Meeting the Challenges of Managing CAI in a Distributed Organization

Asa Manning, National Agricultural Statistics Service, USA

Standardizing procedures are an important part of any statistical organization's attempt to reduce non-sampling errors. The physical structure of the National Agricultural Statistics Service (NASS) with its 45 field offices distributed across the county makes standardization even more important. For those supporting the CASIC activities in that environment, enforcing standards becomes an absolute necessity. Without the standards, support would be impossible.

NASS Structure

It is important to understand the basic structure under which NASS operates. NASS has approximately 400 employees in its headquarters located in Washington, DC and in nearby Fairfax, VA. The remaining 750 NASS employees are located in the 45 field offices, which NASS calls state statistical offices (SSO's). National survey activities are distributed between HQ and the SSO's as follows:

<u>HQ</u>

Instruction manuals & training
Sampling
Instrument development (Paper and Blaise)
Coordination
Compilation of national estimates

<u>SSO's</u>

Interviewer training
Data collection & editing
Survey administration
Analysis
State estimates

CASIC Implementation at NASS

Blaise is used in 43 offices for both data collection (CATI) and interactive editing. The following table summarizes the Blaise 4 applications that NASS currently uses:

Blaise 4 Applications In Use At NASS

Survey Application	Frequency	No. of States	Type	Peak CATI Volume	Peak Editing Volume	Post-Blaise Processing
March Crops/Stocks	March	42	Multiple Frame	25000	50000	mainframe
June Crops/Stocks	June	42	Multiple Frame	25000	50000	mainframe
September Crops/Stocks	September	42	Multiple Frame	25000	50000	mainframe
December Crops/Stocks	December	42	Multiple Frame	25000	50000	mainframe
Cattle Report	January & July	43	Multiple Frame	20000	40000	mainframe
Hog Report	Monthly	19 to 43	Multiple Frame	5000	16000	mainframe
Sheep Report	Jan. & July	42	Multiple Frame	7500	15000	mainframe
Agricultural Labor	Jan., Apr., July, & Oct.	42	Multiple Frame	6000	12500	mainframe
ARMS/Chem. Use Screening	May - July	42	Multiple Frame	45000	50000	mainframe
Cotton Gins	Aug. - Apr.	13	List Frame	1000	1000	LAN
Cattle on Feed	Monthly	13	List Frame	1000	2000	LAN
Catfish	Jan. & July	15	List Frame	1000	1400	LAN
Trout	January	17	List Frame	250	450	LAN
Chicken & Egg	Monthly	30	List Frame	500	1000	LAN
Bee & Honey	December	42	List Frame	2000	5000	LAN
Ag. Yield	May-Nov	42	List Frame	35000	37000	mainframe
June Area	June	42	Area Frame	N/A	200,000	mainframe
Pest Database	Continuous	42	Special	N/A	N/A	N/A
Reimbursable	Varies	Varies	List Frame	Varies	Varies	Varies
Census	Every 5th year	42	Census	200000	N/A	Unix/main

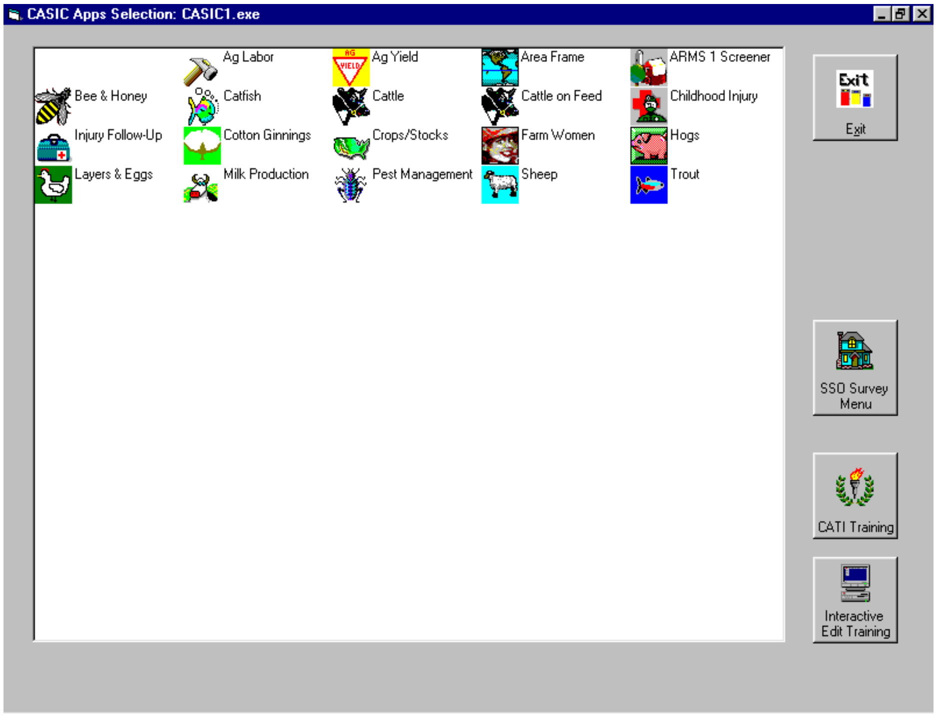
In NASS, our Blaise developers and end users are completely separated. All of our national applications are developed in HQ, but no data is collected or edited there. That activity is located in the SSO's. Not being able to communicate face to face with our end users also presents some challenges. Depending on telephone contacts to resolve problems is very inefficient and frustrating. Standards help reduce the need for support and also serve to expedite it when it does occur.

