Post-Collection Processing with Blaise in a Distributed Environment

Mary Laidlaw, Mike Rhoads and Jane Shepherd, Westat

1.0. Introduction

The processing, quality review, and delivery of survey data for a recent longitudinal study was particularly challenging due to the need to support multiple users in a distributed processing environment with highly structured levels of data access. As on most longitudinal studies, the need to re-field data swiftly to drive the subsequent protocol for each participant competed against the need to deliver clean and consistent data for analyses. Our data processing solution had to meet both requirements simultaneously. In addition, the data that required post-processing was collected in multiple modes: Blaise CAPI, Blaise CATI, and hardcopy forms scanned through data capture software. Reviewing the data with an eye towards consistency between instruments was essential due to the need to determine the appropriate next steps in the specific participant's protocol.

In order to meet the needs of this and other Blaise projects, we enhanced and integrated a set of Blaise-based tools that could be used for a variety of data processing tasks at multiple locations. This paper describes our experience with the implementation of the Blaise Data Processing and Quality Assurance system (BDPQA) we developed. We discuss: the initial requirements of the projects; the system and database architecture; the flow of data to and through the system; and the view of the tool from the users' perspectives. We conclude with some lessons learned from our experience.

2.0. The Post-Processing Requirements

Westat has used Blaise post-processing tools for many years. In recent years, a number of projects have required more extensive post-collection processing. Some of the requirements addressed by BDPQA include:

- The division of collection and editing responsibilities between a central office and distributed sites, each of which was responsible for the quality of its collected data.
- Multiple views of the data so that staff at the distributed sites could access only the data for that site's cases. The staff at the central office could access all collected data.
- The sequencing and division of post-collection processing tasks to be performed by the sites and the central office. For example, while interviewer field comments were most appropriately

reviewed and resolved by the sites which were responsible for that staff, coding of text responses was conducted by the central office in order to ensure project-wide data consistency.

- Two levels of review – the editor and the supervisor – with appropriate task authorization in order to ensure high data quality and the validity of updates.
- Documentation of all data decisions across data collection sites and the central office in a comprehensive Data Decision Log. Again, the site staff could view only the decisions that pertained to the site's cases. The central office staff would need to view the decisions across all centers.
- Integration of data processing for hardcopy form data. All editing of data collected via hardcopy would need to occur during post-data collection processing and would be documented in the data decision logs along with decisions about CAPI and CATI data.
- Easy update of the post-processing tool as new versions of hardcopy forms as well as the Blaise instruments and their data models were fielded. It was also required that the tool be adaptable to any new instruments added to the study protocol.
- An efficient flow through all post-processing steps at both the sites and the central office with ongoing support for all instruments. Because the progress of the protocol for each respondent depended on the likelihood and timing of medical events, the sequence of data collection would result in a flow of data from all instruments into the study's systems on an ongoing basis; there was no specific end date for the use of any instrument and the start up of the next phase of the study. Collected data was expected to complete processing by both the site and central office within three to four days of collection.
- Capability for a wide range of ad hoc and predefined reports to be used for multiple workflow and analytic purposes including evaluating the flow of the data through the system, the consistency of coding, the status of each case's data across instruments at any one time, etc.
- The processing approach needed to manage the storage of the original data file (data as collected and passed to any post-process) and all other data files generated as a result of BDPQA activities. All file versions needed to be immediately accessible for a period of at least one year in case of the need for additional processing.
- The process and flow of each instrument's data for each case (referred to as a production unit throughout this paper) must be independent of all others. To accommodate a variety of protocols for each household and respondent, the existence of one instrument in the BDPQA did not necessarily require the existence of another instrument in the system at the same time.

Given these requirements, Westat leveraged an in-house Blaise editing system as the base for a new distributed approach, BDPQA. The editing process within BDPQA centers on the review and resolution of decision log entries. Decision log entries are created for field comments, Blaise edits, and reported data issues. This paper addresses our experience with the initial version of the distributed processing BDPQA system.

3.0. BDPQA System Architecture

The BDPQA system consists of two primary components: a server-based batch process called Workflow Manager that manages the flow of data within the system, and a Windows desktop interactive application that is used by editors and verifiers. There are also two administrative tools. One of these (BES Setup) is used to generate an initial instance of the system and to manage the deployment of system updates, while the other (Manage Users) is used to control user access and roles within the system.

