

Use of SAS® for Clinical Trial Management and Risk-Based Monitoring of Multicenter Clinical Trial Data from Electronic Data Capture Tools

Robert Hall¹, MS, Rebecca V. Fink² MPH, David Gagnon¹, MD, MPH, PhD

¹Boston VA Cooperative Studies Program - MAVERIC / VA Boston Healthcare System, Boston, MA; ²Abt Associates, Cambridge, MA

ABSTRACT

Risk-based monitoring of clinical trials is an approach that combines on-site monitoring along with centralized remote monitoring by coordinating centers, to improve the quality of research study data and enhance human subject protection.¹ This methodology focuses on risk assessments of the captured information and designing protocol specific approaches with periodic data evaluations to ultimately improve the effectiveness of monitoring activities and study efficiency. One arm of this approach is using operational trial data to identify sites with higher frequencies of data irregularities and target their activities with monitoring and training to improve outcomes data.² Electronic data capture (eDC) systems, which have revolutionized collection of clinical research data, can impact these monitoring efforts. While commercial eDC tools offer a number of data management and reporting applications within their interfaces, one can further capitalize on backend system tables to enhance reporting and inform project stakeholders of study activities. SAS reporting features can be used to analyze system tables and provide custom reports that benefit risk-based monitoring as well as other trial management processes. For instance, tables on case report form (CRF) status levels can be used to generate reports on missing or expected form completion for project managers, site personnel, and monitors. In addition tables on automated data clarifications (DCF), monitoring or data management reviews, audit trails, or data freezing/locking can be utilized to inform trial personnel of ongoing study activities. Reports based on eDC system tables can be enhanced with SAS Output Delivery System (ODS) in generating reports in multiple formats such as HTML, RTF, or Excel files. Efficiency of programs can be improved with PROC SQL and ODBC connections to eDC system tables that are SQL based. This paper and presentation will explore methods that can be used to improve eDC clinical reporting and assist with risk-based monitoring to lead to better quality data, participant safety, and cost-effectiveness and improvements in the administration of research trials.

INTRODUCTION

Multicenter trials require periodic monitoring of patient recruitment, patient visits and captured research data to insure the information is complete and is of the highest quality for analysis. Given the complexity of multicenter trials it can become a challenge to manage this data in a timely and cost effective manner. Electronic data capture (eDC) tools have replaced manual, paper methods and provide many software applications that assist in the management of enrolled patients and study data. Many eDC systems provide backend tables and/or views that can facilitate trial management beyond just the captured clinical data. In addition, such reports can assist in risk-based monitoring efforts to follow site activity and performance during the course of a clinical trial. Risk-based monitoring is a systematic risk-based approach to determine the necessary level of on-site monitoring with various study sites. One aspect of this monitoring process is periodic evaluation of site activities when it comes to data capture, recruitment, protocol deviations, and other data related tasks. For instance, reports can be generated to facilitate the tracking of CRF form completion at the site level. Aggregate summaries of form completion and timeliness of this activity can be scripted for project and data management staff to monitor missing data and forms completion. Another example is the resolution of data clarification forms (DCFs) with site staff to insure that the tasks are completed within a timely manner and that data is as complete as possible during periodic statistical analyses. SAS software is one application that can be used to evaluate and analyze this operational data to manage and present this information to a variety of project stakeholders including site recruiters and staff, data managers, clinical monitors, and project teams.

To better understand the utility of using operational tables for risk based monitoring as well as other clinical trial management efforts, this paper will first discuss the multicenter trial process, the principles behind risk-based monitoring, the features of eDC applications, the type of operational files useful for such reporting, and the types of reports that can be developed. Some real case examples will then be discussed that reflect the types of trial

management reporting that can be tackled with operational files. Finally some basic SAS tools will be mentioned that can assist with trial management and risk-based monitoring.

OVERVIEW OF MULTICENTER TRIALS AND DATA CAPTURE PROCESS

Multi-center trials are necessary to garner the large sample sizes and sufficient statistical power needed for adequate analysis of research data. Obtaining such large numbers can be particularly difficult when the disease is rare, or when phase III and IV pharmaceutical trials are needed for FDA approval. Utilizing multiple sites for patient recruitment can boost sample populations but can be a challenge to manage in order to insure timely recruitment and data capture. Multiple patient visits are often needed, particularly for long term trials and the growing need to assess chronic adverse events. Ultimately this will require data capture using many CRFs, over long increments, across many recruitment sites for adequate trial management.

Multiple CRFs will be required per visit during the course of a trial for different purposes. Some examples of the type of collected data are: participant demographics and characteristics such as gender, race, age, social factors, etc.; assessment of efficacy measures that meet trial objectives at baseline and throughout study; tracking and monitoring of safety markers particularly adverse events and serious adverse events; documentation of study treatment regimens and drug compliance; and monitoring of deviations and participant status. Forms may also be designed to capture information on secondary objectives like pharmaceutical use, physical exam results, laboratory measures, medical history, quality of life, and questionnaire measures. Most CRFs are required to be collected at study visits but some may be designated as 'optional' completion. In certain cases a required CRF may not be completed because of logistical issues during the participant study visit. A mechanism may exist to identify the CRF as not being collected within the visit window. A CRF may be triggered for submission by another event such as the completion of an adverse event form or other safety marker, or based on a clinical outcome response. Some CRFs are not visit specific and are only required based on events in the trial. Protocol deviation or study termination forms are good examples of CRFs that are not specific to a scheduled visit window. Adverse event forms are also a case that could be designated per study visits or be identified as optional collection based on presence of adverse experiences during the trial. Lastly there is the chance of unscheduled events that occur between protocol specified visit windows. Many eDC tools offer mechanisms to incorporate unscheduled events with designated CRFs between study visit windows.

