Working Paper.

UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Workshop on Statistical Data Editing
(Neuchâtel, Switzerland, 18-20 September 2018)

The Evolution of Banff in the Context of Modernization

Prepared by Darren Gray, Statistics Canada

I. Introduction

1. Banff (Statistics Canada, 2017a) is a generalized edit and imputation system designed and maintained by the methodology and system engineering branches of Statistics Canada. In addition to being the primary tool used internally for business surveys (and some household surveys) at Statistics Canada, its users include a diverse collection of national and international agencies, including some National Statistical Institutions (NSIs).

2. Statistics Canada, along with many other agencies, is updating the way data is collected, analysed and shared with users. As part of this modernization initiative, existing methods must be reviewed for suitability, while new methods are tested and developed for use. These obligations present the following challenge: how do we maintain the "generalized" aspect of Banff (focusing on methods with broad application) while facilitating the integration of emerging methods in an efficient manner?

3. In Section II, we present an overview of Banff's current research and development plan, in the context of modernization. The remainder of the paper focuses on one of our key objectives – Banff as a modular, automated data editing tool – and how this fits within the Generic Statistical Data Editing Model (GSDEM) (UNECE 2015).

II. Banff research and development objectives

4. Generally speaking, the current Banff research and development plan is based on two broad goals: to ensure that current user needs are met, and to build the tools and framework required to treat anticipated future needs.

5. A discussion of Banff user needs (based on an extensive user consultation process) and future Banff development was recently presented by Thomas (2017); the primary objective of this process was to provide direction for an upcoming proposal to revisit Banff: *"The main goals of this project are not yet defined. The scope could be very small and include minor enhancements to the existing system or, if warranted, it could be a complete overhaul of the system and its functionality."*

6. In anticipation of future needs, and to position Banff as a *modern* data editing tool, the Banff methodology team is focused on facilitating the integration of external methods (for example, publicly available data editing R packages) into Banff's framework. Combined with the development of generalized data editing assessment methods, the goal is to provide users with the necessary tools to quickly and easily test and evaluate new data editing techniques.

7. In many cases, these two goals overlap. We now present three current research projects, and discuss their relation to these goals.

A. Broadening the scope of Banff's existing methodology

8. This research is focused on reviewing, updating, and when necessary, broadening the scope of the existing Banff data editing procedures. The primary goal is to examine whether the Banff procedures have application beyond their current design, in particular with respect to the *type of data* and *edit constraints* currently accepted by Banff. While Banff was designed to treat continuous numerical business data, the reality is that most surveys (business and otherwise) contain discrete and categorical data as well. Expanding the scope of Banff's procedures to treat non-continuous numerical data would benefit a number of users. Additionally, many of Banff's procedures are designed to handle constraints defined by linear inequalities. This is not always sufficient for users; for a recent discussion on the topic, from the perspective of an external user, we refer the reader to Xu, Kim and Terrie (2017).

9. The Banff methodology team is currently reviewing the various data editing functions provided by Banff to examine the feasibility of relaxing these constraints. Early analysis indicates that expanding such functionality into Banff's error localization method (based on the Fellegi-Holt paradigm) will be the most complex. In fact, the two problems – treating categorical data and incorporating flexible edits into error localization – requires a significant change to Banff's current error localization methodology. Our research is currently looking into expanding on methods proposed by De Waal (2003); a discussion on this topic was recently presented by Gray (2018).

B. Improving Banff as a modular, automated data editing tool

10. One of the primary advantages of Banff is in the realm of automated data editing; large-scale statistical programs such as Statistics Canada's Integrated Business Statistics Program (IBSP) rely on automated data-editing procedures to efficiently and effectively perform data editing and imputation on a large scale within a production timeline. Winkler (2006) provides an overview of the history of automated data editing dating back to Fellegi and Holt (1976), demonstrating the efficiency of automated data editing software (including Banff). The subject is more recently discussed in depth by De Waal, Pannekoek and Scholtus (2011) and Pannekoek, Scholtus and Van der Loo (2013). A number of NSIs have also reported on the specific gains of Banff's automated data editing features; for examples, see Barboza (2011), Johanson (2013) and Terrie (2018).

11. For most modern surveys, the data editing process consists of multiple data editing and imputation steps run in sequence. This requires that the outputs (both statistical data and metadata) from one step in the sequence are properly interpreted and prepared as inputs to subsequent steps. While Banff provides some functionality in this domain, users have remarked that improvements could be made (Thomas, 2017).

