

Spreadsheet Design and Validation for the Multi-User Application for the Chemistry Laboratory

Part I

by

Dennis Cantellops, Evelyn Bonnin and Anne Reid

Southeastern Regional Laboratory

Introduction

Two LIB's covering design and validation of analytical spreadsheets have been prepared. This document covers the general principles of spreadsheet application to be employed by several users (multi-users), Part I. Another document to be submitted, Part II concentrates on spreadsheets to be used by only one user (single-user). The design aspects of spreadsheets and the verification and validation documentation for reporting in a regulatory environment are discussed. The discussion pertains to applications of the Microsoft Excel spreadsheet program.

For multi-users, special care is required because the spreadsheet must be "bulletproof" to minimize the possibility of misuse. The application of the spreadsheet must be clearly defined, which means providing additional formatting and instructional elements. The degree of extra effort depends on the experience level of the expected users, but always, when multiple users are involved, the spreadsheet requires stringent validation and its documentation. Documentation for the multi-user spreadsheet provides evidence that formulas, macros, data integrity, accuracy, reliability, and operational procedures (such as instructions for user) will perform as expected.

The usefulness of software (such as Power Pak Utility-PUP v5.0) that audits the workbook file and generates validation documentation will be discussed. In the production of both, the *analyst* (single-user) and *users* spreadsheets effective and appropriate principles and procedures must be applied to all stages of the analytical workbook files. At the design stage, such principles include design standards, clarity of formulas, documentation, and user-friendliness.

Furthermore, when the basic principles of software engineering techniques are applied to the construction of spreadsheet models, many errors are reduced or eliminated.

Note: *The Laboratory Information Bulletin is a tool for the rapid dissemination of laboratory methods (or information) which appear to work. It does not report completed scientific work. The user must assure himself / herself by appropriate validation procedures that LIB methods and techniques are reliable and accurate for the intended use. Reference to any commercial materials, equipment, or process does not in any way constitute approval, endorsement, or recommendation by the Food and Drug Administration.*

Observed Spreadsheet Application Problems

The following spreadsheet problems were discovered during an FDA inspection of a QC analytical laboratory in a pharmaceutical firm. The QC laboratory failed to follow current good manufacturing practice (cGMP) (1) and good laboratory practice (GLP) (2) regulations. Also, basic principles of multi-user spreadsheet design, validation and documentation were not apparent. Spreadsheet files were not maintained and many were deleted. The SOPs for the spreadsheets were absent and validation and documentation procedures were not in use. These spreadsheet problems can also be present in FDA laboratories. Specific problems included the following:

- Nearly all of the spreadsheets in the laboratory showed rounding-off errors and none used the Excel rounding function, when needed to avoid rounding-off discrepancies between the original and check calculations by the second analyst.
- Formulas in the analytical procedures did not follow the Excel equations.
- Conversion factors were not expressed in the analytical procedures.
- The formulas used by the check analyst for manual calculations were not the same as those used in the Excel spreadsheets.
- The pre-determined specifications or limits were not shown in the spreadsheets.
- Spreadsheets were not clearly documented. For example, the product declaration was not indicated, replaced by only a number in a cell.
- Units were expressed as numbers without descriptive labels such as mg/mL or mg/g.
- Sample weights were described as sample volume and areas as ratios.
- Spreadsheets did not contain provisions for security and integrity of data.
- Spreadsheet applications were not protected from changes, meaning analysts could freely change labels and formulas.
- Regression analysis was calculated with the y and x axis inverted in the Excel formula, which generated erroneous slope and intercept results.

These analytical spreadsheet mistakes can be eliminated if the analyst (developer) ensures that each element in the spreadsheet application is uniquely and unambiguously defined.

As a minimum, a simple spreadsheet validation documentation should include the following items (3):

- A description of what the program does and proper instructions.
- Use of color-coding for data-entry.
- A description of the mathematical formulas used .
- An explanation of the relationship of the formulas in the analytical method to the Excel equations.
- Listings for any macro programs.

- Acceptance criteria, including product declarations and specifications or USP limits.
- Test sheets with anticipated and actual results, signed and reviewed, that have been verified by manual calculations.
- Security and password maintenance.
- Operating system used, date of installation and version number.
- Documentation of the spreadsheet program, including any service packs or add-ins. Some service packs such as the PUP v5.0 or the additions of macros may not be reflected in the software program version designation but should be documented all the same.

