

BADVAM Manual

Blood Audit Data Visualization and Manipulation



NovaNAIT

8 Jan 2016

1 Background

The *Blood Audit (2015)* project was a joint project that sought to perform a detailed end-to-end audit of the costs entailed in collecting, processing and delivering whole blood to patients in Alberta's capital region. The level of detail collected

The *Blood Audit Data Visualization and Manipulation* (BADVAM) application was designed to afford the project sponsors and other professionals and decision makers with the capacity to see both the detailed values that were collected as well as aggregate totals and averages in a familiar interface. BADVAM was further designed to permit users to manipulate the data by editing, adding and removing data points and system parameters in order to facilitate planning and modelling the system with an interest to improving efficiency and reducing extraneous cost.

This application was further designed to serve as a test bed and foundation for the generation of similar studies and audits in the future, and the useful delivery of detailed audit findings to end-users.

2 Installing and Launching BADVAM

The current version of BADVAM (1.1) is delivered as a zip archive containing the application files, data, and default settings. It may be expanded (unzipped) into any directory the user has the rights to access and run executables from. It is necessary to retain all files in the provided in the provided directory structure in order for the application to function properly (see Figure 3.1 for a list of the expected files).

In order to run the program, simply double-click on the `badvam.exe` executable program. It should have a pair of red droplets as its icon (see Figure 2.1).

3 Application Directory Contents

The directory where the application is stored may be named anything you like, but must be in a location the user can access and modify files in. However, within that directory, BADVAM expects certain files to be

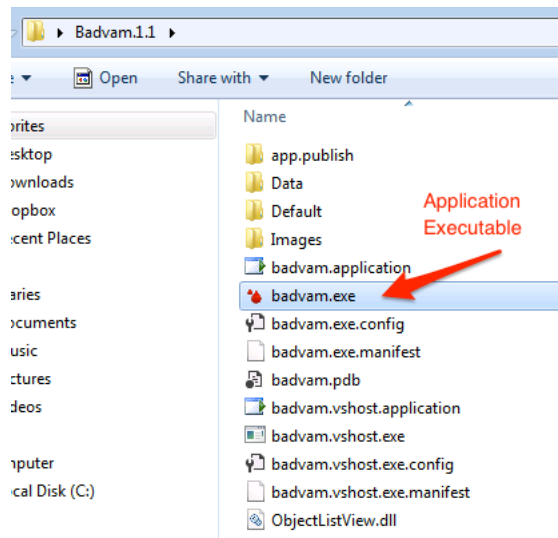


Figure 2.1: badvam.exe Executable in Application Directory

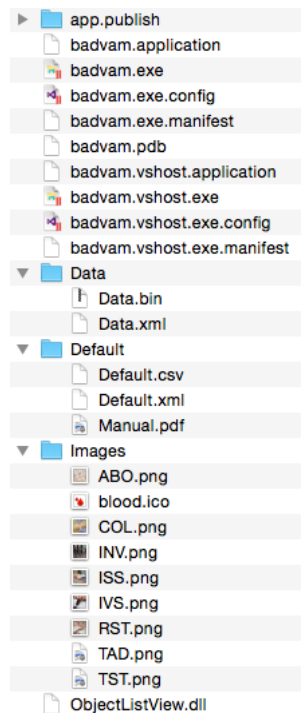


Figure 3.1: Application Directory Contents

present (see Figure 3.1). Missing files or paths will generate errors from the application. Files of particular interest are listed below.

- `badvam.exe` - The application executable. Starting this starts the application.
- `ObjectListView.dll` - A custom control used for the Site Panel items lists. It is documented at [ObjectListView: A Much Easier Way to use a ListView](#). All other application components are standard .Net components from Microsoft.
- `Data` - Directory containing cache from the last time the application was run. This is where BADVAM keeps the state of the data by storing the current contents when it is closed, and reloading them when it is reopened.
- `Data/Data.bin` - The main configuration file for the application, including the names and ordering of sites, location of images, global options, etc. This is maintained by the application itself.
- `Data/Data.xml` - A file containing all the data (line items) for the sites identified in `Data.bin`. This is generated when the application is closed and stores all data entries.
- `Default` - Directory containing the default data that was shipped with the application, based on the results of the Blood Audit project.
- `Default/Default.xml` - Default site list, images, ordering, system configuration settings.
- `Default/Default.csv` - Default data values.
- `Default/Manual.pdf` - This document.
- `Images` - Directory containing images such as those used to decorate the Site button icons in the Home Panel.