12. The objective of this research project is to modify and promote Banff as a modular, automated data editing tool, allowing users to run data editing processes (i.e., modules) in any sequence with minimal set-up and data management between steps. While this addresses some user needs, it is primarily being developed to facilitate the integration of external functions within the Banff framework.

13. The majority of this paper is focused on this research project: to improve Banff's utility as a modular, automated data editing tool. Section III discusses the standardization of automated Banff procedures, and how such an exercise facilitates the integration of external (i.e., non-Banff) data editing functions. Section IV examines planned improvements to the Banff processor, giving users more control and flexibility over a data editing process flow. In Section V we discuss the possibility of expanding Banff's metadata framework to include record-level statuses.

C. Developing a generalized data editing assessment and evaluation tools

14. As Statistics Canada looks to modernize operations, there is an increased appetite for implementing new, cutting-edge methodology. The objective of this project is to expand on the research of Stelmack (2018) to develop a standardized assessment and comparison tool, allowing users to evaluate not just new imputation methods, but the effect of imputation strategies and specific parameters on an imputed file.

15. While the short-term objective is to create an assessment tool, further research will focus on estimating variance due to non-response and imputation, based on established methods (Rancourt 2007). The generalized estimation system G-Est (Statistics Canada, 2017b) currently provides a method for assessing the variance due to non-response and imputation, based on Banff metadata outputs; however, this method is only compatible with select imputation methods and the goal is to provide a generic tool that can assess variance based on an *arbitrary* automated imputation method. This research only considers the effect of imputation to treat missing values; further thought must be given to the impact of data editing designed to detect and treat erroneous values (Scholtus et al., 2017).

III. Standardizing the automated Banff procedures

16. Banff consists of nine data-editing functions, called as SAS procedures, described briefly below; a complete description of each procedure can be found in the Banff Functional Description (Statistics Canada, 2017a) and User Guide (Statistics Canada, 2017c).

- **VerifyEdits**: Evaluates a set of edits for consistency and redundancy; can optionally generate extremal points and implied edits.
- **EditStats**: Determines the number of records within a dataset that pass, miss, or fail each edit; provides summary statistics only.
- **Outlier**: Outlier detection using the Hidiroglou-Berthelot or Sigma-Gap methods.
- **ErrorLoc**: For records that fail to pass a set of edits, identifies the minimum number of variables that must be amended to pass all edits, following the Fellegi-Holt error localization paradigm.
- **Deterministic**: Deterministic imputation for records where only one value (or vector of values) will pass all edits.
- **DonorImputation**: Nearest-neighbour donor imputation ensuring that all amended records pass specified edits.
- **MassImputation**: Donor imputation without edit constraints; generally used for blocks of missing data.
- **Estimator**: Estimator (or model-based) imputation; users can choose from a number of pre-defined algorithms or create their own; may reference auxiliary and historical data.
- **Prorate**: Prorates record values to meet specified edit constraints.

17. This assessment is motivated by two objectives: to facilitate the integration of external data editing processes into the Banff framework, and to simplify the process of running multiple Banff procedures in sequence as part of a data editing process flow. In light of these objectives, we concern ourselves only with those Banff procedures that can be considered *automated data editing functions* within the GSDEM. This excludes VerifyEdits and EditStats. The VerifyEdits procedure is not dependent on the input statistical data, and as such does not fit the GSDEM criteria of data editing *function*. While EditStats does fit within the GSDEM, it would be considered a review function, and does not fit into an automated sequence of data editing processes.

A. Review of procedures, data structure and status flags

18. The remaining seven procedures can be classified into one (or in some cases, both) of the following function types, as defined in the GSDEM (page 8):

- *Selection.* Functions that select units or fields within units that may need to be adjusted or imputed or, more generally, identify selected units or variables for specified further treatment.
- *Amendment.* Functions that actually change selected data values in a way that is considered appropriate to improve the data quality. This includes changing a missing value to an actual value, i.e. imputation.

The classification of each Banff procedure (omitting VerifyEdits and EditStats) is given in Table 1.