All of the above applications are developed and supported by the CASIC and Editing Section in HQ. Currently there are only five people that handle this activity. During the busiest time of year each SSO will have from 5-10 Blaise applications running concurrently. Without a stable hardware and software environment, the support demand could easily overwhelm the HQ support staff. NASS LAN policy assures that we will find almost identical LANs in every SSO. Stability in the Blaise software is obviously a must. We will often use a new version of Blaise in HQ for several months before deploying it to the SSO's. This provides a chance to shake down the new version before it is used in production.

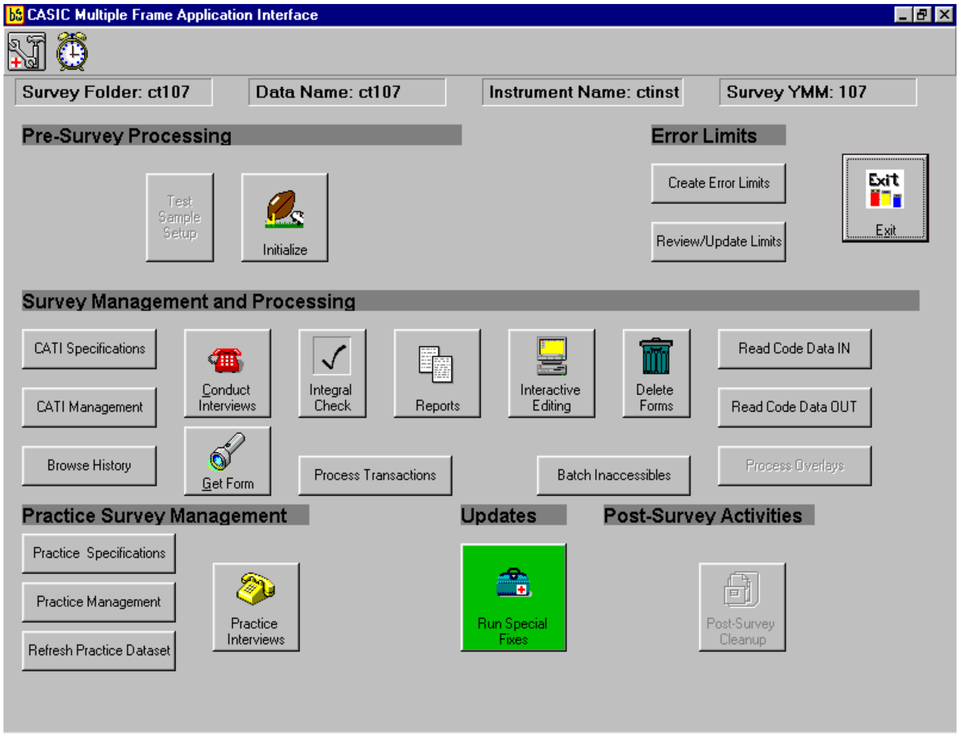
Since staff in the SSO's may be working with multiple Blaise applications simultaneously, we take care to make certain that where possible those applications look and feel the same. Strictly enforced CASIC standards assure that there is consistency from application to application and state to state. Several techniques that we use will now be discussed.