4 Application Description

BADVAM is a standard Windows application, built with Visual Studio 2015 in C# using the .Net libraries. It was tested and evaluated in a Windows 7 environment, but is expected to work in any modern Windows environment. Issues, errors, bugs and enhancement requests may be reported to the application maintainer using the website identified in the *Help and Support* section later in this document.

When the application first opens, it will load data from its own *Data* directory (any changes to its data are saved back to that directory when the application closes). The user may then view, modify or add data to the system in order to facilitate their own planning and analysis. Modified versions can be stored and loaded from their own files, and importing and exporting data to or from a human-readable CSV format supported by most spreadsheet programs is also supported. The original default data may always be restored.

The application architecture was designed with an interest to permitting it to serve as the foundation for visualizing data in other similar auditing projects. Please contact us through the support interface if you are interested in using this or similar technology in visualizing audit data in different regions or organization types.

5 Home Panel

When the application first opens, you will be presented with the *Home Panel* (see Figure 5.1). This is designed to provide an overview of the costs in the entire system. Components of the interface include:

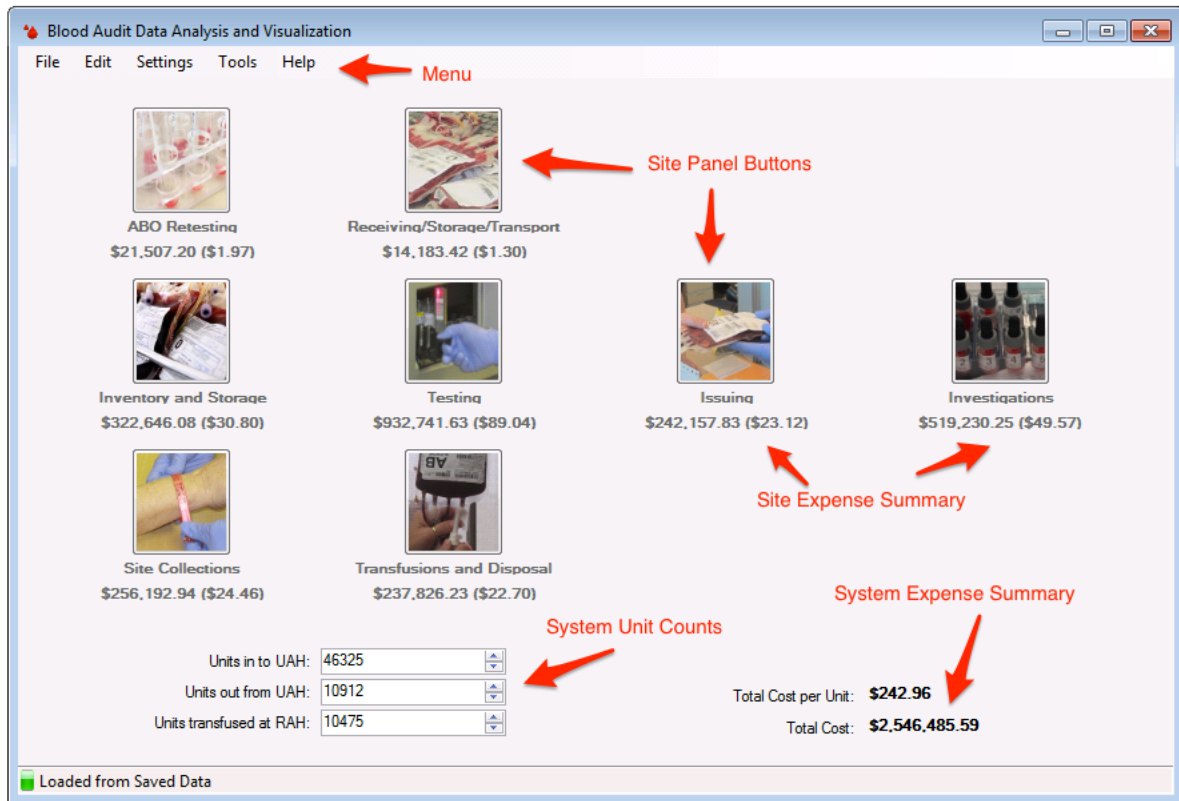


Figure 5.1: BADVAM Home Panel

- A toolbar *Menu* for controlling various tools and settings in the application (described below).
- *Site Panel Buttons* that indicate the various sites involved in acquiring, testing, distributing and delivering blood transfusion services. Clicking on the buttons will take you to the indicated site panel (see next section).
- Underneath each panel button, there is a *Site Expense Summary* which details the total annual cost (and a parenthesized per-unit cost) for expenses at that site.
- Numeric entry fields for *System Unit Counts*, which are used for generating per-unit costs. The three values are:
 - *Units in to UAH* - Total number of blood units acquired by the system per year from Canadian Blood Services for the Capital Region at the University of Alberta Hospital. This is used to track the per-unit cost for blood processing at UAH.
 - *Units out from UAH* - Number of blood units transferred from the University of Alberta hospital to the Royal Alexandra Hospital for transfusion per year. This is used to determine what proportion of UAH costs should be applied in auditing the costs per-unit for transfusions at RAH.
 - *Units transfused at RAH* - Number of blood units actually transfused to patients at RAH in one year. This is used to calculate the final per-unit end-to-end costs for transfusing.
- A *System Expense Summary* that details the Total Cost per annum and Total Cost per unit for blood transfused at RAH.