Table 1 Banff procedure classification

Banff Procedure	Selection function	Amendment function
Outlier	X	
ErrorLoc	X	
Deterministic		X
DonorImputation		X
MassImputation	X	X
Estimator		X
Prorate	X	X

19. The Banff procedures act on statistical data. For clarity, we will use the following terminology when referring to Banff input statistical data:

- The data consists of a set of ***records*** (also referred to as units, or observations) each with a unique identifier (***ID***), and associated data ***values***.
- The values are categorized by distinct ***fields*** (or variables), with each distinct field having a unique identifier (***FieldID***).

This terminology is chosen to correspond with the GSDEM, although the unique identifiers ***ID*** and ***FieldID*** are specific to Banff. The Banff documentation always refers to the act of changing a variable as *imputation*; as this term is more often reserved for the act of filling in a missing variable, we will use the more general term *amendment*, as in the GSDEM, within this document.

20. The Banff selection and amendment procedures always act on individual values in the input dataset; they do not alter the overall structure of the data. Individual values in the dataset can always be identified by the unique key <ID, FieldID>. Most procedures track the action they take on the data by outputting metadata in the form of a ***field status*** (also referred to as ***status flags***) associated with affected values. (Note that when we use the term field status, we are referring to a status that is assigned to a specific value associated with an individual record; never to the field as a whole.)

21. Each procedure (with the exception of MassImputation) automatically produces an output metadata table containing these flags, along with the associated identifier <ID, FieldID>. While some of these flags are for diagnostic purposes only, the following are used to pass information from one procedure to another:

- **FTI (field to impute)**: Assigned by selection procedures to indicate which fields require amendment.
- **FTE (field to exclude)**: Assigned by Outlier to identify outlier values that do not meet the threshold for amendment, but should be excluded from other procedures in the data editing process.
- **Ixxx (imputation flag)**: Assigned by the amendment procedures to indicate which fields have been successfully imputed (or amended). The first letter in the code denotes “Imputation” while the

remaining letters indicate the type of imputation that has taken place: for example, deterministic imputation is denoted IDE while donor imputation is denoted IDN.

22. These flags form the basis for Banff's modular, automated framework: output metadata (the *outstatus* file) from one procedure can be subsequently used as input metadata (the *instatus* file) for a subsequent one. Some procedures cannot function without this input metadata: the amendment-only procedures (Deterministic, DonorImputation and Estimator) are not capable of identifying records for amendment as currently programmed – they require an instatus file with FTI flags generated from a previously run selection procedure. DonorImputation and Estimator automatically detect FTE flags on the input status file and exclude them from their procedures. (For DonorImputation, this means excluding the identified fields from being donated, and for Estimator, this means excluding them from model parameter calculations.) Finally, imputation flags (Ixxx) can optionally be read by some procedures (DonorImputation, Estimator and Prorate) that need to distinguish between original and amended data.

B. Identified issue: data management "holes"

23. A common user complaint about the Banff procedures is in regards to the format of their outputs; in particular, first-time Banff users are often surprised that amendment procedures do not output an amended version of the input statistical data. (They only output the subset of data that was amended.) This requires an additional data management step, on the part of users, to merge the amended data with the original dataset. The same issue applies to the status files; none of the procedures automate the process of updating an instatus file with the results of a subsequent outstatus file.

24. In a multi-process flow, this requires data-management steps on the part of users that could easily be automated. This "*requirements by the user to process and analyse function output and prepare this for input into the next step of the editing process*" was labelled as "holes" in the Banff framework by users from Statistics Finland (Thomas, pg. 5).

C. Standardizing Banff procedures

25. Based on our review of existing Banff procedures, and user feedback discussed in this section, we propose the following criteria defining a ***modular, automated data-editing procedure*** in the Banff framework:

- Procedures must be classifiable as one of the following, as defined within the GSDEM:
    - Selection procedures
    - Amendment procedures
    - All-in-one (selection and amendment) procedures
- Mandatory inputs:
    - Statistical data file
    - Status file
- Mandatory outputs:
    - An amended statistical data file (for amendment procedures)
    - An updated status file
- Procedures must use the following status flags:
    - Selection procedures must output FTI or FTE flags (or both)
    - Amendment procedures must input FTI flags and output Ixxx flags
    - All-in-one procedures must output Ixxx flags

26. Most Banff procedures already meet the majority of these criteria; the only significant change is to modify each procedure such that they automatically update the input statistical data and status files for the

next procedure. (In fact, the code for this feature is already developed and incorporated into the Banff Processor, discussed in Section IV.) Modifying them to meet this criteria should address the data management "holes" brought up by users, allowing them to run the Banff procedures in an automated sequence without any intermediate code updating the data or status files.