All system components are primarily written in VB.NET. The system makes use of numerous Blaise capabilities, including the Data Entry Program (DEP), Manipula, Blaise Datalink, and the Blaise Component Pack (BCP). We use Microsoft SQL Server 2005 for data storage, and utilize SQL Server triggers and stored procedures to implement such functions as change auditing. In addition to these primary technologies, BDPQA uses various COTS software to perform certain functions. These include Microsoft Access (generating reports), eDocPrinter PDF Pro (converting reports to PDF format), Adobe Reader (viewing PDF documents), and Microsoft Word Viewer (viewing audit trails).

Since the system was designed for access by widely distributed site-based staff, we thought about whether we should use Blaise Internet as the basis for the user interface component. We decided against this option for two primary reasons. First, as we have discussed in previous papers (Allan, et al., 2001; Dulaney and Allan 2001; Gowen and Clark 2007; and Frey and Rhoads 2009), Westat has evolved an in-house Blaise editing system over many years, and we wanted to take as much advantage as possible of that desktop-based user interface code. Second, while we have used Blaise Internet for several interviewing projects, we did not have any experience with it for data editing applications. Although we certainly did not rule out the possibility that Blaise Internet could be used as a suitable case review and updating tool for data editors, we did not have nearly enough time for research and development in this area given the tight time pressures of the initial project.

Since we decided to stay with a desktop client rather than implementing a Blaise Internet solution, we needed an alternate way to make the system available to site-based staff. Fortunately, Westat had already implemented a highly-secure Citrix platform for other project purposes. We were thus

able to implement the user interface component as an application on the Citrix desktop, which was then available to authorized site-based and home office staff.

3.1 Workflow Manager Component

The Workflow Manager component of the BDPQA application controls the flow of all data into and within the system. It is a nightly batch process, although it can also be run on-demand if special needs arise. Workflow Manager performs the following tasks:

1. Loads all newly completed Blaise interview data into interview data tables in the site-level SQL Server database
2. For all new cases, correctly names (by instrument and ID) and loads all acceptable audit trails into the system.
3. Updates the site's decision log table by going through all non-closed out cases in the interview data site tables and adding an entry for:
    a. all new interviewer comments found by invoking the GetRemarks method from the Blaise API
    b. all new errors found by invoking the Blaise API's CheckRecord method
4. Loads the interview data from all newly closed out site office cases into interview data tables in the central office SQL Server database
5. Loads the decision log site office table entries for all newly closed out site office cases into the decision log table in the central office database
6. Updates the central office decision log table by going through all non-closed out cases in the interview data central office tables and adding an entry for all new errors found by invoking the Blaise API's CheckRecord method

All of the data loading steps listed above are performed by Manipula programs, which take advantage of Blaise capabilities for accessing and storing data in multiple formats—native Blaise, SQL Server, and XML (which is used to bring in records from the data capture system). Workflow Manager also uses Blaise API routines to extract interviewer remarks and to identify any violations of Blaise editing rules in incoming cases.

3.2 User Interface Component

Section 5 below describes the interactive component of BDPQA from a user perspective in terms of its look and feel and its functionality. From a systems perspective, an essential element of this

component is its integration with Blaise through the use of the Blaise Component Pack (BCP). The system invokes the Blaise DEP both for data updating and for data browsing. The use of DEP ensures that the user of the system cannot make any changes to the data that violate the rules of the data model.

The user interface component also interacts with the SQL Server database tables that Blaise uses to store its data. We set up an insert trigger on each of these tables, which is activated whenever a data change is made (causing a new row to be inserted into the table). The trigger then invokes a stored procedure that compares "before" and "after" values to identify which specific data items were changed. For each difference that is found, we insert a new row into a separate table that includes the case ID, variable name, and newly-assigned value.

3.3 Data Storage

All data within BDPQA is stored in Microsoft SQL Server. Typically three separate databases are used: one for each of the two phases of editing (site and central office), and a third database that contains various types of metadata that are used by the system. This is configurable at the time an instance of the system is created, so that all of this information could be stored within a single database if desired.

Within the site and central office databases, some of the tables are effectively "owned" by the Blaise Datalink component. Workflow Manager uses Blaise to initially load case data into the system, and the Data Entry Program that is invoked from the interactive component of the system uses the data in these tables to replay and update cases. These tables are read by non-Blaise modules of the system for report generation and other purposes, but they are written to only by Blaise. Blaise Datalink offers a number of data storage options, which it refers to as data partition types, and it also provides a capability it refers to as "generic" storage. For BDPQA, we use the Flat Blocks data partition type. We also selected the generic data storage option so that we could take advantage of versioning. This data storage structure is described in much more detail in an earlier paper (Frey and Rhoads, 2009).