6 Site Panel

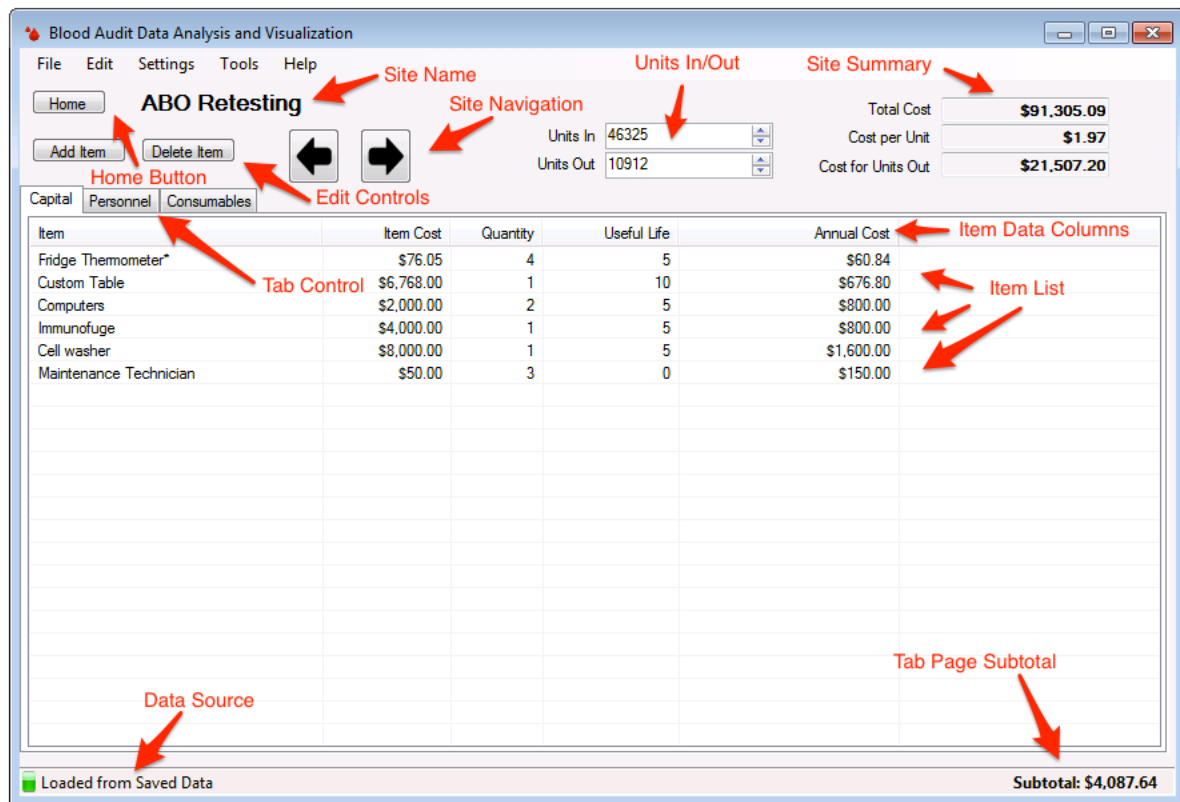


Figure 6.1: BADVAM Site Panel

Clicking on a Site icon in the Home Panel or using the navigation arrows to cycle from site to site will bring up the Site Panel for the site of interest. This provides details of the costs associated with blood supply activities performed at that site.

Important elements of this interface include:

- The *Home Button* which will return you to the Home Panel (see previous section).
- The *Site Name* indicating which site's data you are currently observing.
- The *Site Navigation* buttons (arrow buttons) that allow you to navigate to the next or previous Site's data. This navigation is circular, so clicking one direction repeatedly will eventually return you to the site you started with. You can also use the *Home Button* to return to the Home Panel and from there go to a specific Site Panel.
- *Units In/Out* controls allow you to view or modify the Units In and Units Out counts when looking at one of the UAH sites. Note that this is a global setting and changes will also be reflected on the Home Panel and in other sites' per-unit costs. On RAH sites, only the Units Transfused count is visible as it is the only one relevant to that site.
- The *Site Summary* which details the total cost for that site's activities, the cost per-unit from that sites, and the total cost for units out from the site that ultimately get transferred to RAH. Only the relevant fields appear on the summary for a given site.
- The *Data Source* for the data currently being viewed. If data was loaded from a file, that file's path will be shown. "Loaded from Defaults" means the data was reset to default during the current session. "Loaded from Saved Data" means the data was loaded from the data directory that was saved when the application was last used and closed.

The main focus of the Site Panel is the *Item List* which has three tabs with three distinct listings of cost items for the categories of expense: *Capital* (amortizable expenses), *Personnel* (salaried staff) and *Consumables* (simple expenses for expended supplies). Each tab has a discrete set of fields detailing information for that type of expense. Some fields are calculated from other fields and therefore cannot be edited.