Managing multicenter trials, with various visits and numerous case report forms, requires frequent monitoring of form and data quality to insure adequate and timely capture of clinical data throughout a study. This must occur throughout the general trial timeline of participant visits which ranges from initial screening visits, enrollment and baseline visits, periodic follow-up visits, and ultimately study completion and freezing/locking of data for analyses. Reports of performance metrics can be useful in gauging such activities to insure accuracy of data from participant visits and throughout the course of a trial. One of the most basic reports is an assessment of participation enrollment and completion based on pre-defined recruitment objectives for the trial. Furthermore there are additional trial activities that require periodic assessments such as site enrollment and data completion, data management reviews, source document verification, trial audits, collection of electronic signatures from investigators, and the freezing and locking of study data.

RISK-BASED MONITORING

The International Conference on Harmonization of Technical Requirements of Pharmaceuticals for Human Use (ICH) guideline for GCP E6 provides industry guidance on the monitoring of clinical trials. As stated by ICH E6, 'the purposes of trial monitoring are to verify that: the rights and well-being of human subjects are protected; the reported trial data are accurate, complete, and verifiable from source document verification; and the conduct of the trial is in compliance with the currently approved protocol/amendments, with GCP, and with the applicable regulatory requirements'.³ Clinical trials monitoring at the site level has been primarily focused with on-site monitoring, which involves direct interaction with site personnel in clinical trial recruitment activities. This can take on many roles including assessments of protocol compliance, safety monitoring, source document verification, and assurance of data completion and correctness. Most of these efforts can also be achieved through centralized monitoring of site activity to identify high-risk sites that require action.⁴ This approach uses defined metrics from data capture tools used to monitor the integrity of both safety and outcome data.

The FDA drafted an industry guidance document in 2011 that outlines a risk-based approach to the clinical monitoring process. This risk assessment model focuses on critical study parameters and utilizes both centralized and on-site monitoring efforts. The goal is to enhance human subject protection as well as the quality and integrity of captured data. The strategies outlined provide sponsors with methods to focus on the most critical data elements to stated research goals, and thus conduct clinical investigations more efficiently. The approach works to improve

efficiency and costs of clinical trials by reducing the need for frequent on-site monitoring and 100% source document verification. Instead the risk-based approach is more targeted and uses centralized monitoring of site activity to evaluate site performance throughout the study on many aspects of trial activity.⁵ One part is the assessment of data errors that occur at the site level such as missing data, data errors, protocol deviations, etc.. On-site monitoring is still included but may be less frequent and more geared to work with problematic sites. The approach is also risk-based and should consider factors related to the type of data collected. Some parameters for risk assessment include: complexity of study design, safety and outcome measures, the type of study endpoints (objective measurements versus subjective ones), the clinical complexity of the study population, geographical variations in medical practice, experience of the clinical investigation team, safety profile of the study drug/device, quantity of data, and the clinical relevance of the captured CRF data.⁶ The outcomes of centralized monitoring review will dictate the corrective actions required to reduce impact on study quality, such as communication of problems with site staff, re-training, increased monitoring activities, and possibly placing sites on probation.

BASIC FEATURES OF ELECTRONIC DATA CAPTURE TOOLS

The electronic data capture (eDC) approach is a vast improvement from earlier paper data capture where participant visit data were collected at sites on paper case report forms (CRFs) and then shipped to coordinating centers for processing. Double data entry was used to enter data at a coordinating center, followed by running routine data checks, or data clarifications (DCFs), to flag potential data problems. Listings of DCFs were returned to sites for resolution. Conversely the eDC web based model is a more direct approach where collected research data is entered remotely by site staff into electronic based case report forms (e-CRFs). Front end user interfaces (UIs) are designed with participant visit structures containing e-CRFs. Most DCFs can be designed to trigger upon data entry rather than waiting for periodic data assessments of checks at coordinating centers. The advantage of the eDC approach is that the information is made available to all stakeholders of the trial within real-time. The process removes the costly and timely approach of back and forth mailing of CRFs, data entry into in-house data systems, and the processing of DCF scripts to check for missing data and data errors. Because of the real time nature of data capture and the resource reduction over paper methods, eDC systems have become the standard approach within the pharmaceutical industry for management of clinical trial data.

Commercial eDC systems are designed to establish user roles and privileges based on the staff utilizing the tool and their designated responsibilities. Many systems are customizable with regards to assigning system user roles and allowing coordinating centers to establish their own user definitions. An organization can define the different user roles and assign their tasks within the system. Access privileges can be tailored to specific site views for site staff or can be granted to all site forms for more global center activities. As an example, site staff would have the ability to enter and correct data in e-CRFs only for their site, while clinical monitors would have the ability to view all forms across sites for source document verification. Further down the process, data managers would have the ability to monitor incoming data and assess DCFs for all sites, which would then require review and response at the site level. Coordinating centers can establish security rules for access to the system's interfaces through password-protected accounts to limit entrance to only approved users. Most modern eDC tools are also designed without the need for client side software and can be viewed by a variety of internet browsers providing greater flexibility for implementation across different platforms.

Because of the ongoing entry of clinical trial data in a research setting, most captured data is dynamic and constantly changing as different users perform assigned responsibilities within the interface. All the data captured via the front end interface is maintained in a relational database, often SQL based, but other relational database systems may be used. Data is normalized in the database tables for efficiency, though de-normalized views may exist of captured data by form. Such views may also provide a variety of operational data on system activities or summarize contents of aggregated reports. The captured clinical data can be corrected during the trial, usually as a result of DCF resolution or source document verification. Systems are designed to maintain an audit trail to reflect who, when, and how data changes were made to insure 21 CFR 11 compliance. Data edits can be viewed on screens but can also be accessed from database tables.

Depending on the eDC system a number of backend tables may be available that can be immensely important for risk-based monitoring of study performance and for general trial management. More eDC systems are expanding reporting capabilities directly in their front end user interfaces to facilitate these types of data management activities, by establishing pre-defined reports based on user specifications and by adding customizable reporting tools. These tools may only cover a portion of the reports deemed necessary by study management teams or may not have the metrics prescribed by the study protocol. Access to these database files is thus imperative in order to design customized reports of study metrics.