Good Laboratory Practice (GLP) Regulations

The use of spreadsheets to judge the quality of a product (4) is controlled by cGMP's, GLP, regulations (1) (2) and ISO 17025 accreditation (5). The following are some of the requirements established by GLP regulations:

- **Configuration Management.** Quality assurance for managing software addresses all documentation associated with the system and is applied during all operational phases of software use, including the development and maintenance phases. It includes identification of changes, implementation of changes as reported, and revalidation of formulas and workbook files at specified times or when the developer introduces a new version of the spreadsheet. If modifications, up-grade or changes to the software have been implemented, perform regression analysis to determine what portions of the software that had not been changed might have been adversely affected by changes elsewhere in the software. In addition, conduct regression testing on those portions to determine if any new errors or problems were introduced.
- **Written Standard Operating Procedures (SOPs).** SOPs are defined at 21 CFR 58.81 (a) (2), as "A testing facility shall have standard operating procedures in writing setting forth non-clinical laboratory study methods that management is satisfied are adequate to insure the quality and integrity of the data generated in the course of a study. All deviations in a study from standard operating procedures shall be authorized by the study director and shall be documented in the raw data. Significant changes in established standard operating procedures shall be properly authorized in writing by management."
- **Completeness of Data.** Validated analytical applications need a performance-monitoring program to establish data integrity and completeness. Compliance Policy Guide (CPG) 7132a.07 (21 CFR 211.68) defines the program for input/output (I/O) checking. The I/O test involves inputting a set of data packages that have an accurately known solution and checking this known solution against the result given by the software application. In addition, all workbook files should be preserved, controlled and maintained for future audits against the reviewed printed sample reports. A backup-and-restore system should be in place to ensure data completeness

and security. A backup-and-restore SOP provides detailed instructions as to how to recover the spreadsheet files after a system failure, designate the backup location, and define the schedule-and-restore methodology. The SOP should identify the person responsible for retaining backup records, including schedules or logs, the type and storage location of the media used, system failures and recovery logs.

- **Raw Data.** Raw data is defined at 21 CFR 58.3 (k) (2), as “any laboratory worksheets, records, memoranda, notes, or exact copies thereof, that are the result of original observation and activities of a non-clinical laboratory study and are necessary for the reconstruction and evaluation of the report of that study.” In addition to hand written observations in ink on paper, “raw data may include photographs, microfilm, or microfiche copies, computer printouts, magnetic media, including dictated observations, and recorded data from automated instruments”. In the chemistry laboratory, the raw data from notebooks or paper worksheets can be entered manually into the computer spreadsheet program or can be directly collected (from balances, GC, LC, etc.) into the spreadsheet program, by means of an interface or a Laboratory Information Management System (LIMS). Under those circumstances, the spreadsheet becomes raw data. A worksheet generated by a spreadsheet program can be designed to be used as a fill-in-the-blanks worksheet for raw data. After the raw data is entered manually into the paper worksheet (fill-in-the-blanks worksheet), the worksheet becomes original data, which is the result of original observations. The raw data worksheet can not be remade if errors are found. Errors cannot be erased or overwritten; a line must be drawn through an incorrect entry and the correct figure or word written nearby and initiated. Also, significant data must not be discarded without explanation. Discarded analytical data must be crossed out, initiated, and the reason for discarding the data explained. An example of the raw data worksheet will be presented in Part II of this LIB.
- **Secondary Data.** The secondary data worksheet is used for the calculations and results, which utilize the data from the raw data worksheet (standard and sample weights and dilutions, sample dilution factor, etc.) previously developed and printed, which included the original observations. It can be remade if errors are found. To verify the secondary data spreadsheet (calculation and results sheet), data in the spreadsheet are checked against the printed and used raw data sheet (fill-in-the-blanks *Excel* Worksheet in which the blanks had been manually written). The results of the verification must be documented, and the check analyst must sign for approval. A laboratory notebook in which raw data is entered manually also becomes original data and copies can be attached to the spreadsheet printout. Figures 2 and 3 are examples of the secondary data worksheets.