In addition to the tables used by Blaise Datalink, the databases also contain numerous other tables that are managed by the system entirely outside of Blaise. These include tables that are used to store and manage decision log entries, maintain user information and log system access, and hold various items of metadata that are used by the BDPQA system.

3.4 Access Control and Security

The BDPQA implements a highly granular user access control model. Access is controlled by site, by instrument, and by function. For example, some users may be provided with read-only access so that they can view case data but not change any values, and only designated users may be allowed to view

Blaise audit trails. User access information is stored in SQL Server tables, and there is an interactive interface module (Manage Users) that allows the system administrator to add and remove users and to modify access rights. System users can access the SQL Server database only through the interactive component of the system, which enforces the access constraints described above. As an additional precaution, the case data from each site is segregated into a separate schema and set of tables.

The BDPQA can operate within an overall environment that provides additional security measures. For instance, the system can be reached only through a Virtual Private Network (VPN), and access to it requires the use of two-factor authentication.

3.5 Configuration and Customization

BDPQA includes a setup tool that is used to generate an initial instance of the system and to manage the deployment of system updates. This tool, combined with a set of initialization (INI) files, provides for identifying the necessary network data locations, SQL Server databases, Blaise data models, and sites. The primary function of this module is to create the necessary schemas, metadata, BOI files, instrument tables, and other objects within the databases.

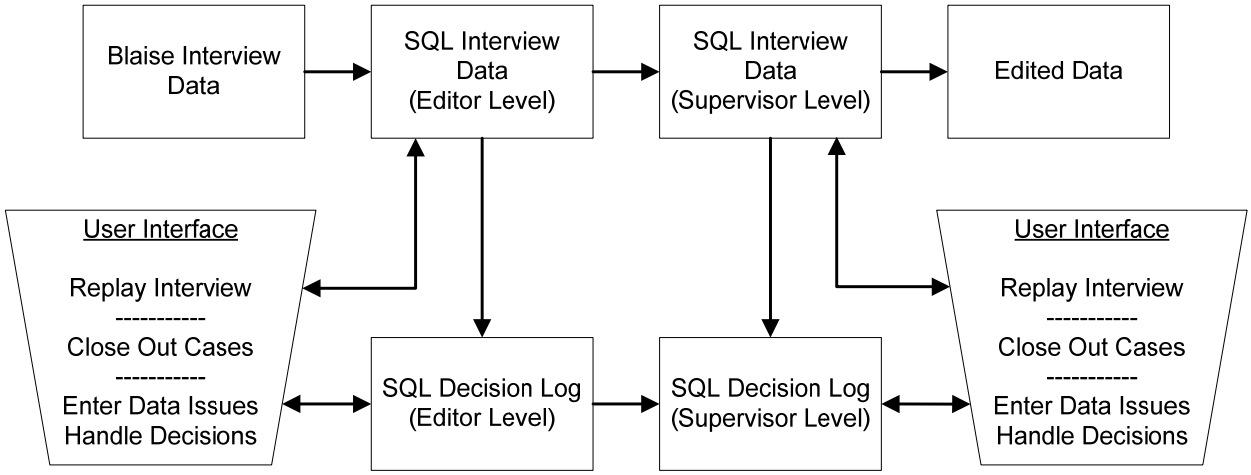
Since the BDPQA is a corporate system intended for use on multiple projects, it was designed to be highly customizable. This is achieved through a combination of INI files and metadata tables within SQL Server; as much of the system logic as possible is implemented through table-driven programming. Among the aspects of the system that can be customized for each project are the following:

- Verifier Role – Whether the user who edits a decision log entry can also verify the same entry
- Edit Check Exclusion – Which edit checks, if any, should not produce decision log entries
- Critical Data Items – Which fields, if any, should be included in or excluded from the Critical Item report
- NonDeliverable Items – Which fields, if any, do not need to be included in deliverables
- Watch Window – Whether the Blaise watch window should appear on the screen during replay
- Field Audit Trails – Whether field audit trails will be provided to system users
- Name to be used for instrument (form) in UI screens and report headings
- Name to be used for study in UI screens and report headings
- Internal security – List of the login IDs and roles of all individual users