EXAMPLES OF USEFUL BACKEND DATA TABLES FOR TRIAL MANAGEMENT

A number of data tables can be useful for risk-based monitoring and trial management that go outside the captured clinical data. It should be noted that database tables will vary with different eDC systems. Each software tool will have different management applications, database architecture, and data tables for capturing and storing trial data. Use of operational tables will vary based on the structure and architecture of a center's chosen eDC system. From our experience, the following types of tables or views are likely to be present and available in some form and can assist in these efforts:

- Site Tables – Tables that capture designated site information including site keys, site names, site IDs, etc.. This information can be useful for enumerating or presenting activities by recruitment sites.
- Participant Status Tables – Tables that capture participant status or disposition throughout the course of the trial. Definitions are dependent on the naming convention used by the application but could include such levels as screening, failed screening, enrollment, discontinuation, and completion.
- Form Status Tables - Tables that capture information at the form level such as e-CRF submission and flags that reflect form status levels captured during the course of the trial. Terminology is dependent on the eDC tool, but at its most basic level would include a status for blank or missing CRF versus a partial or completed CRF submission. Form specific tables are likely to include other CRF specific activities such as flags to mark auditing tasks, source document verification by clinical monitors, data reviews, and capturing of electronic signatures.
- DCF Status Tables – These tables capture information about data clarifications (DCFs) that fire within the system or are manually entered into the tool, and are tracked till the point of resolution. The information would include dates of DCF actions and the individual responsible for the action, the reasoning behind the DCF, the form and question associated with the form, and metrics about the length of time, or age, of the DCF on the forms.
- System Account Tables - Tables that manage account permissions by user and role based permissions defined in the system. Information could include usernames, user domains, account status, and data information on users for system management (such as contact information and e-mails for notifications).
- Form Versioning Tables – These tables would capture information on form versions and when new versions are uploaded and archived by the eDC tool. This would include metadata on form versions as well as dates the forms were published to the system for use during the trial.
- Master Data Files – These are tables used in the capture of trial data at the question level. While this is drilling down to the collected research data, the tables reflect important audit trail data. They may also contain information on whether the data question was marked as not collected at the site, the presence of a DCF, the existence of a question comment by field staff, or other question specific eDC functions. The table structure is likely normalized with a record for each data value entered into the field by patient, visit, and form.

REVIEW OF INFORMATIONAL REPORTS FOR TRIAL MANAGEMENT

Based on the tables and views provided in the backend of an eDC system, a coordinating center can generate a variety of reports for different purposes and study stakeholders. Reports should be designed depending on the targeted audience involved with the review of a trial including project and data managers; site staff such as principal investigators, research coordinators, and assistants; clinical monitors; and pharmacy and safety staff. Some of the most frequent reports are those for management teams and business stakeholders that evaluate study enrollment, assessments of recruitment goals, and timing of trial completion. Another type of report is centralized monitoring metrics that assess site activity including recruitment, CRF completion, and DCF resolution. Reports for site staff on these activities are essential for ensuring field recruiters are cognizant of completion tasks. Reports are also helpful for clinical monitors and data auditing teams to track and facilitate data and form reviews.

Beyond the audience type and operational needs, careful consideration should be given to the structure of reports and report measurements. Various report structures can be useful depending on the management activities. Listings, for instance, are ideal when the need is to communicate specific completion tasks for trial projects. Examples would be listings for clinical monitors of form review at a site or a summary of expected CRFs for site staff. Aggregate reports with summary measures of trial metrics can be useful to evaluate recruitment goals or

monitor trial activities across teams. Form completion rates per site, for instance, are aggregated measures that can greatly assist site staff on their data activities and help managers insure target goals are met.

EXAMPLES OF TRIAL MANAGEMENT BASED ON EDC OPERATIONAL TABLES

In this next section, the discussion will focus on a few case examples where backend operational tables are used for eDC trial management and risk-based monitoring of site activities. The examples relate to a specific eDC tool but the goal is that they can be extrapolated to other systems that offer similar database structures. The discussion of these approaches will highlight objectives with the operations involved in trial management which should be considered when developing processes internally. Three examples are discussed: 1) site and management reports regarding CRF completion, 2) site reports on DCF aging, and 3) management reports on data audits.

CASE REPORT FORM COMPLETION RATES

From a risk-based monitoring perspective it's important to have clinical trial data entered into e-CRFs in an expeditious manner to have the most complete trial data available for analyses. Defined endpoints can be established as metrics that site staff should adhere to and project management staff can periodically evaluate. The ideal scenario for site completion of form data is direct entry into CRFs during the visit while the information is being gathered from the participant. In our experience, most site staff capture the information initially on paper and transcribe into e-CRFs at a later time, hopefully soon after the participant visit. An acceptable window of time for form completion should be defined and communicated to staff to insure adequate completion rates. This time window will vary by protocol, and maybe by type of CRF or type of participant visit.

Reports on e-CRF completion were established for two purposes. One was to provide site staff with information on the CRFs that were outside an established time limit. In this case, 14 calendar days was considered an acceptable time frame to enter participant data into e-CRFs. Forms outside this window would appear on generated report listings. The grace period was a flexible parameter that could easily be altered depending on the protocol needs or if there was a need to complete all forms prior to a periodic data freeze. The second purpose was to provide project and data management teams information on the percentage of forms completed based on the number of expected forms and designated review points. Expected totals were based on the date of enrollment for an individual participant and the expected date of a study visit. These reports not only provided the rates of completion by site but also stratified these numbers by visit window, e-CRF, and a combination of both factors to determine if these contributed to completion rates.

To generate reports on e-CRF completion our team was able to capitalize on a table that provided a number of descriptors to classify various states of e-CRF tracking. The form status table included fields that captured form level activities such as: form entry and submission, flags for data reviews (such as source document verification or data management reviews), the status of electronic signature collection, and periodic freezing and locking of data. Form submission status and date of submission were two operational fields that were useful for the generation of this report. Primarily the form was classified as 'Submitted' when data was entered, as a 'New Entry' when the form was empty, or 'Omitted' if the form was marked by the site as not collected at the field. The 'Omitted' status level occurred if the form couldn't be completed because of logistic reasons. Completion of the form was based on turn over to 'Submitted' or 'Omitted' status at the date that the form was expected be filled out.