27. This criteria also provides a simple, direct framework to evaluate the suitability of external data functions as automated, modular procedures in the Banff framework. In particular any function that is data-driven, performs selection or amendment (or both) and whose effect on the dataset can be tracked at the field level, fits into the framework; all that is required is to "wrap" the function within a SAS environment such that the inputs and outputs meet the criteria above. The Banff methodology team is currently developing and testing prototypes to demonstrate this functionality.

28. Finally, the criteria above should only be taken as a set of *minimum* requirements. They have been chosen to ensure that any function meeting the above criteria can function within the Banff framework and integrate within a modular sequence. However, there is no restriction preventing procedures from using additional inputs (e.g., historical data) or producing additional outputs (e.g., donor maps). Procedures may also output additional status codes containing relevant information, though there is no guarantee that non-standard codes (i.e., codes other than FTI, FTE, and Ixxx) will be detected or correctly interpreted by other procedures without additional programming.

IV. The Banff processor

29. The Banff processor (BP) (Statistics Canada, 2017d) is a metadata-driven tool designed to facilitate the use of Banff procedures in a sequential data editing process flow. The BP offers the following advantages over calling Banff procedures directly from SAS:

- It automatically handles the intermediate data management steps between individual procedures (thereby avoiding some of the user concerns addressed in the previous section).
- All parameters related to the individual procedures and overall process flow are stored in linked metadata tables.

These two features allow users to focus on the details of the data editing process flow, rather than the programming. The overall data editing process flow, along with each process's parameters, can be easily inspected and changed from within the metadata tables. This is especially convenient for large processes (consisting of many procedures in sequence), and for testing different data editing strategies.

30. In this section we review the BP and describe a user-identified issue. The proposed solution fits within our objective to provide users with a highly flexible and intuitive tool; it also ensures that external procedures that fit the standardized Banff model described in Section III can be seamlessly integrated into the BP.

A. Description

31. Data editing process flows within the BP are referred to as ***jobs***. Each BP job is described by the user in terms of sequenced processes in a main ***driver table*** whose relevant columns (for the purpose of this paper) are described below:

- **JobID** (job identifier): An identifier to declare which processes are to be run together in sequence. This allows users to specify multiple jobs in the metadata tables, although only one job can be called at a time.
- **SeqNo** (sequence number): Numerical values indicating the order in which each process is sequenced. Within a given job, the sequence numbers must be unique.

- **Process**: Identify which Banff procedure is to be called. Users may also call custom data editing programs by specifying a SAS program in this column.
- **SpecID** (specification identifier): Used to point the BP towards the process-specific specifications found in additional metadata tables.

32. The BP is called from a SAS macro, which must reference all input datasets and the location of the BP metadata tables. Once called, the BP runs each process in sequence. After each process, the BP generates a current data file (consisting of original and amended data) and a current status file (consisting of the most recent status assigned to a field). These files provide a "snapshot" of the data editing process between each process in the job, and are used as inputs for subsequent procedures. At the end of the job (i.e., after the last process) the final amended statistical file is output, along with the final status file.

B. Identified issue: procedure linkage

33. Consider a process flow consisting of multiple selection and amendment procedures in sequence. There may be reasons to link specific amendment procedures with specific selection procedures; for example, one amendment procedure may be best suited to correct only a certain type of error, and not others. Restricting an amendment procedure to treat only one type of error, within a process flow, is an example of what the Banff methodology team refers to as a ***procedure linkage*** need.

34. The BP already forces one type of procedure linkage, between ErrorLoc, which identifies records whose values fail a set of edits, and DonorImputation, which amends values such that the amended record satisfies a set of edits. When running a DonorImputation process, the BP will only amend records that were identified as requiring amendment using ErrorLoc *with the same specified set of edits*. While this example of procedure linking may be appropriate in many circumstances, a number of Banff users have expressed a desire to bypass it.

35. On the other hand, there are cases where users would like to *impose* a procedure linkage within a process flow, but cannot do so within the BP. Most consist of scenarios where users wish to link an amendment procedure to a subset of selection procedures.

36. We note that this degree of control (procedure linkage) can be managed outside the BP (calling the Banff procedures directly in SAS) or in some cases, by calling custom SAS procedures in the BP. While such workarounds are possible, the Banff methodology team believes that the need for procedure linkage should be considered as part of a data editing strategy, and managed from the BP driver table.