The workbook files containing the raw data (fill-in-the-blank form) and secondary data worksheets should be preserved, controlled, and maintained for future audits of

the electronic file. In addition a copy of the completed sample printout should include the name and “saved” date for the workbook file, and the software name, version, and upgrade designations.

- **Security.** 21 CFR Part 11; Electronic Records; Electronic Signatures (6) “Sec. 11.10 Controls for closed systems. Users who use closed systems to create, modify, maintain, or transmit electronic records shall employ procedures and controls designed to ensure the authenticity, integrity, and, when appropriate, the confidentiality of electronic records. Such measures ensure that the signer cannot readily repudiate the signed record as not genuine. Such procedures and controls shall include the following: (a) Validation of systems to ensure accuracy, reliability, consistent intended performance, and the ability to discern invalid or altered records. (b) The ability to generate accurate and complete copies of records in both human readable and electronic form suitable for inspection, review, and copying by the agency. Individuals should contact the agency if there are any questions regarding the ability of the agency to perform such review and copying of the electronic records.” Other part 11 requirements may be implemented as well (6).

For example, the access control to the computer and the Excel workbook files and audit trails (Section 21 CFR 11.10(e)) should be implemented as follows:

(a) For computer access control, the Agency established security in accordance with Federal mandates (Computer Security Act of 1987, Computer Fraud Abuse Act of 1986, and others, see <http://intranet.fda.gov/oirm/itsecurity/useful.htm>).

The Agency is required to provide “adequate security” to manage potential risk to its automated information. The Southeast Region Information Technology Services (SRITS) manages the computer security program. The Information System Security Officer (ISSO) serves as the Agency focal point to direct and oversee the security program. The ISSO will develop policies, procedures, and guidance establishing, implementing, maintaining and overseeing requirements for the Agency’s information technology (IT) security program. For example, the agency and the SRITS have established access controls at various levels. The ORA’s wide area network (WAN) is a closed system protected by two internal firewalls. This closed system is contained within another closed system that consists of the HHS WAN (the departmental level). The ORA’s WAN has a secured gateway to the Internet through the various firewalls. At the ORA’s WAN level, all computers (laptops as well as desktops) have been migrated to Microsoft’s Windows 2000 Professional operating system (OS). This new OS has improved security features. The most important ones are:

- ✓ Authenticated Logon - Users now must enter a username / password combination in order to access the files and resources in any computer. If a user does not have

an account in the ORA network domain, then the user would not be able to access any files or computer resources.

- ✓ File Level Security - All hard disks will be using NTFS (New Technology File System from Microsoft) as the file system (not the common FAT file system). This provides additional security features for file access over previous file systems such as FAT (File Allocation Table) or FAT32. Files stored in local hard drives can be secured by using access lists. Administrators can designate who can access a specific file and what level of access the user can have (read, write, change, delete).
- ✓ Audit Trails - NTFS and Windows 2000 provides features that allow the system to log the use of a file or folder.
- ✓ File Encryption - Users can encrypt the data files to protect them from unauthorized access. The highest encryption algorithm is being implemented (128-bit).

In addition, data files stored in network drives are also protected from unauthorized access. File servers use NTFS as the file system and data is backup every night.

(b) For access control to Excel workbook files applications, the developer should implement adequate security. For example, Excel has several levels of protection that can be applied to a workbook file. The highest level of protection is set on the file level. A password can be set in order to open, access and save a workbook. After the workbook file is open, security options can be set inside the application file. There are three levels of security: workbook, worksheet, and object. Protecting the workbook prevents the user from inserting new worksheets. Protecting the worksheet prevents the user from changing the contents of a worksheet, such as formulas. Protecting the object prevents the user from deleting, moving or resizing objects.

In addition, spreadsheet applications should be protected with the safeguards identified in 21 CFR Part 11. These include validation with its documentation; security, and audit trails (Section 21 CFR 11.10(e)). The security, audit trails, and true electronic signatures can be implemented by software and services such as what is available from the company Wimmer Systems (www.wimmersystems.com). For example, Wimmer Systems can take our verified *Excel* application spreadsheet, add the compliance package, and when finished, install the application and provide training.