Site based reports acted as an alert mechanism to prod site staff into e-CRF completion. Even though the eDC user interface provided an easy to follow visual display with graphical cues of visit schedules and completed e-CRFs, the sense was that sites were less responsive to these visual summaries and required periodic reminders on form completion. Notifications included a listing of outstanding e-CRFs by participant ID as well as rates of completion at their site to assess their success. In general, report alerts were e-mailed to site staff on a bi-weekly basis, although the reports were scheduled to run daily after updates of clinical trial data sets. These daily reports were ideal for management staff to review current activity. In addition it allowed for immediate notification to sites during periodic data freezes and statistical analyses. At the management level, aggregate reports on the actual rate of form completion were an ideal metric for assessing site performance level during the course of the trial.

With regards to form completion, consideration should be made by coordinating centers as to frequency of reminders to site staff of missing e-CRFs as well as to the amount of time or grace period for form completion. Parameters should be defined by management staff as to how and when sites should be alerted regarding site delinquency with regards to this activity. For instance, such metrics could consider percentage of timely completion of forms versus late completion or missing form rates. An example is having no greater than 10% of forms be incomplete after the designated grace period. Another approach is graphing expected form completion rates by time during the course of the trial.

DATA CLARIFICATION STATUS TABLES

A similar risk-based monitoring metric is the evaluation and assessment of Data Clarification Forms (DCFs) by site staff. DCFs are quality data checks that alert site and coordinating center staff of possible errors or anomalies with entered data. They can be scripted within the interface to fire upon data entry or form completion depending on the complexity of DCF and the inter-relationship of the check to other data elements. A value that's outside a specified range check, for instance, can be flagged as problematic at entry allowing the user to correct the value. During form submission, problematic data values trigger the generation of an official DCF record to backend tables. At this point a response is required from the site, and possibly a review from the coordinating center to resolve.

DCFs generally reflect two scenarios, one being that the data was entered incorrectly and requires data correction by the site. The other is a situation where the data is correct but is flagged for being different than what is expected. A range check for weight value might be scripted to prevent data entry errors, but the limit defined could flag a correct data point. In such cases a response, or answer, from the site is expected with review by coordinating center staff.

From a risk-based monitoring approach, the objective is to have sites correct and/or resolve DCFs in a timely manner, hopefully during the entry process or at least once the DCFs have been fired. The likelihood of immediate DCF resolution maybe low, so it behooves coordinating centers to insure data corrections or site responses are done quickly. An ideal metric for assessing timeliness of DCF correction is to evaluate the DCF aging time and make efforts to keep these values low across sites. The calculation is based on how long an individual DCF has remained open without data correction or site response. The plan was to produce a SAS report that provided a list of all outstanding DCFs by site. As with the e-CRF, a summary report of total DCFs and DCF aging was important for management staff.

The eDC tool in use provided a DCF interface that included a calculated aging value. Through a designated DCF tab, a user could view the number of DCFs for sites along with information about DCF aging and average aging that could be specific for patient, visit, and form combinations. In addition, there was a backend view that listed all DCF records along with their defined status levels that included the calculated DCF age. However, the aging values tended to be skewed with large aging values for some DCF records and zeros for others. Upon inspection it appeared that calculations in the system were based on the point when a site coordinator physically opened and viewed each DCF record. Thus records that were observed but never reviewed at the interface level remained at zero and provided lopsided totals and averages. In addition, the metric reflected time to complete DCF resolution. Those answered by the site that required coordinating center resolution increased in age and didn't accurately reflect the actions at the site.

To correct this metric the study team generated a new aging value that was site specific and was based on the length of time that the DCF had been in the system from the time of automatic firing. Different status levels reflecting the DCF business process, along with dates of occurrence, were identified from a backend view and used in the development of the new aging calculation. The new aging rates were more consistent per site and were considered better markers for assessing resolution of data errors.

In this use case, information from the backend tables was used to calculate metrics that the study team felt better reflected the true nature of site response to DCF resolution. Careful consideration should be given to calculating DCF age that best reflects the response of data resolution at the site level. While not conducted, a second aging metric could have been drafted on DCFs not resolved at the coordinating center level. In this case the calculation would be based on DCFs that require data management resolution. With regards to summary metrics, further evaluation of DCF aging could enhance risk-based monitoring efforts. For instance reports that aggregate DCFs by total unanswered and by percentages with lengths longer than set times (i.e. 15 or 30 days) could improve site assessments.

MASTER DATA FILE

The clinical data captured at the question level can be important for generating reports that reflect data edits and audit information. Referred to in this paper as 'Master Data Files', these are normalized database tables with records that represent all entered data values, including not only initial entries but also data updates. Of value is information on when a data change was made and by whom. Reasons for a data change are also captured which can be a coded response and/or a text field to describe the edit. Because these files are at the question level, the size of the datasets may be in the millions of records. Programming efficiency is recommended when using such tables. For example, one can gain performance by using proc sql commands to extract tables or join records.

For this use case, the study team required reports of all CRFs containing a data audit and a list of the audit changes. Flagging participant CRFs with data audits could be useful to assist on-site monitoring staff in reviewing data edits made in the field by site coordinators. Records in the master file are indexed by a patient data key that is unique for each patient, visit, form, and question. From there one can flag records that have an edit status that indicates a data change. In this case, the system reason field is either 'Initial Entry' or 'Data Change'. Other reason codes exist but for this use case these two levels are considered relevant. Once identified the data changes can be sorted by participant, visit, and form for by site report generation.

This scenario reflects a trial management case where audit trail data is extracted from tables that capture information at the individual question response level. Access to master file tables can provide numerous functionality in trial management activities. Another use case is flagging participant records that changed during the course of a trial, providing useful information to statistical teams on updates that occurred since previous analyses. From a risk-based monitoring perspective, identifying frequent CRF edits can also identify sites that are having problems with CRF completion that require assistance and/or re-training.