Standard Interfaces

When NASS moved from Blaise III to Blaise 4, we developed a series of standard interfaces, which were developed in Visual Basic, for our users in the SSO's. When the staff in the SSO's want to start a Blaise application, they click on an icon on their desktops. The following screen appears.



This screen has an icon for every Blaise 4 application that is available. From this screen the user clicks on the desired application. The standard CASIC interface for managing any Blaise application then appears.



Survey Folder: ct107

Data Name: ct107

Instrument Name: ctinst

Survey YMM: 107

This interface is identical for every Blaise application. It is organized into separate areas which correspond to phases of the survey process, such as pre-survey processing, survey management and processing, updates, and post-survey activities. These areas of the screen never change from application to application. Once a button is programmed into a position on the interface, no other button will ever use that position. This explains the empty area to the left of the “Test Sample Setup” button on the interface. The Crops/Stocks application instruments are prepared in the SSO, so a “Prepare Instrument” button appears on the interface for that application. Since HQ prepares the instrument for Cattle that position remains blank on the interface in our example. Buttons are present but inactive if certain key input files are not available. The “Test Sample Setup” button is an example. The general flow of the processes is from left to right. The Survey Management and Processing moves from CATI management on the far left, then CATI, then Interactive Editing. The bright green color of the “Run Special Fixes” button is designed to attract attention to the need to run an update. This indicates that the CASIC & Editing Section has delivered a fix for this application and it has not yet been applied. This button will be the same color as the other buttons and disabled once the fix has been applied. All of these functions make it easier for the end user in the SSO’s to manage their Blaise application without HQ assistance.

Standard Blaise Shells

When the CASIC & Editing Section develops a Blaise application, we only have to program the questions specific to the application. NASS already has a series of shells that handle all of the administrative type questions. The developer essentially “wraps” the appropriate shell around the code they developed by including the pre-existing shell code in the preparation process. Once prepared, the instrument automatically meshes with existing Manipula processes and with the CASIC interfaces just discussed. This assures that administrative questions on similar Blaise surveys always function the same.
One key ingredient in the success of this approach is that NASS has already committed to the use of

standards in all survey instruments, paper and automated. NASS uses only four types of administrative designs on its questionnaires. They are multiple frame, list only, census, and area. Blaise shells are maintained for each one. The initial effort to build these shells was significant. Maintaining them requires some time, but because the original effort was designed to be flexible, is actually minimal. The amount of development time saved in each application has paid back the shell development time many times over.

Application Distribution

NASS uses a standard method called LAN Updates to deliver any software related update to the SSO's. SSO's will receive about 500 updates each year with CASIC related updates accounting for about half. All LAN Updates are coordinated by staff within the Information and Technology Division (ITD). The staff in the CASIC and Editing Section prepares the LAN Update. It will consist of a series of zip files which correspond to destination directories on the SSO LAN, an install BAT file which will unzip the files onto the SSO server, and documentation. The Blaise developer in HQ will login to a HQ server which is configured like the SSO servers. The install process is then tested, followed by application testing once successfully installed. Once all testing is completed, the LAN Update is forwarded to ITD. They assign the LAN Update a unique number and place it on a server in HQ. The SSO is then notified by email that the LAN Update is available. The SSO then must use FTP to move the LAN Update onto their server. They then follow the instructions to install the update. By using the rigid policy to manage software updates going to the SSO's, NASS protects the SSO's from poorly planned updates. It takes extra time in HQ to prepare the LAN Update, but if efficiency is gained in 43 receiving offices, the time in HQ is worthwhile.

The CASIC and Editing Section can build certain safeguards into the installation batch file. We may look for certain files to assure that the application being updated is present or current. We may also look for the presence of users that may already be in the Blaise application. The presence of these safeguards are designed to avoid problems created by prior updates being missed or files being open when we try and replace them. Even though the SSO's are given specific instructions that should assure these things don't happen, you can't count on all instructions being followed exactly in 43 offices every time.

NASS strives very hard to avoid changing the data model after data collection has begun. It is very problematic to coordinate a data model change in our distributed environment. Thus, unless the problem being fixed is significant, we will usually find an alternative way to deal with it.