C. Proposed solution

37. To address this issue, the Banff methodology team is proposing a solution that provides a clear and consistent default approach, combined with new functionality allowing users a higher degree of control and flexibility in the BP. Specifically, we propose the following changes to the BP:

- The removal of any existing, automatically generated procedure linkages. (Specifically, the link between DonorImputation and ErrorLoc.)
- The addition of a "status filter" tool allowing users to specify procedure linkages from the BP driver table.

Implementing the first change in the BP is trivial. Implementing the second change requires two modifications to the BP: the addition of a metadata column (StatusFilter) in the main driver table, and an additional step in the BP to modify the current status file before each procedure is run. The details of the proposed modifications are given in the Appendix, illustrated with an example.

38. By removing existing linkages, and allowing user-specific ones, we ensure that users can control exactly how an external procedure interacts with existing Banff procedures (or other external procedures) in the BP.

V. Record-level status flags

39. The objective of the Banff procedures is to identify and amend data values in a dataset; none of the Banff procedures offer the ability to alter the structure or size of the input dataset, by combining or splitting units, or variables. For this reason, it is sufficient to track the effect of a procedure at the value level, identified by the unique key <ID, FieldID>.

40. Users may also have a desire to track the effect on records, and not specific values. For example, after DonorImputation, a user may wish to know if a record was a donor, a recipient, or a failed recipient. In most cases, this information can be derived by comparing the instatus and outstatus files.

41. However, there are some procedures whose effects cannot be determined from the instatus or outstatus files alone; in particular this occurs when certain procedure outcomes can only be linked to a record (and not to fields within the record). ErrorLoc and Prorate both identify "rejected" records; these are records for which processing could not be completed – i.e., ErrorLoc could not identify which fields minimized the error localization problem within the user-defined time constraints, or Prorate could not properly amend records within user-specified parameters. In both cases, the *record* requires further treatment, but specified fields cannot be identified as FTI.

42. Other procedures also allow the option for users to exclude complete records from a process; this is done by specifying a data exclusion variable (DataExclVar) on the input set, with values of "E" (for exclusion) indicating a record that should be excluded.

43. We propose standardizing the treatment of record-level information by introducing the concept of a ***record status***. Similar to the existing field status, each procedure would optionally be able to import or export record status flags, and treat them accordingly. Incorporating record status flags into the current system does not require any additional data management; they could be included on the existing status files, and identified by blank FieldID values.

44. This concept is in the early stages of development. To be fully integrated, each Banff procedure would need to be reviewed so as to identify appropriate input and output record status codes. These codes would need to be standardized, so that the outputs from one procedure could be read as inputs in another, and appropriately handled. Adding this functionality would provide another layer of metadata information within the standardized Banff framework, and may allow for the integration of more diverse external data editing procedures into the Banff library.

VI. Conclusion

45. The majority of this paper has been devoted to a discussion of Banff procedures and the Banff processor, and how each can function as a modular, automated data editing tool. Our objective with this development is to provide users with an intuitive, flexible system that requires very little extraneous data management, allowing users to focus on overall data editing strategies without worrying about the hassle of implementation.

46. While these improvements should benefit current users and address certain identified issues, the driving motivation is to facilitate the integration of external functions into the Banff framework. Doing so is a key component of Banff's evolution in the context of modernization, providing users with an established, trusted data editing environment within which new methods can be tested and compared. The development of

a generalized assessment tool, along with the broadening of Banff's methodological scope, further positions the system as a powerful and comprehensive data editing tool. The Banff methodology team is currently focusing on developing and testing these new features.

VII. Acknowledgements

The author would like to acknowledge the other members of the Banff methodological team, Daniel Finch and Joël Bissonnette, for their hard work in the development of these research goals, as well as senior management (Steven Thomas, Susie Fortier and Jackey Mayda) for their vision and support. This paper could not have been completed without the valuable insight and feedback provided by Peter Wright and Steve Matthews.

VIII. References

Barboza, W., & Turner, K. (2011). Utilizing Automated Statistical Edit Changes in Significance Editing. In National Agricultural Statistics Service, Joint Statistical Meetings.

De Waal, T. (2003). Solving the error localization problem by means of vertex generation. Survey Methodology, 29(1), 71-80.