- *Capital* items have 5 fields:
 - *Item* - Descriptive name for the capital item.
 - *Item Cost* - Cost to purchase one item.
 - *Quantity* - Number of items in service.
 - *Useful Life* - Number of years over which item cost should be amortized.
 - *Annual Cost* - Annual cost of ownership (calculated).
- *Personnel* items have 5 fields:
 - *Title* - Descriptive name for the profession or job category.
 - *Salary* - Annual salary for that job category.
 - *Benefits %* - Percent of salary as an additional expense for benefits. A typical value would be 18.
 - *Benefits* - Total annual benefits cost for that profession (calculated).
 - *FTEs* - Total number of Full-Time Equivalent person-years for that profession at the site. Four members of a profession full-time annually would be 4. A professional who works in 2 different sites might have their FTE for that site recorded as 0.5.
 - *Cost* - Annual salary and cost for that class of personnel at that site (calculated).
- *Consumable* items have 5 fields:
 - *Item* - Descriptive name for the item.
 - *Item Cost* - Cost to purchase one item.

- *Quantity* - Number of items consumed in one year.
- *Annual Cost* - Total cost of that type of item (calculated).
- Hovering the mouse over an expense item will display an information note for items that have such data recorded.
- Editable fields in the *Item List* can be edited if you double click on the value of interest. Hitting return or clicking elsewhere will save your changes.
- Double clicking on the column header in the *Item Data Columns* will sort the list by the values in that column. Double clicking again will toggle between ascending and descending sorts.
- Finally, there are two *Edit Controls* that allow you to add and remove items from the displayed item list. Selecting one or more rows on the displayed item list and clicking *Delete* will remove those lines. Clicking *Add* will add a new editable item entry named "new item".