Also, during an in-house audit, the laboratory must provide the validation documentation and copies of the applications in both hard copy and electronic form as per 21 CFR 11.10 (a) and (b), if the copies are requested by the auditor. Therefore, Part 11 requires that laboratories be able to give the auditor electronic copies and hard

copies of their electronic records. Providing copies of the software program (for example Microsoft *Excel* program) that produce the records is not required.

In addition all workbook files (single-user and multi-user) should be preserved, controlled and maintained for future audits against the reviewed printed sample reports (single-user) or the validation documentation (multi-user). Otherwise, if workbook files were not controlled, problems might result (e.g., paper printouts or printed reports might be challenged as being incomplete or inaccurate representations of the corresponding electronic records).

- **Training.** Training is described by 21 CFR 58.29 (a) (2), as “Each individual engaged in the conduct of or responsible for the supervision of a non-clinical laboratory study shall have education, training, and experience, or combination thereof, to enable that individual to perform the assigned functions.” In spreadsheet training, it is very important to properly implement the analytical spreadsheet guidelines (SOP) and ensure worksheet standardization. If no guidelines or procedures are implemented, the spreadsheet development may go out of control, increasing the errors among the applications.

Quality Standards and Guidelines (See ISO/IEC 17025 Guideline)

The “General Requirements for the Competence of Testing and Calibration Laboratories”, (International Organization for Standardization ISO 17025) (5). Sections 3. Document Control, 12. Control of Records, and 4.7 Control Data, includes several paragraphs on the manipulation of quality and technical records and the use of computers. Where computers or automated equipment are used for the capture, processing, manipulation, recording, storage or retrieval of calibration or test data, the laboratory shall ensure that:

- Computer software is documented and suitably validated as being adequate for use.
- Procedures are established for protecting the integrity of data; such procedures shall include, but not be limited to, integrity of data entry or capture, data storage, data transmission and data processing.
- Computer and automated equipment are maintained to ensure proper functioning. Environmental and operating conditions necessary to maintain the integrity of test and calibration data must be provided.
- Appropriate procedures for the maintenance of security of data including the prevention of unauthorized access and/or amendment of computer records are established and implemented.
- Procedures are established to describe how changes in documents maintained in computerized systems are made and controlled.

General Spreadsheet Design

Spreadsheet structure is fundamentally similar to a computer program (7). Formulas in a spreadsheet are basically a form of computer programming. Therefore, the basic application of software engineering techniques and principles shall be applied in the construction of spreadsheet models. For example, instead of creating the workbook file first and then checking for errors, one should check for errors at various stages of the development process. This approach of stage-by-stage component testing is a software engineering principle. A spreadsheet model is a set of worksheets and macro sheets designed to evaluate and organize data or to solve a particular problem (8). Such a model should meet two design objectives:

- ***Guaranteed correctness.*** If you cannot guarantee that your model always yields the right answers, it is not effective and may actually be dangerous. Design your model to make errors immediately apparent.
- ***Adaptability.*** Most likely, the task you want your model to perform will change in time. Design the model so that you or others can easily improve it.

The following principles will help achieve these design objectives:

- ***Use a vertical, modular design.*** In a given spreadsheet use only one or two screen widths of columns, containing as many rows as needed. A strict vertical scheme promotes a clearer flow of calculation and provides rapid recalculation. Worksheets should be organized into manageable sections, each occupying a horizontal slice of the generally vertical worksheet. Each section should serve a single purpose. Try to make all sections the same width and no more than two screens high. Clearly distinguish each section from the adjoining ones by using cell formatting, colors, cell borders or separate sheets for each section.
Every formula should refer only to the cells above it. As a result, calculations should go strictly downwards from raw data near the top of the worksheet to the results at the bottom.
- ***Use a standard layout.*** A standard worksheet layout will facilitate maintenance. It should contain the following separate sheets:
 - ✓ ***Documentation:*** This sheet includes critical documentation about the rest of the worksheet; title, purpose, and author, a list of other worksheets or macros that must be opened for the model to work, and a table of contents that outlines the order of the sections in the worksheet. There should be no calculations in this section.
 - ✓ ***Menu:*** This sheet can include the instruction for the user and all the worksheets associated with the application. By pressing a button in the worksheet, a dialog box can open showing the user instructions. The instructions can include the location of data-entry cells or drop-down lists or other pertinent information to help the user move around the application.