De Waal, T., Pannekoek, J., & Scholtus, S. (2011). Handbook of statistical data editing and imputation (Vol. 563). John Wiley & Sons.

Fellegi, I.P., and Holt D. (1976). "A systematic approach to automatic edit and imputation." Journal of the American Statistical Association, 71, 17-35.

Gray, D. (2018). Anticipating edit & imputation needs in light of Statistics Canada's modernization initiative. (Presentation) Statistical Inference for Complex Surveys, Montreal, QC.

Johanson, J. M. (2013) Banff Automated Edit and Imputation Applied to the US Hog Inventory Survey. (Presentation) Federal Committee on Statistical Methodology Research Conference

Pannekoek, J., Scholtus, S., & Van der Loo, M. (2013). Automated and manual data editing: a view on process design and methodology. Journal of Official Statistics, 29(4), 511-537.

Rancourt, E. (2007). "Assessing and dealing with the impact of imputation through variance estimation." Statistical Data Editing: Impact on Data Quality 3: 168.

Scholtus, S., Bakker, B. F. M., & Robinson, S. (2017). Evaluating the Quality of Business Survey Data before and after Automatic Editing. UNECE Work Session on Statistical Data Editing.

Statistics Canada (2017a). Functional Description of the Banff System for Edit and Imputation, Technical Report.

Statistics Canada (2017b). G-Est 2.01 User Guide, Technical Report.

Statistics Canada (2017c). Banff 2.07 User Guide, Technical Report.

Statistics Canada (2017d). Banff 2.07 Processor User Guide, Technical Report.

Stelmack, A. (2018). On the Development of a Generalized Framework to Evaluate and Improve Imputation Strategies at Statistics Canada. Submitted to UNECE Work Session on Statistical Data Editing

Terrie, L. (2018). Assessing the Automated Imputation of Missing and Erroneous Survey Data: A Simulation-Based Approach (No. 0108). Bureau of Economic Analysis.

Thomas, S. (2017). Future Development of Statistics Canada's Edit and Imputation System Banff. UNECE Work Session on Statistical Data Editing.

UNECE (2015). Generic Statistical Data Editing Models (GSDEMs) Version 1.0, October 2015. http://www1.unece.org/stat/platform/display/kbase/GSDEMs

Winkler, W.E. (2006). “Data Quality: Automated Edit/Imputation and Record Linkage.” U.S. Census Bureau, Research Report Series (Statistics #2006-7).

Xu, M., Kim, A. K., and Terrie, L. (2017). Automated Data Editing and Imputation for Surveys of Multinational Enterprises, a Banff Implementation. UNECE Work Session on Statistical Data Editing.

IX. Appendix

In this appendix we provide an example of a procedure linkage problem, and how it would be treated in the proposed modification to the BP.

We consider a dataset consisting of fifty records (ID: R01 – R50) and two fields (Var1, Var2). We suppose a user wants to run an automated data editing strategy within the BP, specified by Table 2. The columns are the typical columns from the BP driver table, as described in Section IV.A, with an additional column containing a description of each step in this hypothetical process flow.

Table 2 Banff Processor Driver Table (example)

JobID	SeqNo	Process	SpecID	Description
Example	1	ErrorLoc	ErrorLocSpecs1	Identifies missing fields and assigns an FTI status
Example	2	ErrorLoc	ErrorLocSpecs2	Identifies records failing the edit "Var1 <= Var2" and selects fields for imputation (FTI)
Example	3	Outlier	OutlierSpecs	Identifies outlier records for imputation (FTI) and exclusion (FTE)
Example	4	DonorImputation	DonorSpecs	Imputes fields with FTI status using nearest neighbour donor imputation (IDN)
Example	5	Estimator	EstimatorSpecs	Imputes any remaining FTI fields using mean imputation

After each process, the BP updates a current status file, containing any field statuses assigned by the previous processes. If a single field has been assigned multiple statuses, only the most recent one is kept. To illustrate how this works in practice, we consider the hypothetical current status file shown below, which is generated after process 4 (donor imputation) in the specified process flow. From the current status file, we can see that three fields were amended using donor imputation (status = IDN), two fields still require amendment (status = FTI) and one record has been detected as an outlier and flagged for exclusion (status = FTE).