Using ReachOut and the WAN to Aid Support

Solving a problem that surfaces in an SSO can be difficult with the best of tools. Listening to someone describe a situation over the phone without being able to see their workstation wastes tremendous amounts of time. One of the biggest aids for support at NASS in the last few years is the availability of our Wide Area Network and a software package called ReachOut. Staff in the CASIC and Editing Section can now "ping" a workstation in an SSO and watch as applications run. The savings in support time are not easy to measure, but we would estimate a savings of 75 percent.

Security is controlled by the SSO. We can only reach the other workstation if the office has set up ReachOut on it and gives us the necessary ReachOut login and password. The SSO can also give control of the workstation session to the HQ user when necessary to expedite finding a solution.

CASIC Coordinator Training

Probably the most significant factor that determines the amount of support demand from an SSO is the level of expertise available in that office. Most support calls do not really require a developer, but rather an experienced Blaise user. When we are able to elevate the Blaise knowledge in an office, we see a reduction in support calls from that office. Since NASS adopted Blaise as our primary CASIC software in the early 1990's, we have twice held a series of CASIC Coordinator workshops. These workshops were designed to teach SSO staff how to manage the CASIC process, not develop applications. The 3.5-4.5 day agenda consisted of sessions on CATI Management, using data collection and editing instruments, using Blaise utilities to monitor and solve problems, and generally how to manage the flow of data through the Blaise process.

We held these workshops prior to introducing Blaise III and Blaise 4. We trained two staff from each SSO during the first series of workshops and one in the second. Since the change from Blaise III to Blaise 4 was less dramatic than the original shift from CASES, we felt that one well-trained person in each office would be sufficient. Once the person returned to the SSO from the workshop, it was their responsibility to help manage the first Blaise 4 applications that arrived in the office. They were also charged with passing their expertise to others in the office. Almost everyone in the office would touch Blaise, so it was important for this knowledge to be shared. The success of this training transfer depends on the individual's training skills and the commitment of management in the office to allot time for that training. Since we can't train everyone in each SSO, training an "expert" and depending on the transfer of that expertise to others is at the core of all NASS training.

Blaise Developer's Training

Each SSO has certain survey applications which are not covered by the HQ survey umbrella. For these applications, the SSO must develop their own survey processes. Since Blaise is the standard tool for CATI and IE, most SSO's need the ability to develop their own Blaise applications. To support this need, the CASIC and Editing Section developed a training course for potential SSO developers. This 3.5-day course consists of a series of slide shows about the Blaise language, followed by a series of hands-on exercises. During the exercises, each participant gradually assembles pieces of a crop acreage and production instrument. The course closes with an introduction to Manipula and guidelines on using a shell designed for state applications. If the shell is used, the developer also benefits from being able to easily add their application to the standard CASIC interfaces without knowledge of Visual Basic.

NASS has conducted six of these workshops over the last seven years. Thirty-two states have developed their own applications. These workshops and subsequent observations of the participants as they develop their own applications provide us with a glimpse of potential candidates for the CASIC and Editing Section. We don't advertise it, but the developer workshop is an audition of sorts.

One side benefit to an SSO having a developer in house, is a reduction in support needs in that office. That developer has an understanding of Blaise that exceeds almost every CASIC Coordinator. That understanding translates into a reduction in support calls from that office, because the local expertise has been elevated.

Application Testing

In an environment where support is challenging, obviously the easiest way to minimize the challenge is to eliminate the need for support. Perfect applications require almost no support. We have not achieved perfection, but do as much testing as possible before applications go to the SSO's. We have up to four phases of testing depending on the complexity and stability of the application; developer testing, special testing in HQ by SSO staff, testing of a preliminary instrument in the SSO's, and a review of the final application.

Simpler applications that have changed only slightly will be tested by the developer and then the SSO's will test the application once the "final" application is received. No rigorous testing in the SSO's takes place.

For the more complex applications, we add a phase where the SSO's test a preliminary instrument. This instrument is delivered to the SSO's about 3-4 weeks prior to the start of the survey. The preliminary instrument is distinguished from final instruments by a bright green background to minimize the likelihood that it will mistakenly be used for live data collection. The SSO's run cases completely through the preliminary application to test the entire process. They communicate any problems to the CASIC and Editing Section and those bugs are removed before the final instrument is delivered to the SSO's. The final application is due in the SSO's 3-5 business days before the start of the survey.