4.0. The Data Flow within BDPQA

As indicated above, the Workflow Manager module of the BDPQA ran daily to pick up the new production units transmitted from the field and made them accessible to the editor (site office) level of the

BDPQA. Once the site had completed all work on the production unit, the Workflow Manager moved the closed production unit into the database accessible to the Supervisor (or central office) view. This movement is illustrated in the following diagram:



The diagram illustrates the centrality of the Decision Logs to the movement of the data through the system. Because the BDPQA centers on the decisions, a case cannot be closed out of the system – either at the site (editor) or central office (supervisor) level - until all decision log entries have been reviewed and resolved. This review and resolution process normally involves the following four steps for each decision log entry:

1. Review and resolution by an editor level editor
2. Review and resolution by an editor level verifier
3. Review and resolution by a supervisor level editor
4. Review and resolution by a supervisor level verifier

The resolution at any step overrides the resolution made at any previous step; this gives the supervisor level verifier the final say. There are some coding-related and other circumstances in which a decision log entry is not generated until the production unit reaches the supervisor level. For these entries, the review and resolution process is limited to only the last two steps.

5.0. The Users' Perspective: Two Views of the Tool

The overall approach for the initial project's data processing was to give each site office primary responsibility for the accuracy and completeness of its collected data and the home office responsibility

for the consistency of the processing of the data across all sites. Therefore, in terms of post-processing division of labor, the site offices were to perform the following tasks:

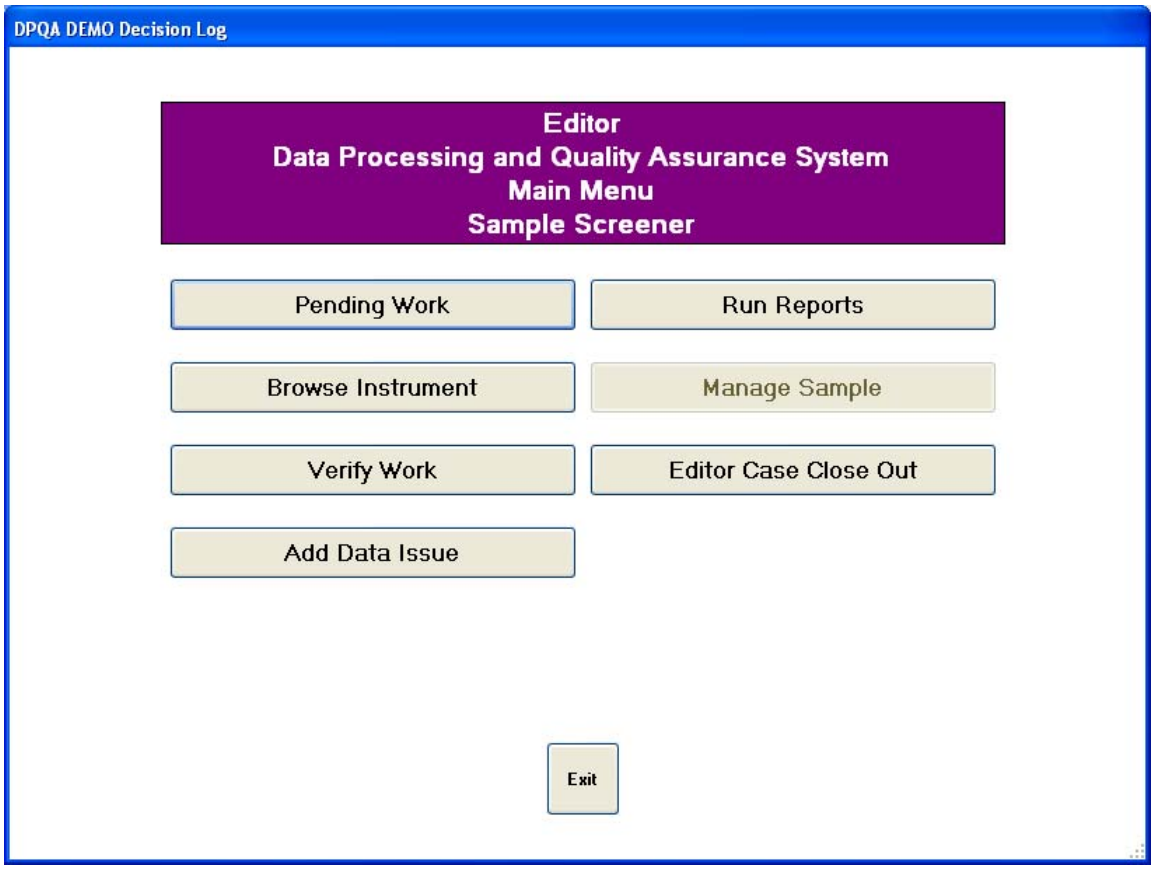
- Documenting in the system any issues reported to the office that might affect case data;
- Reviewing comments entered into the CAPI and CATI instruments;
- Updating the case data, if necessary, in accordance with issues and comments;
- Running the edits programmed into the Blaise data models to check that the transmitted and edited data was recorded as expected by Blaise;
- Reviewing case data as needed to support inquiries from the field or to validate interviewers' work; and
- Releasing the data to the home office for finalization.

