Deploying an Automated Data Read In Process at the National Agricultural Statistics Service

Emily Caron
National Agricultural Statistics Service
IBUC XVI
Beijing, China
The Old Read In Model

• 46 state field offices
• Read in manipula program ran locally, on demand by users
• LAN based processing
  – Local Blaise datasets (.bdb) containing each office’s sample
• Very fast!
Early Centralized Read In Model

- 46 state field offices, transitioning to 12 regional field offices
- Read in manipula program ran locally, on demand by users, sometimes concurrently
- WAN based processing
  - Centralized MySQL database containing national sample
- Initially not as fast, but still tolerable. After more/larger surveys were added, NOT acceptable!
Current Read In Model

- 12 regional field offices
- Read in manipula program runs on the Blaise data server at set intervals
- Server based processing
  - Centralized MySQL database containing national sample
- Much better!
Queuing the Data

- User clicks on “Queue Files for Read In” button

- SurveyInfo.txt file is also created
  - Contains metadata used by read in program
Queue Options

• User has choice of two available queues
  – Overnight (default) 🌙
    • Data is read in overnight at 12:15 am
    • Runs Tuesday - Sunday
    • **Strongly** preferred for large quantities of data
    • DATA\CASIC\<survey name>\HQ\QueueNight
  – Daytime ☀️
    • Data is read in every 15 minutes from 6am-11pm
    • Runs seven days a week
    • Use when data being loaded will be edited today
    • DATA\CASIC\<survey name>\HQ\QueueDay
Trigger Files

• Queuing data creates Trigger files
  – File naming format: `<survey>src.TRG`, where src = RAW or EDR
  – Saved under DATA\CASIC\TriggerNight or DATA\CASIC\TriggerDay
  – Path names and parameters used by read in manipula are stored in these files
Auto Read In Program

• Created as VB.NET Console Application
• Set up as Scheduled Tasks on each Blaise Data Server
• Logs progress & any errors at server level
• Sweeps appropriate Trigger folder to see if any surveys need to read in data
  – If no, writes to log and exits
  – If yes…
For each survey with a trigger:

- Program looks for InUse.txt file
  - If found, indicates a job is already running for this particular survey. Program writes to log and this survey is ignored.

- Surveys with no InUse.txt file are processed
  - InUse.txt file is created
  - Trigger files move to TriggerStage folder
  - Thread is started for each survey
Auto Read In Threads

- Threads = faster processing
- No degradation for read in jobs run concurrently across surveys

- Thread actions:
  - Queued data files move to QueueStage folder
  - Data is merged together by type
  - Read In manipula program runs
  - Logs at survey & server levels
Item Level Rejects

• New fields added to Blaise instruments
  – OPEN field: invalid, blank, or duplicate item codes, or item codes with blank values
  – Enumerated field: “Reviewed”
• Critical error - If not Reviewed then OPEN field must be empty
Batch Assignments & Report

- Users prefer to edit by batch
  - CATI data batches = Julian date (1 – 366)
  - Auto read in batches = Julian date + 500

- Batch Count Report
Other Improvements

• New Interactive Edit sort options
  – Incorporated Capture Source
• Email alerts
• Data automatically re-queued after random COMException error
Conclusion

• Auto Read In program resulted in…
  – Faster read in jobs
  – Better overall system performance
  – Time savings for users and developers
  – Overall increased user satisfaction
Questions

Emily.Caron@nass.usda.gov

Roger.Schou@nass.usda.gov