTS**
The current process has been a great improvement for our organization. The inventory tracking system is validated and has a valid audit trial, since all the data used and generated are stored in Clintrial. It is no longer necessary to enter data in two different databases, and there are no longer any data quality issues. The only databases that are compared are that of the warehouse and our clinical database. Because of the user-friendly interface, users no longer need programmers to generate pull lists -- they launch the application on their personal computers and then have at it. The whole process has been improved and time has been saved. This has been a positive change for us, and it shows how SAS can be used to help improve an organization's business processes.
### Pull List Excel Format File

<table>
<thead>
<tr>
<th>PatientID</th>
<th>Injection</th>
<th>Visit</th>
<th>BarCode</th>
</tr>
</thead>
<tbody>
<tr>
<td>005193</td>
<td>1</td>
<td>0</td>
<td>1359058401</td>
</tr>
<tr>
<td>005193</td>
<td>1</td>
<td>7</td>
<td>1359059001</td>
</tr>
<tr>
<td>005193</td>
<td>2</td>
<td>0</td>
<td>1359057001</td>
</tr>
<tr>
<td>005193</td>
<td>2</td>
<td>7</td>
<td>1359060401</td>
</tr>
<tr>
<td>005193</td>
<td>2</td>
<td>14</td>
<td>1359057201</td>
</tr>
<tr>
<td>005193</td>
<td>2</td>
<td>21</td>
<td>1359056801</td>
</tr>
<tr>
<td>005193</td>
<td>2</td>
<td>28</td>
<td>1359057501</td>
</tr>
<tr>
<td>005193</td>
<td>2</td>
<td>35</td>
<td>1359058201</td>
</tr>
<tr>
<td>005193</td>
<td>2</td>
<td>42</td>
<td>1359062301</td>
</tr>
<tr>
<td>005193</td>
<td>2</td>
<td>70</td>
<td>1359064101</td>
</tr>
<tr>
<td>005327</td>
<td>1</td>
<td>0</td>
<td>1359058501</td>
</tr>
<tr>
<td>005327</td>
<td>1</td>
<td>7</td>
<td>1359058801</td>
</tr>
<tr>
<td>005327</td>
<td>2</td>
<td>0</td>
<td>1359058101</td>
</tr>
<tr>
<td>005327</td>
<td>2</td>
<td>7</td>
<td>1359059201</td>
</tr>
<tr>
<td>005327</td>
<td>2</td>
<td>14</td>
<td>1359060101</td>
</tr>
</tbody>
</table>

### Pull List Text Format File

<table>
<thead>
<tr>
<th>Patient ID</th>
<th>Day</th>
<th>Code</th>
<th>Visit</th>
<th>Time</th>
<th>BarCode</th>
<th>ExpDate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>1359037801</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359037901</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359038001</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359038101</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359038201</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359038301</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359038401</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359038501</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359038601</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359038701</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359038801</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359038901</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359039001</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359039101</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359039201</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359039301</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359039401</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359039501</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359039601</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359039701</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359039801</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1359039901</td>
<td>1001</td>
<td>2</td>
<td>1</td>
<td>20011127</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1.1

Table 1.2
Sample of Query.scl source code

/**********************************************************************
/* QUERY.SCL */
/* PREPARE DATA FOR QUERY DISPLAY */
/* DEVELOPMENT/MAINTENANCE HISTORY */
/* DATE BY NOTE */
/* 17SEP2001 M.CHARNY Initial Creation */
/**********************************************************************
/* Sample of Query.scl source code */
rc = rc ;
INIT:
PanelNameX = "SEROLOGY" ;
TabDispl.visible = 'No' ;
Pull_listX = "INVENTORY_data" ;
return ; /* INIT */

MAIN:
return ; /* MAIN */

TERM:
_statuses_ = ('H') ;
return ; /* TERM */

EXIT:
link TERM ;
submit continue ;
    proc datasets library = work nolist ;
    run ;
endsubmit ;
return ; /* EXIT */

PROTNUM:
/* Select Protocol Number */
ProtNumX = ProtNum.selectedItem ;
if ProtNumX eq ' ' then return ;
DataTable1 = PanelNameX!!'_data' ;
DataTable2 = PanelNameX!!'_update' ;
submit continue ;
    options nodate pageno = 1
        sasautos = ('G:\Clinical\SysInvent\SysMacro') ;
    libname SR ODBC USER=sasread PASSWORD=sasread
        DATASRC="CT42" schema = &ProtNumX ;
    data SEROLOGY_ALL ;
        set sr.&DataTable1
            sr.&DataTable2 ;
    run ;
endsubmit ;
if exist('SEROLOGY_ALL') then do ;
    submit continue ;
    %macro CRTUB ;
        %global MaxTub ;
        /* Determine the Max Number of the CryoTubes */
        proc sql noprint ;
            select max(substr(NAME ,5)) into: MaxTub
                from sashelp.vcolumn
                    where libname = upcase("WORK") &
                        memname = upcase("SEROLOGY_ALL") &
                        NAME like "TUBE%" ;
        quit ;
        /* Extract Serology Data based on User Input */
        data SEROLA ;
            set SEROLOGY_ALL(keep = SID INJECTION BARCODE
            /************* */
        /* Macro CRTUB: This program extracts */
        /* Serology Data from Oracle */
        /* DataBase based on the user input. */
        /************* */
    %endmacro ;
/* Create New Variable CRYOTUBE and for each Visit Day */
data SEROLOG ;
length CRYOTUBE 3 ;
set SEROLA ;
by SID INJECTION VISIT ;
array TB(*) $ TUBE1 - TUBE&MaxTub ;
if first.VISIT then do ;
do I = 1 to dim(TB) ;
if TB(I) ^= "" then CRYOTUBE = (I) ;
else CRYOTUBE = . ;
output ;
end ;
drop TUBE1 - TUBE&MaxTub I NOTDONE ;
run ;

data SEROLOG ;
length CRYOTUBE $ 2 ;
set SEROLOG ;
if CRYOTUBE ^= . then
   CRYOTUBE = put(CRYOTUBE , z2.) ;
else CRYOTUBE = '' ;
drop CRYOTUB  ;
run ;