The Crops/Stocks application is even more challenging in that every SSO has a unique instrument. Because of this additional complexity, we will at times include a special phase of testing as the HQ developer begins to create the preliminary instruments. We will bring one of our strongest CASIC coordinators from an SSO into HQ to test each individual state's instrument. They will help update the Blaise specifications database that is used to assemble the instruments, generate the state instruments, and verify that each one is accurate. Once that phase is complete, the preliminary instrument is delivered to the SSO's and testing of the preliminary instruments continues as described in the previous paragraph.

Challenging Instrument Requirements

NASS maintains three applications that present special challenges. The Crops/Stocks CATI/IE and June Area IE applications have unique instruments for every state. The Census of Agriculture CATI instrument has 14 unique regions. All of these surveys collect information on crop acreage and production. Since crops vary by state or region, so do the instruments. NASS employs three different methods of developing these instruments. All of these have been the subject of entire papers at previous International Blaise User's Conferences, so only an overview will be provided here.

Crops/Stocks This application collects information on crops grown and grain stored in March, June, September, and December. The initial efforts on Crops/Stocks go back to Blaise 2.

First, we created a library of modules for every combination of questions ever asked for each crop. For example, early in the growing season only planted acres might be asked, while in December we might ask acres planted, harvested, and grain produced. The same questions are asked for multiple crops, so we built a structure for each combination of questions. Then we generated the actual crop specific code module by feeding the individual crops into the structure using Manipula. By generating the crop modules from a single clean structure module, we assured consistently clean crops modules. This step was called the code generator.

We maintain a Blaise specifications instrument with a record for each state and quarter. The Blaise instrument brings up a series of questions on each crop and stock item. The user indicates whether the commodity will be included, and if so what questions will be asked. Once the information has been input, we use Manipula to read the specifications and assemble the needed code modules. Then through a series of batch files, the instrument is concatenated and prepared. We call this process the instrument generator.

This process performs the customization at preparation time. The entire library is distributed to the SSO's and instrument is prepared there. This approach has allowed us to reduce the staff time needed to create the 42 unique instruments and at the same time produce an instrument that is more reliable.

June Area

This application is used to edit data that is collected via personal interviews with farmers each June. Essentially, the major purpose of the survey is to identify the crops grown on a tract of land. As with the Crops/Stocks application, the crops for each state are unique. The IE instrument was originally developed as part of a CAPI research project. The IE instrument was kept after the CAPI effort was discontinued. A different method is used for handling the unique crops in this instrument.

We build a toggles file for each state that lists the crops to be edited. This file is read at run time to customize the crop section for the state in question. This application is now one of the most popular Blaise instruments used in NASS because of the tremendous time savings gained by editing this data interactively rather than with the old batch edit.

Census

The Census of Agriculture is conducted every five years. The Blaise instrument is only used for CATI. No editing is done in Blaise. The instrument must be able to collect any crop, no matter how minor, grown anywhere in the United States. Our typical survey instruments only collect data for the crops that are included in our estimation program. Many minor crops are not included. On surveys, we ask the farmer about each crop. That method would not work on the census because the list of crops is so vast, so we used a completely different technique.

Before we ask specific acreage questions, we build a profile of crops grown on the operation. We ask the farmer to identify the crops grown and one by one capture them using a trigram coding module. When the farmer indicates that all crops have been covered, the interviewer reviews the list of crops with the farmer to make sure nothing has been overlooked. Special efforts are made to screen for certain crops such as hay, which experience has taught us farmers tend to forget. Once the list is complete, we ask the appropriate acreage, production, and irrigation questions for each crop identified in the profile.

This was the first time we used the trigram coding module at NASS. This technique was very successful and we plan on using a similar process during the 2002 Census of Agriculture.

Summary

As you can see, NASS uses a variety of techniques to attempt to minimize the support demands for our Blaise applications. We feel that these standardization techniques, which we have implemented over a period of over eight years, are the only reasons that we are able to support our many applications with a small number of developers. The NASS structure gives us very little choice. We hope other organizations can learn for our experience. Even if you don't need to enforce standards as rigidly as we do, some of these techniques might serve your organization as well.