SAS TOOLS FOR REPORTING ON EDC OPERATIONAL TABLES

Backend eDC operational tables can be used for many elements of clinical trial management and risk-based monitoring activities by a coordinating center. The third part of this paper focuses on some basic SAS tools that can be used for accessing and reporting this information for the intended audiences. Four concepts are discussed including: 1) access and management of relational database tables, 2), ODS output methods to present the material for various stakeholders, 3) use of macros to facilitate reports for multiple sites, and 4) a modular report writing approach that can summarize site activity.

ACCESS TO SQL RELATIONAL DATABASES THROUGH SAS

Electronic data capture systems are built on a relational database framework. In our case the foundation of our eDC tool was a MS-SQL platform. One possible solution to accessing these files is through export and import of intermediary CSV, XML, or MS Excel files. Chron jobs could be established to export the data into these formats with subsequent code to import the data to generate analytical SAS files. This solution while doable requires more programming to read in files from a delimited file formats and is not the most efficient approach. A better access method is through an ODBC connection using SAS/ACCESS license. This setup can be established in either PC or Linux environments. The Open Database Connectivity (ODBC) application provides an interface and wizard administrative tool to assign a Database System Name (DSN) that can be used to link one's SAS environment and a SQL database platform. The DSNs can be established with windows authentication using network login or through SQL authentication if the tables are password protected. Because of the confidentiality of the captured clinical data our business stipulated a password protected environment for data access.

There are two basic approaches for connecting to the study database once an ODBC connection has been established. The first is a data connection through a libname declaration (fig. 1). In this approach the libref is named mydblib and the syntax 'ODBC' must follow the name to inform SAS of the connection type.⁷ The ODBC DSN was defined as 'Study_VA101010'. In defining the ODBC connection through the wizard, the SQL server location and database name must be declared (in this case a SQL table listed as VA101010 on a remote server location). Since this is a password protected SQL database, the libname statement incorporates userid and password (in the example the user was defined as 'batch_service' and the password was 'X'ed out). Further efficiency can be obtained by using proc SQL commands because of the size of the operational tables.

Fig. 1

```
libname mydblib odbc user='batch_service'  
pwd='XXXXX' dsn="Study_VA101010" ;  
  
proc sql;  
  create table edc1 as  
  select *  
  from mydblib.mf_site  
  ;  
quit;
```

The second approach, referred to as a Pass Through Facility, places the ODBC connection reference within the SQL code as opposed to a separate library reference. The benefit of the pass through approach is that it allows one to directly pass SQL language commands to the SQL server to generate a SAS dataset. This utilizes more SQL settings such as indexing that may be more advantageous, but the choice to utilize the pass through facility

over a libname ultimately comes down to ones comfort with SQL commands.⁸ In the Figure 2 example, there are similar syntactical elements such as dsn, user, and pwd that are defined within the ODBC connection or within SQL server. The difference reflects the reference to the ODBC connection within the SQL statement, which states 'connect to ODBC as my-ODBC'.

Fig. 2

```
proc sql;
  connect to odbc as myODBC (dsn="Study_VA101010"
  user='batch_service' pwd='XXXXXX' );
  create table edc2 as
  select * from connection to myODBC
  (select * from mf_site order by siteid)
  ;
quit;
```

OUTPUT DELIVERY SYSTEM FOR GENERATING CUSTOM REPORTS

The SAS Output Delivery System is a remarkable tool for generating professional style reports in a variety of output formats including HTML, PDF, and MS Word (via Rich Text Format (RTF)). The various styles and the ability to customize reports can enhance risk-based monitoring efforts with project stakeholders. HTML reports, for instance, are an ideal format to submit to site staff since they can easily be accessed through any web browser.

Returning to the case example of missing CRFs, it was important to alert sites about delinquent form completion but also to provide summary reports for project and data management staff. We were able to utilize a backend table from our eDC system that captured status levels for each participant CRF record. Flags were generated to identify expected form status. Frequencies based on these were used to generate a table that listed each site and the number of expected, completed, and missing forms along with the percentage missing to gauge completion rates.

Fig. 3

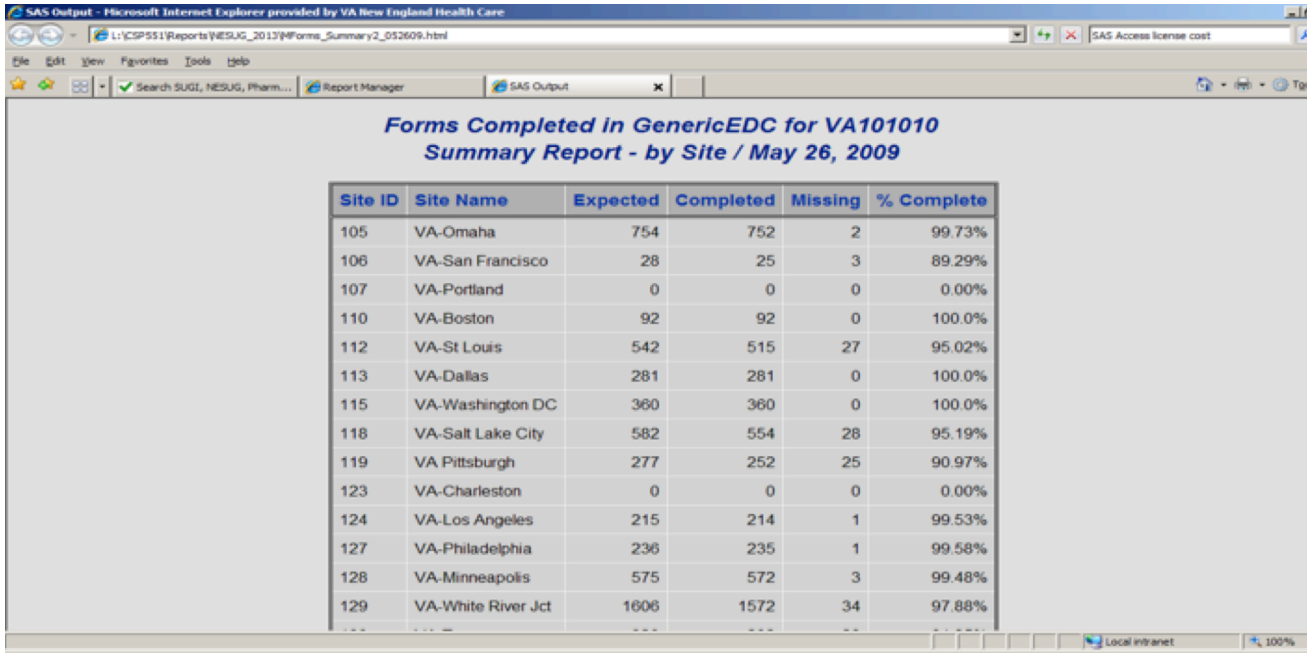
```
ods listing close;
ods html file="B:\VA101010\Reports\
MForms_Summary2_052609.html";
proc print data=sitex4 noobs label;
  var siteid sitename exp act miss pct ;
  title1 "Forms Completed in GenericEDC for
VA101010";
  title2 "Summary Report - by Site / May 26, 2009";
run;
ods html close;
```