Table 3: Current Status File after process 4

ID	FieldID	Status
R01	Var1	IDN
R02	Var1	IDN
R02	Var2	IDN
R03	Var1	FTI
R04	Var1	FTI
R05	Var1	FTE

The fifth procedure in the sequence is Estimator, which has been specified to perform mean imputation. Based on the specifications, Estimator will attempt to replace the identified fields (<R03, Var1> and <R04, Var1>) with the mean value of Var1 (over the dataset). Because Estimator automatically detects incoming FTE flags, field <R05, Var1> will be excluded from the mean calculation. Depending on the specifications given by the user, the fields with an IDN status may also be excluded.

Suppose that in the final process (mean imputation), the user wishes to impose the following restriction: only those records that are *still missing* should be imputed. The current status file does not provide this information; all we can deduce is that two fields are still flagged as FTI, meaning that they were selected for

amendment by one of the first three procedures, and failed to be amended in donor imputation (perhaps because donor imputation could not find a donor whose values would satisfy the edit requirements). Within the current version of the BP there is no method to automatically detect which of the values flagged as FTI are still missing, and to constrain Estimator to amending only these values.

This is a real, user-specified case of a "procedure linkage" need. In this example, the user would like to link an amendment procedure (process 5) with two other steps in the process (processes 1 and 4) but ignore the rest (processes 2 and 3). This level of control is not currently provided in the BP.

As outlined in Section IV.C, the Banff methodology team is proposing a solution through the use of a "status filter" tool. Implementing this functionality is relatively straightforward, as the BP maintains a "cumulative status file" throughout the process flow; this file contains the specific output status files of each individual process, along with the associated sequence number. For example, there are many *possible* cumulative status files associated with the current status file of Table 3; we present one possibility below:

Table 4: Cumulative status file

SeqNo	ID	FieldID	Status
1	R02	Var1	FTI
1	R02	Var2	FTI
1	R04	Var1	FTI
2	R01	Var1	FTI
2	R02	Var1	FTI
2	R02	Var2	FTI
2	R04	Var1	FTI
3	R03	Var1	FTI
3	R05	Var1	FTE
4	R01	Var1	IDN
4	R02	Var1	IDN
4	R02	Var2	IDN

From this sequence it is easy to identify which field is still missing after process 4; we simply compare process 1 (which identified missing values) and process 4 (which amended values) to deduce that <R04, Var1> is the only value still missing.

The current status file (Table 3) is simply the most recent status of each field in the cumulative status file (Table 4). In order for the user to exclude certain status flags from an input status file, they must be able to exclude records from the cumulative status file at each process step. One way to do this (the proposed method) is to add an additional column to the BP driver table, ***StatusFilter***, with which users can specify field status exclusions using standard SAS Boolean commands:

Table 5 Driver table with proposed StatusFilter column

JobID	SeqNo	Process	SpecID	StatusFilter
Example	1	ErrorLoc	ErrorLocSpecs1	
Example	2	ErrorLoc	ErrorLocSpecs2	
Example	3	Outlier	OutlierSpecs	
Example	4	DonorImputation	DonorSpecs	
Example	5	Estimator	EstimatorSpecs	SeqNo in (1,4)

Applying this filter to the cumulative status file (Table 4) and taking only the most recent status for each field would produce the following input status file for the final process, with only <R04, Var1> requiring amendment, as desired. Note that a StatusFilter of "SeqNo not in (2,3)" would have produced the same result.

Table 6 Current status file after status filtering

ID	FieldID	Status
R01	Var1	IDN
R02	Var1	IDN
R02	Var2	IDN
R04	Var1	FTI

One effect of this filter is that the field <R05, Var1> with a status of FTE is not in this status file. If the user wished to keep the FTE flags from process 3 (but ignore the FTI flags) they would need a slightly more complex, but still straightforward, StatusFilter command:

(SeqNo in (1, 4)) or (Status = FTE)

Note that every record on the cumulative status file is linked to the BP driver table via the SeqNo variable. It is therefore also possible to apply restrictions to the cumulative status file by filtering on the information in the driver table. For example, the following StatusFilter would achieve the same result, provided the BP was programmed to link the two tables together:

(SpecID = ErrorLocSpecs1) or (Process = DonorImputation)

The Banff team is currently undertaking a research project to develop a BP-like prototype that could offer the functionality above. The objective is to consult with users (primarily IBSP) to assess the usefulness of such a tool, and the intuitiveness / simplicity of the StatusFilter column. As with the standalone procedures, any proposed changes must be discussed with the Banff system engineering team to assess feasibility.