For the initial study, Westat served as the central office. Westat's responsibilities for post-processing included:

- Establishing and documenting a consistent body of editing rules (based on situations encountered) to be implemented by the site offices;
- Reviewing all data updates and data decisions made by the site offices to ensure standardization of data processing across the study;
- Coding text responses as required by the study;
- Reviewing frequencies and cross tabulations to identify unusual data patterns; and
- Finalizing the data and corresponding documentation for delivery in preparation for analysis.

5.1 The Users' View: Editor Level

It was intended that the site offices would use the editor view of BDPQA. After the user signed in to BDPQA, and connection and loading were complete, the user selected the instrument to be worked. Regardless of the instrument selected, BDPQA users at each site office were limited to the data for their PSU and role only. We worked with the sites to ensure that only appropriate staff had only the access they needed. After the instrument was selected, the system displayed the BDPQA Editor Main Menu.

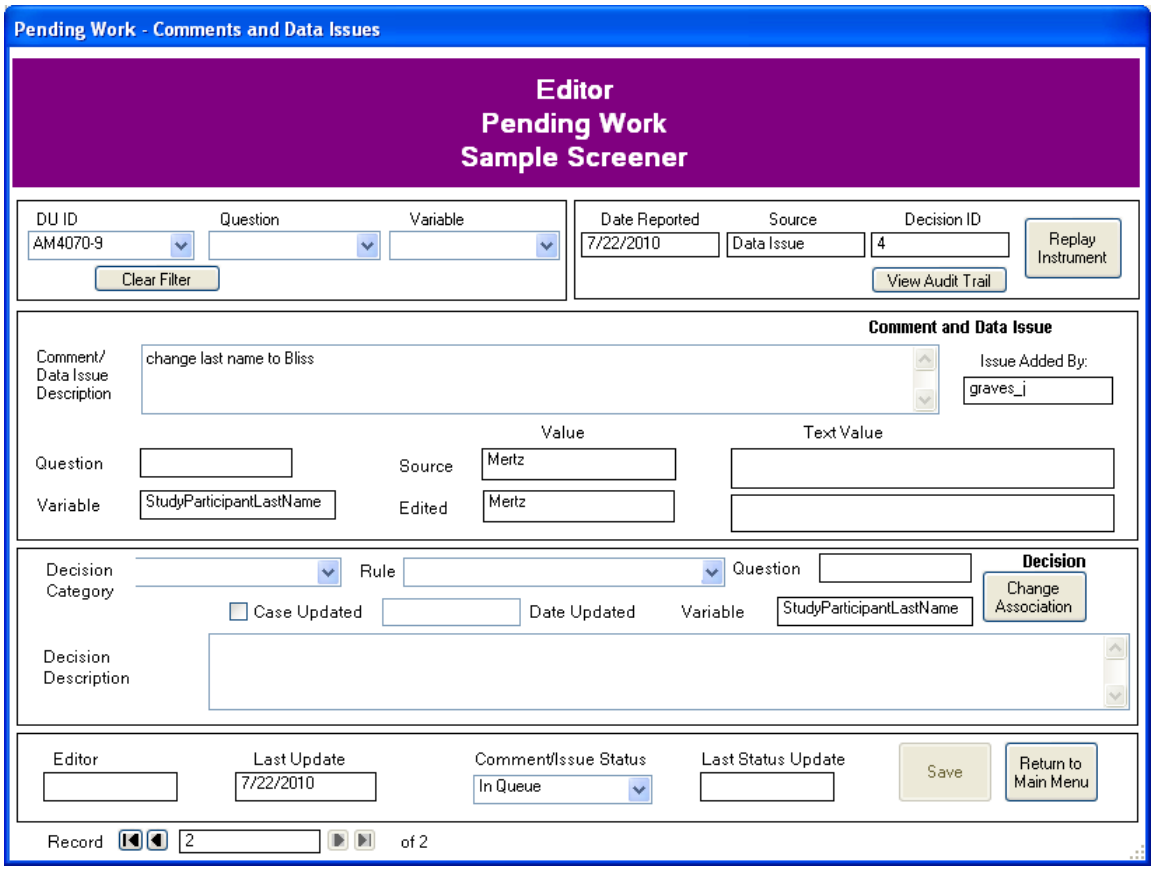


As you can see on the illustration, the BDPQA Editor Main Menu presented options in the order of expected workflow. (For this version of the tool, Manage Sample was not active, because this function was not required for the initial phases of the study.) Within the editor view, BDPQA site office users could select:

- **Pending Work** to review, resolve, and document in the Decision Log all field interviewer comments, post-data collection data issue reports, and edit checks;
- **Browse Instrument** to replay the instrument in read-only mode;
- **Verify Work** to review and verify data decisions at the 100% level, to run the reports that could assist in the verification of work, and to update the status of the data decisions to close them out individually;
- **Add Data Issue** to enter into the decision log any data issues as reported to field supervisors and data managers. The Add Data Issue screen functioned as an electronic template and all fields were required so that issue documentation would be consistent and interpretable;

- **Run Reports** to run the programmed monitoring reports to track and supervise workflow through data processing. The user had the option of viewing the report on the screen, saving the report in PDF format or printing the report; and
- **Editor Case Close Out** to release the production unit to the next level of processing – the supervisor level.

The core of the system is the Pending Work module, which allows users to review the details of a field comment, data issue, or edit check, and to document their resolution in the decision log.



The Pending Work screen has three main sections. The items displayed in the Heading section include:

- **Subject ID (here, a Dwelling Unit or DU ID)** with which the decision log entry is associated.
- **Question and Variable fields** which act as filters. Filters can identify all work related to a particular question or variable.
- The **Source** of the issue; this could be a Field Comment, an Added Data Issue or an Edit Check failure.

- The **Replay Instrument** button takes the user to the instrument in data collection mode to review and update data.

The next screen area is the Comment/Data Issue section and displays:

- **Question** and **Variable** with which the comment/issue is associated.
- **Source** - the original value for the variable.
- **Edited** - the current value. If the data has not been changed within BDPQA, the Source and Edited values will be the same.

The lower portion of the screen is the Decision section and includes:

- The **Change Association** button is used to change the Question and Variable with which the comment or issue is associated. Often in the field, an interviewer will need to enter a comment at a later point that does not directly relate to the relevant question. In order to include the most relevant associations in the data decision log for reporting and sorting purposes, this functionality was included in the processing options.
- The **Decision Category** is initially blank. It is used to indicate the level of action (or inaction) required to resolve the issue. The selections include Leave As Is, Action Required, or Supervisor Review. It should be noted that if Decision Category is set to Supervisor Review, the Comment/Issue Status of the issue will remain "In Queue" and cannot be changed. A verifier will have to look at the issue before it can be resolved and the status updated.
- **Rule** indicates the standard rule that was followed to resolve the issue and update the data, if required. It was intended that the set of rules would build throughout the course of the study as a variety of data issues were encountered.
- **Decision Description** is the detailed description of the action that was taken.
- The **Case Updated** box and the **Date Updated** field are related. If the box is checked by the user to indicate a data change, the system fills the date.
- The system populates **Editor** with the current user's ID when the user saves work and auto-fills the date in **Last Update.**
- **Comment/Issue Status** describes the point to which the issue has been worked. Status begins as "In Queue." Once the user completes work on the comment/issue/edit, he or she changes Status to "Ready for Verification" and the issue moves to the Verify Work queue. The system auto-populates **Last Status Update** whenever the Status is set.
- **Save** is activated, not grayed out, when the user has updated any of the information in the Decision section of the screen.