The ODS statements used to generate an HTML report of

expected CRF completion rates are presented in fig. 3 with the generated report in fig. 4. This basic ODS structure requires opening the ODS for HTML format and designating the folder location for the file ('ODS HTML file= ___'). The 'ODS listing close' / 'ODS Listing' statements bracketing the code allow for shutting off the generation of an output listing and returning to default settings, thus reducing system resources in the SAS runs.

This demonstrates the essential foundation ODS programming code. For more customizable reporting one can expand this basic structure to include more stylistic features, greater flexibility in placement of metrics, and different output formats.

Fig. 4



Site ID	Site Name	Expected	Completed	Missing	% Complete
105	VA-Omaha	754	752	2	99.73%
106	VA-San Francisco	28	25	3	89.29%
107	VA-Portland	0	0	0	0.00%
110	VA-Boston	92	92	0	100.0%
112	VA-St Louis	542	515	27	95.02%
113	VA-Dallas	281	281	0	100.0%
115	VA-Washington DC	360	360	0	100.0%
118	VA-Salt Lake City	582	554	28	95.19%
119	VA Pittsburgh	277	252	25	90.97%
123	VA-Charleston	0	0	0	0.00%
124	VA-Los Angeles	215	214	1	99.53%
127	VA-Philadelphia	236	235	1	99.58%
128	VA-Minneapolis	575	572	3	99.48%
129	VA-White River Jct	1606	1572	34	97.88%

MACROS TO FACILITATE SITE REPORTS

As part of risk-based monitoring efforts, it was important to script individual site reports of missing CRFs as a means of alerting staff of delinquencies in CRF completion. One way to accomplish this was to create a macro that looped through the form status dataset using a list of site identifiers. Variables were scripted that flagged whether the CRF was expected based on the visit window, whether it was completed based on the submitted form status, and whether it was delinquent by these factors. The report provided a listing per site of all forms missing by participant and the number of days the form was overdue. It also included the total number of forms unaccounted for that needed to be entered into the system by the site.

The macro used a %do-loop that was generated for the range of sites in each study. It created datasets parameterized by site key that included all delinquent forms outside the defined grace period. The data step outputted

macro call statements such as site ID (**siteid**), site name (**siteNm**), and total number of delinquent CRFs (**srec**) that were unique for each site report. Since the site number varied based on the timing of site launch, the macro had to account for an ever growing number of sites. A site list table from the eDC system was used to identify total site keys, which allowed flexibility for changing site numbers and the ability to use the program across studies (see fig. 5). The final site number was outputted as a “call symput” function that could establish the bounds of the macro run (**siterpt**).

Fig. 5

```
proc sort data=mydblib.mf_site out=rptsum1;
by sitekey;

data _null_;
  set rptsum1 (keep=sitekey) end=final ;
  by sitekey;
  if final then do;
    call symput('SK',put(sitekey,2.));
  end;
run;
```

The macro also had to take into account site datasets that had no records. This could happen because the site had no delinquent CRFs or no enrolled participants at the time of report generation. A macro variable had to be available that reflected the presence or absence of form records. The **srec** macro variable, which captured the number of delinquent CRFs, was an ideal candidate that could be used to trigger the report with a %if, %then statement when data was available (%if &srec. >=1 %then %do;). However the dataset was only generated when site data set existed in the dataset. In other words, an empty dataset would not output a new **srec** macro of zero (0) but would instead reflect

the dataset from the previous site. To work around this, the **srec** macro was reset to zero (0) in a `data _null_` statement that ran only when the table was empty to clear out macro elements for the site report. A portion of the macro is presented below:

```

%macro siterpt;
%do I=1 %to &SK;
*   Pulls out Data by Site   ;
data rptx;
  set formflg_all;
  where sitekey=&I. and d >= 1;
  by sitekey;
  count+1;
  if last.sitekey then do;
    call symput('Site',put(siteid,$3.));
    call symput('srec',put(count,3.));
    call symput('SiteNm',put(sitename, $30.));
  end;
run;

*   Resets Macro Vars if Empty Dataset   ;
data _null_ ;
  if 0 then set rptx nobs=howmany;
  if howmany < 1 then do;
    siteid=""; sitename=""; count=0; reportdt=today();
    call symput('Site',put(siteid,$3.));
    call symput('srec',put(count,3.));
    call symput('SiteNm',put(sitename, $30.));
  end;
run;

* Run only for datasets with records (no empty datasets) ;
%if &srec. >=1 %then %do;

*   Missing Form Reports by Site   ;
ods html file="&rpt.\MForms_&site._&rptdt2..html";
proc print data=rptx2 label;
  var eventlabel form fu_frmdays ;
  id patientidentity patientstatus;
  by patientidentity patientstatus;
  title1 "Forms Not Completed in GenericEDC for &study.";
  title2 "Site:  &Site.-&SiteNm";
  title3 "Date Reported: &rptdt";
  footnote1 "Currently there are &srec. Forms Unaccounted for at this Site";
  footnote2 ;
  footnote3 "Please Complete Missing Forms for These Patients or Mark CRFs
Not Collected in GenericEDC";
run;
ods html close;
%end; %end;
%mend siterpt;

```

Ultimately this macro %do loop provided a mechanism for individual HTML reports to be generated automatically for each site. This saved effort in manual generation of by site reports where the number of sites could change.