7 Menu Controls

The application

Additional user control is implemented with the use of a menu strip with the following functions:

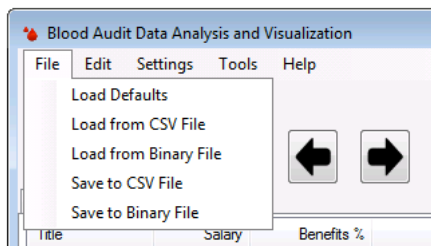


Figure 7.1: File Menu

- *File* → Load Defaults - Restore the system defaults.
- *File* → Load from CSV File - Load data from a stored CSV text file.
- *File* → Load from Binary File - Load data from a binary file previously stored by BADVAM.
- *File* → Save to CSV File - Save current data to a human-readable and spreadsheet editable CSV file.
- *File* → Save to Binary File - Save current data to a binary file that can be reloaded into BADVAM

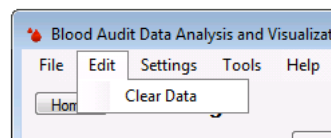


Figure 7.2: Edit Menu

- *Edit* → Clear Data - Remove all data, but keep site organization.

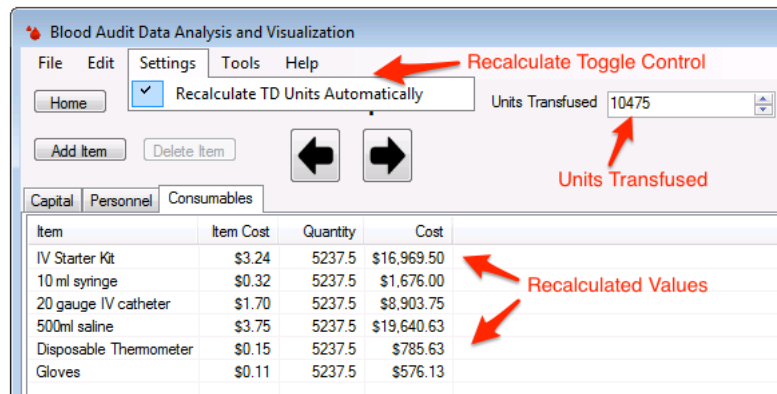


Figure 7.3: Settings Menu

- *Settings* → Recalculate TD Units Automatically - Toggle whether or not the consumable item counts on the *Transfusion and Disposal* site should be recalculated to reflect changes in the *Units Transfused*. This simplifies calculations as most of those costs are based on a per-unit cost.