/* Create new var BarCode by concatenation Master BarCode and Cryotube Value */
data SEROLOGY ;
length BarCode $ 10 ;
set SEROLOG (rename = (BARCODE = BARC)) ;
BarCode = compress(BARC) || compress(CRYOTUBE) ;
format DateProcess date9.;
drop BARC ;
run ;
%
endsubmit ;
end ; /* PullList data exist */
else
   do ; /* PullList data does not exist */
      do ; /* Extract Serology Data Set */
      proc sort data = SEROLOGY(drop = DateProcess) out = PLIST ;
         by BarCode CryoTube ;
      run ;
      /* Add Information from the Data Base: BarCodes , */
      /* DataProcess which where already set to the Lab */
      data ALL_PLIST ;
         length VISIT $ 7 ;
         set PLIST (rename = (VISIT = VS)) ;
         VISIT = VS ;
drop VS ;
run ;
endsubmit ;
end ; /* Extract Serology Data Set */
end ; /* PullList data does no exist */
else
   do ; /* PullList data exist */
      submit continue ;
      proc sort data = SEROLOGY(drop = DateProcess) out = PLIST ;
         by BarCode CryoTube ;
      run ;
      /* Sort PullList Data Set */
      proc sql ;
      create table OldList as
      select BarCode , CryoTube ,DateProcess ,BATCH
      from sr.&Pull_ListX
      order by BarCode , CryoTube ;
      quit ;
      /* Exclude BarCodes which where already sent to the Lab */
      data ALL_PLST (keep = SID INJECTION VISIT BARCODE DATE_PROCESS)
      retain SID INJECTION VISIT BARCODE DATE_PROCESS CRYOTUBE BATCH ;
      merge PLIST (rename = (VISIT = VS) in = CL)
      OldList as
      select BarCode , CryoTube ,DateProcess ,BATCH
      from sr.&Pull_ListX
      order by BarCode , CryoTube ;
      length VISIT $ 7 DATE_PROCESS $ 9 ;
      VISIT = VS ;
      if DATEPROCESS ^= . then
         do ;
         DateProc = datepart(DATEPROCESS) ;
         yy = put(year(dateproc) z4.) ;
         dd = put(day(dateproc) ,z2.) ;
         mm = put(month(dateproc) , z2.) ;
         Date_Process = compress(mm||dd||yy) ;
         end ;
      if CL ;
drop VS DATEPROCESS DateProc yy dd mm ;
run ;
proc sql ;
create table All_Plist as
select SID , INJECTION , VISIT , BARCODE ,
DATE_PROCESS , CRYOTUBE , BATCH
from All_PLST
order by SID ,VISIT ,INJECTION ,CRYOTUBE ;
quit ;
endsubmit ;
end ; /* PullList data exist */
else
   do ; /* PullList data does not exist */
ts = open('ALL_PLIST' , 'i') ;
else
   do ; /* PullList data exist */
      submit continue ;
      proc sort data = SEROLOGY(drop = DateProcess) out = PLIST ;
         by BarCode CryoTube ;
      run ;
      /* Sort PullList Data Set */
      proc sql ;
      create table OldList as
      select BarCode , CryoTube ,DateProcess ,BATCH
      from sr.&Pull_ListX
      order by BarCode , CryoTube ;
      quit ;
      /* Exclude BarCodes which where already sent to the Lab */
      data ALL_PLST (keep = SID INJECTION VISIT BARCODE DATE_PROCESS CRYOTUBE BATCH)
      retain SID INJECTION VISIT BARCODE DATE_PROCESS CRYOTUBE BATCH ;
      merge PLIST (rename = (VISIT = VS) in = CL)
      OldList as
      select BarCode , CryoTube ,DateProcess ,BATCH
      from sr.&Pull_ListX
      order by BarCode , CryoTube ;
      length VISIT $ 7 DATE_PROCESS $ 9 ;
      VISIT = VS ;
      if DATEPROCESS ^= . then
         do ;
         DateProc = datepart(DATEPROCESS) ;
         yy = put(year(dateproc) z4.) ;
         dd = put(day(dateproc) ,z2.) ;
         mm = put(month(dateproc) , z2.) ;
         Date_Process = compress(mm||dd||yy) ;
         end ;
      if CL ;
drop VS DATEPROCESS DateProc yy dd mm ;
run ;
proc sql ;
create table All_Plist as
select SID , INJECTION , VISIT , BARCODE ,
DATE_PROCESS , CRYOTUBE , BATCH
from All_PLST
order by SID ,VISIT ,INJECTION ,CRYOTUBE ;
quit ;
endsubmit ;
end ; /* PullList data exist */
if ts > 1 then ts = close(ts) ;
   TabDispl.visible = 'Yes' ;
   QueryDisp.table = "ALL_PLIST" ;

return ; /* PROTNUM */

***** END OF THE TEXT ***** ;

CONTACT INFORMATION
Contact the authors at:
Mariya Charny
Nabi
12276 Wilkins Ave.
Rockville, MD 20852
(301) 255-6923
Mcharny@Nabi.com

Ala Sweidan
Nabi
12276 Wilkins Ave.
Rockville, MD 20852
(301) 255-6915
Asweidan@Nabi.com