CUSTOMIZED REPORTING OF STUDY BENCH MARKS

A key feature of study benchmark reporting is repeatability, which makes macros an important programming tool. Copying and pasting for study reports introduces extensive effort and human error into the process.

The macro described below represents a framework for aggregate report generation. Given the repeated nature of these reports, we define important features in a table format data set and use a macro that has modular components which define various statistics that can be generated.

```

data tabformat;
input varname $14. order @18 varlabel $14. @34 type $2. +2 fformat $8.;
if fformat eq ' ' then fformat = '4.';
if _n_ gt 3 then varcat = 'DCF Status';
else varcat = 'DCF Age';
call symput('maxcount',_n_);
datalines;
aging_s_x      1 DCF Age Mean      n1
aging_s_x      1 DCF Age Median  n2
dcfagecat      2 DCF Age         c2   dcfagef.
dcfstatus      3 DCF Status      c2   dcfstaf.
;
run;

```

In the case of DCF reporting, one record per DCF would be expected as input, and the format of the table and requested statistics is defined by another data set with the following variables:

VARNAME = row variables to be included in the table

ORDER = sort order for variables in the table

VARCAT = category of row variables [must exist, but may be blank]

VARLABEL = a label for the VARNAME as it should be shown in the report

TYPE = type of variable. n1 = numeric [Mean (SD)], n2 = numeric [Mean (IQR)], c1 = dichotomous [N (%)] or c2 = multilevel category [N (%)]

FFORMAT = Format for multi-level variables (TYPE = c2) code missing formats to '4.'

&MAXCOUNT = number of variables in the table, defined as a macro variable and automatically calculated using the CALL SYMPUT function.

The macro **%tab1mac_doc** requires the following parameters:

RDSN = the table format dataset you created above with the variables and table structure

DSN = the dataset name containing the report data

BYVAR = (optional) the variable that defines the groups of columns (must be numeric)

COLVAR = the variable which defines the columns within the BYVAR (must be numeric)

LIBOUT = the defined libname for the ODS output tables to be used by PROC DOCUMENT

FILEOUT = the name and path location of the Excel file

Significant features of the macro are presented below. The macro begins with a parsing of the formatting data set, where one loops through the records, where each record from the formatting data set has instructions for a particular variable (code presented below). This information is assigned to macro variables, which will guide the macro steps deciding which modules will be used. This is followed by modules that define and create the different statistics that can be generated.

```

%do count = 1 %to &maxcount;
  data tmpk1; set &rdsn;
  if _n_ eq &count;
  call symput('vcat',varcat);
  call symput('rf',varname);
  call symput('vartype',type);
  call symput('varlab',varlabel);
  call symput('vorder',order);
  call symput('fformat',fformat);
run;
  Module 1, Module 2, etc.
%end;

```

In the case of a numeric variable, where the mean and standard deviation are requested, the module is conditional on **&vartype = n1** (as shown below). After all variables are processed, the resulting data sets are then modified to create a single variable which will show up in the final table.

```
**** for Continuous variables (mean and sd)****;
  %if &vartype eq n1 %then %do;
    proc means data=&dsn noprint;
      by &byvar &colvar;
      var &rf;
      output out= contout1 mean=mean std=std;
    run;
    data contout1; set contout1;
      length exposure $42;
      sorder = &vorder; exposurecat = "&vcat";
      exposure = "&varlab";
    run;
    data results_m; set results_m contout1;
    run;
  %end;
```

In the case of means and standard deviations [as seen in second part of the code below], the mean and SD are first rounded and then are concatenated to form a single variable [**freqpercent**].

```
data results_m; set results_m;
length freqpercent $30;
  mean = round(mean,.01);
  std = round(std,.01);

freqpercent= cat(mean, ' (', std, ')');
```

The next step uses PROC REPORT to generate the final table using ODS to save it as a document data set. For this example, this section assumes nonmissing &byvar and &vcat options (code below). The creation of the **freqpercent** variable and this deliberately minimal use of PROC REPORT capabilities force the calculated statistics into a single table cell and reduces the number of columns produced, which would be present if we were using PROC REPORT's many features.

```
%if (&byvar ne ) and (&vcat ne ) %then %do;
ods listing; ods document name=&libout.&fileout (write);
proc report data=tabres nowd headline ps=130;
column exposurecat exposure &byvar, &colvar, freqpercent ;
define exposurecat/ group order=data width=55;
define exposure /group order=data width=45;
define &byvar /across order=internal format=byf.;
define &colvar /across order=internal format=colf.;
define freqpercent /group width=20;
run;
ods document close;
run; quit;
%end;
```

The final step is to call the macro and assemble the tables into a final document (code below). It can be called multiple times if there are multiple tables to produce. In this case metrics are outputted to Excel workbooks. This can be modified for any ODS output destination. The ODS tagset can be repeated for multiple spreadsheets in a workbook (under brown text).

```
libname tabout 'K:\DatabaseR_D'; ** Output directory for tables';
%tablmac_doc(tmp1,sample2, SitekeyX, SiteGrpX,tabout,Table1a);
run ;

ods listing;
  ods tagsets.ExcelXP style=listing
```

```

options (Sheet_Interval='proc' embedded_titles='Yes' Index='Yes'
Absolute_Column_Width='20,20,20'
Row_Heights='15,15,15,15,15,15,15')
file="O:\SAS\Users\VA101010\Reports\Tables.xls";

```

```

ods tagsets.ExcelXP options(sheet_name="Table 1a");
proc document name=tabout.Table1a;
replay ;
run;
ods tagsets.ExcelXP close;
run;

```

The Excel workbook (fig. 6) presents metrics relevant to centralized monitoring for the DCF aging case including: 1) age as mean, median, and category groups, and 2) percent DCF status levels (Open, Answered, and Closed).