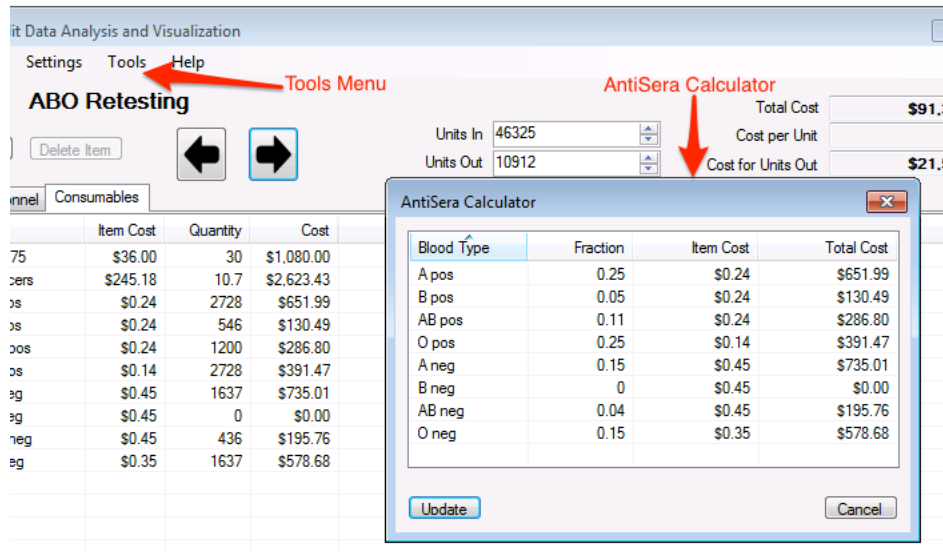


Figure 7.4: Tools Menu

- *Tools* → AntiSera Calculator - Open the *AntiSera Calculator* utility which allows you to recalculate Antisera costs on the *ABO Retesting* site by modify unit costs or blood type fractions.
- *Help* → About - Display the program splash screen which lists the organizations and people responsible for the project.
- *Help* → Manual - Display this document in your system's default PDF viewer.

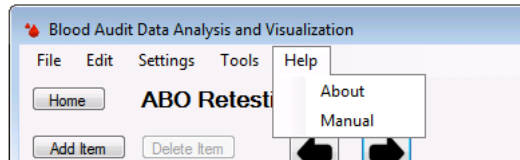


Figure 7.5: Help

8 Help and Support

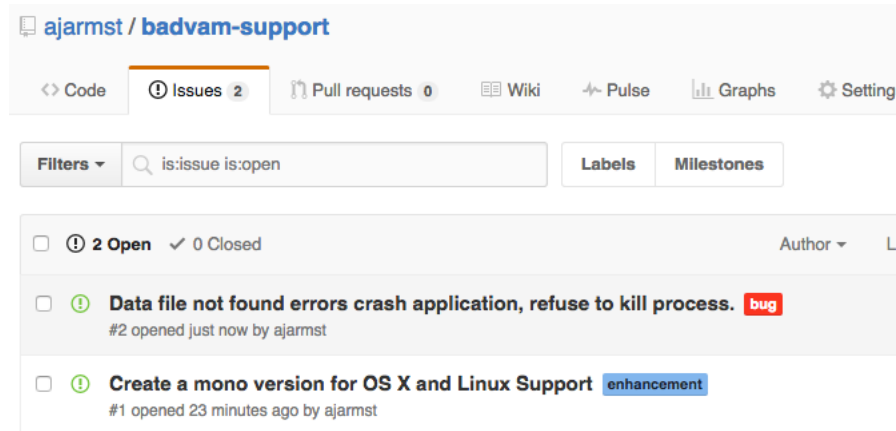


Figure 8.1: Submitting Issues at Github

If you encounter problems using BADVAM, please use the *Github* issue tracker at <https://github.com/ajarmst/badvam-support/issues>. You may need to create a free Github account to post. When posting there, please be as specific as possible about the problem you are encountering. Particularly valuable information includes:

- What you are trying to do with the application.
- What specific actions you took (clicking on controls, typing, menu selections, etc.)
- What you expected to happen.
- What actually happened.
- Any error messages or warnings that were generated. Screenshots in this case are particularly helpful. See Figure 8.2 for an example of such a screenshot. Include the exact text of any error message if you cannot provide a screenshot.

Copyright © 2016, NovaNAIT. All rights reserved.
 Queries and comments concerning this document may be directed to the author, AJ Armstrong, at aja@nait.ca

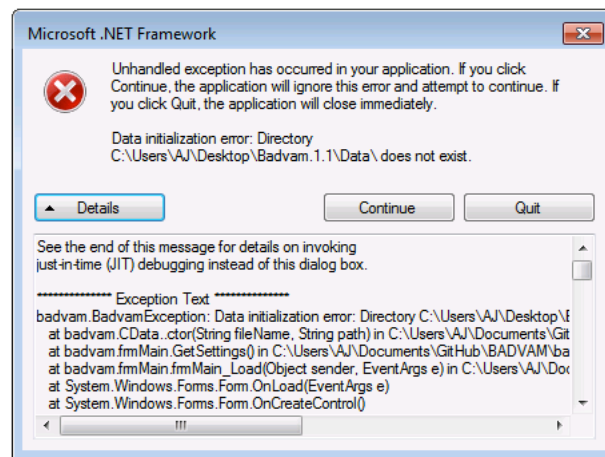


Figure 8.2: Detailed Error Message Screenshot