The **Verify Work** screen functions in the same way as the Pending Work screen; however, the screen includes a Verifier field (which is automatically filled in when the Verifier saves his/her work – this field functions like the Editor field in Pending Work) and the Comment/Issue Status selection is "Ready for Verification" and "Complete." When all comments/issues/edits (decision log entries for the production unit are "Complete," the unit is available in Case Close Out. Units are available for close out only when all associated decisions have a status of Complete. If a unit in BDPQA had no field comments or data issues and has passed all editing and QA/QC reporting, it is immediately available for close out. Once a unit is closed out at the editor level, it was no longer available for update within the individual site's version of BDPQA.

5.2 The Users' View: Supervisor Level

Once a case is Closed Out at the editor or site office level, it is available to the central office or supervisor level. The options at the supervisor level are similar to those at the editor level but are arranged a bit differently based on typical workflow at the central office. Here, Run Reports appeared first, followed by **Add Data Issue**, **Pending Work**, **Edit Case**--which allows changes to the instrument data; in effect, what "Replay Instrument" was to the editor, **Verify Work** and central office **Case Close Out**. Running reports was particularly important to the central office because, in addition to supporting the same functions as the reports at the editor level, reports were also necessary for the coding function to be performed by the central office only. Due to the need for coding of text responses, the central office view of the BDPQA relied on an updated data model with additional fields for coded values.

6.0. Lessons Learned

It is advisable when designing a Blaise instrument that will be used for by BDPQA to avoid using within the data model:

- Time stamps that depend on today's date and/or time
- Sampling criteria that can be changed by data value changes
- Auxfields that can give the impression of a non-existent error during replay (this can happen with auxfields, as opposed to fields, because their values are not stored in the database)
- Walls within the instrument that prevent interviewers from returning to previously-answered questions
- Field names that conflict with reserved words in SQL Server

For existing instruments with any of the above problematic features, workarounds can usually be devised.

Within the context of the decision log and various reports, Blaise fields are identified by a variable name (the actual Blaise field name) and a question name (the tag value associated with the field in the Blaise data model). For this reason, tags should be carefully chosen. You may want a unique clearly-named tag for each field and possibly shared tags for related fields.

In addition to paying attention to tags when programming the original Blaise data model, you are also advised to provide unambiguous wording with each potentially complicated edit check. When no explicit wording is programmed with an edit check, Blaise uses by default the relevant programming logic. This programming logic can be confusing and will appear by default in a decision log entry if no explicit wording has been programmed.

For field data collected via hardcopy forms and stored in a SQL table, an appropriate Blaise data model must be produced. This can be done using the Blaise OLE DB Toolbox to produce a data model from the SQL table and subsequently adding question text and tags.

7.0. Conclusion

The BDPQA met the challenges of this longitudinal, multi-mode, and distributed data collection effort. By leveraging the Blaise metadata available through the instrument authoring process and customizing our existing in-house tool to meet the requirements enumerated at the start of this paper, we were able to ensure that secure and high quality data processing ran right on the heels of data collection. The tool can be adapted for use on other projects in which raw field data are stored in either Blaise or SQL Server databases. The system edits and cleans the data using a two-level review process.

BDPQA offered a number of advantages in addition to those that accrue from using the superb editing capability offered by Blaise. These advantages include:

- Automatic versioning that can be used to ascertain for each production unit the date of any data changes and the values stored in the database at any given date.
- Ability to store and retrieve field audit trails for each production unit.
- Ability to generate a wide variety of reports
- Ability to provide internal security that identifies users and limits their roles
- Ability to use different data models at the editor and supervisor levels. This can support an approach whereby coding or other tasks can occur only at one of the two levels.
- Ability to provide decision log information either electronically or in hardcopy format to meet standard client requirements.

8.0. References

Allan, Boris, O'Reagan, Kathleen, and Lohr, Bryan. "Dynamic ACASI in the Field: Managing All the Pieces." *Proceedings of the 7th International Blaise Users Conference.* September 2001.

Dulaney, R. and Allan, B. "A Blaise Editing System at Westat." *Proceedings of the 7th International Blaise Users Conference.* September 2001.

Frey, R. and Rhoads, M. "Blaise Editing Rules + Relational Data Storage = Best of Both Worlds?" *Proceedings of the 12th International Blaise Users Conference.* June 2009.

Gowen, L. and Clark, P. "Lifecycle Processes to Insure Quality of Blaise Interview Data." *Proceedings of the 11th International Blaise Users Conference.* September 2007.