Fig. 6

		Site Type					
		VA			Non-VA		
		Site Number					
		Site 1	Site 2	Site 3	Site 4	Site 5	Site 6
exposurecat	exposure	N(%)	N(%)	N(%)	N(%)	N(%)	N(%)
DCF Age	DCF Age Mean	(53.22)	(50.85)	(58.54)	(41.1)	(37.96)	(50.37)
	DCF Age Median	32 (2,59)	9 (0,48)	15 (0,58)	7 (1,21)	7 (0,30)	7 (0,28)
	DCF Age 0-15	346 (43.7)	536 (58.7)	802 (50.3)	845 (71)	670 (58.1)	298 (60.7)
	DCF Age 16-30	49 (6.2)	75 (8.2)	115 (7.2)	133 (11.2)	196 (17)	93 (18.9)
	DCF Age 30-45	84 (10.6)	64 (7)	153 (9.6)	88 (7.4)	83 (7.2)	27 (5.5)
	DCF Age 45-60	117 (14.8)	68 (7.4)	145 (9.1)	41 (3.4)	35 (3)	16 (3.3)
	DCF Age >60	195 (24.7)	170 (18.6)	378 (23.7)	83 (7)	170 (14.7)	57 (11.6)
DCF Status	DCF Status Open	49 (6.2)	31 (3.4)	26 (1.6)	24 (2)	11 (1)	10 (2)
	DCF Status Answered	0 (0)	5 (0.5)	4 (0.3)	6 (0.5)	12 (1)	4 (0.8)
	DCF Status Closed	742 (93.8)	877 (96.1)	1563 (98.1)	(97.5)	1131 (98)	477 (97.1)

CONCLUSIONS

Operational relational database tables from eDC systems can have many applications for improving risk-based monitoring and clinical trial management efforts. SAS software is an application that can be used to generate periodic reports of trial activity for site personnel and management staff. As presented in this paper reports of site activity can be generated based on defined parameters that reflect various business needs throughout the trial process. Some examples discussed include participant recruitment, visit and CRF completion, assessment of protocol deviations, and DCF resolution, to name a few. Aggregate reports of trial activity are important for management staff as part of the risk-based monitoring process. From such reports, decision rules can be established for training and corrective action if sites don't meet pre-determined protocol goals. These by site evaluations reflect the increased capability of centralized monitoring, reduction in on-site monitoring by coordinating centers, and, overall, a greater efficiency in multi-center trial management.

Modern electronic data capture software applications are building more flexible reporting features offering system defined reports and customizable reporting capabilities for centralized monitoring. In addition, SQL based reporting tools can be linked to eDC databases for real-time reporting. These approaches can be beneficial however, using SAS for clinical trial management reporting did provide many benefits for our coordinating center. There are many aspects to consider in selecting a tool for reporting. One is the programming expertise of the coordinating center. Our trials research center was an academic setting and had more SAS programmers than SQL experts. Complex reporting required greater SQL experience that could be better developed with our SAS knowledge base. We also found we had greater flexibility in generating protocol specific reports utilizing backend tables than using eDC reports defined within the software application, as in the example with calculating DCF aging. In addition, we could leverage off SAS ODS features and develop SAS macros to generate sophisticated

reports with more design flexibility. Another aspect is the need for real-time reporting. Though the described SAS approach did not generate real-time reports, we were able to establish near real-time reports through daily automated jobs that were sufficient for the type of centralized reporting needed for protocol management. We also found that an active approach to sending reports to sites impacted their ability to take action on data deficiencies as opposed to directing them to a reporting interface. Submitted reports like this made site staff more aware of deficiencies and more likely to complete the needed recruitment and data tasks necessary to meet study management objectives.

We found use of operational eDC databases with SAS effective in our coordinating center's efforts for centralized monitoring of site activity. These methods greatly improved interaction with site staff and enhanced efforts in the management of clinical trials. SAS ODS and other programming features gave the team greater capabilities in designing and scripting site reports. As one facet of risk-based monitoring the outlined approaches with eDC databases expanded our centers efforts in improving data quality and enhancing human subject protection.

REFERENCES

¹ Food and Drug Administration. 'Guidance for Industry Oversight of Clinical Investigations – A Risk-Based Based Approach to Monitoring', August 2011. Retrieved June 30, 2013 from <http://www.fda.gov/downloads/Drugs/.../Guidances/UCM269919.pdf>

² Ibid.

³ International Conference on Harmonization of Technical Requirements for Registration of Pharmaceuticals for Human Use. ICH Tripartite Guideline: Guidelines for Good Clinical Practice E6(R1). Retrieved June 30, 2013 from

http://www.ich.org/fileadmin/Public_Web_Site/ICH_Products/Guidelines/Efficacy/E6_R1/Step4/E6_R1_Guideline.pdf

⁴ Ibid. 1

⁵ Ibid. 1

⁶ Ibid. 1

⁷ Riley, Candice, Getting SAS[®] to Play Nice With Others: Connecting SAS[®] to Microsoft SQL Server Using an ODBC Connection. SAS Global Forum 2008, Paper 135-2008.

⁸ Ibid. 7

ACKNOWLEDGMENTS

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

Acknowledgements are also made to Erika Holmberg and Allan Lewis at VA CSP Boston Coordinating Center for providing background and review of this paper/presentation.

CONTACT INFORMATION

Contact the author at:

Bob Hall, MS
VA CSP Boston Coordinating Center / MAVERIC
VA Boston Healthcare (151-MAV) / 250 S. Huntington St.
Boston, MA 02130
Phone: (857) 364-6134
Email: Robert.hall9@va.gov

Complete macro information presented in this paper can be found at <http://people.bu.edu/gagnon/>.