- ✓ **Assumptions:** This sheet contains any initial assumptions used in the worksheet formulas.
- ✓ **Document assumptions.** Critical assumptions about formulas should not be hidden in the middle of the worksheet. Instead, clearly document all the assumptions in a special assumption section that helps everyone who looks at the worksheet to understand how the results were derived, and the inherent limits.
- ✓ **Data:** This sheet includes the raw data, such as records of data from which the model will perform calculations.
- ✓ **Calculations:** Include as many calculation sheets as needed to process the data. As mentioned, each section should be based only on the results of the previous sections so that calculations progress through the worksheet from top to bottom. Break the task into manageable segments. Divide the worksheet's task into a series of calculation steps, and then plan one section on the worksheet for each step. As each section is developed, record its purpose. For example, you might enter text such as "This section refers to the database on the supporting worksheet and summarizes results from drug or food samples for a single month"; see Figure 2; see statement in cell comment. This helps other users to understand the formulas in the model.
- ✓ **Results:** The last section should contain the final results. Along with the assumptions section, this section should provide a comprehensive summary of the model's calculations.
- **Write clear formulas:** This reduces calculation errors and makes the model easy to maintain. You should:
 - ✓ Spread complex calculations over several cells instead of trying to pack them into a single one.
 - ✓ Use a text label in an adjacent cell or insert a cell comment whenever someone might misinterpret what a number means.
 - ✓ Always document how the formulas work. For more complex explanations of the thinking behind a formula, use cell comments, which are available when required.
- **Include controls and checks.** To assist in meeting the objective of guaranteed correctness, explicit controls using the functions "IF" and "ERROR" should be included as needed.

Examples:

- ✓ In tables of data, ensure that the sum of the row totals equals the sum of the column totals.
- ✓ Where input data falls within a certain range or limit, use an expression of the type:

"=IF(K35<80,"Fail","Pass")"

If K35 is less than 80 the formula will return to “Fail” or if is above 80 it will return to “Pass”.

- ✓ Use error condition functions such as “IF(ISERROR ...)” to prevent an error display (such as #DIV/0!) in the cell if the formula returns an error. This is useful if formulas are used to perform calculations on cells that might have no data. As an example the following formula calculates a percentage “=ROUND((E31/D4)*100,1)”, If cell D4 contains 0 or is empty, the formula will display a #DIV/0 error. To display a blank cell when D4 contains 0 or is empty use the expression: “=IF(ISERROR(ROUND((E31/D4)*100,1)), "", ROUND((E31/D4)*100,1))”. Also, this function can eliminate many other error values such as #N/A, #VALUE!, #REF!, #NUM!, #NAME?, or NULL!.
- **Ensure that the model is easy to use:**
 - ✓ Use cell protection to prevent users from changing critical cells.
 - ✓ Provide clear instructions to the user by using text boxes, justified or wrapped text, and graphic objects. Always provide a brief set of instructions in the title / instructions / contents section at the top of every worksheet. Use the cells comments feature.
 - ✓ Use color on the worksheets to show which values can be changed by the user. Tell the users in the instruction section that they are allowed to change anything that appears, say, in blue.
 - ✓ Use brackets rather than the minus sign to show negative or adverse figures.
 - ✓ Use borders, underlining and shading to make reports clearer.
 - ✓ Automate common operations using the macro recorder, and assign the macro to buttons or icons on the worksheet. Document the macro.

Safety Design Procedures

The following are safety features for developing an analytical spreadsheet (9).

- **Use drop – down lists whenever possible.** This feature restricts the user to valid entries, when they are known beforehand. It is created by the sequence: *Data* → *Validation* → Choose “*Setting*” Tab → in “*Allow*” choose “*List*” → in “*Source*” give the cell range where the allowed choices will be written. A message will be activated when the cell is addressed. For setting up the message, select *Data* → *Validation* → choose *Input Message* → enter short instructions in the window *Title*. Optionally, choose: *Error Alert* and write the correction instructions in the window: *Error Message*.

- **Introduce the units in all appropriate cells.** This feature allows creation of new number formats. For example, units (e.g. mg/mL) used in analytical methods can be added by selecting from the Excel main menu bar: *Format* → *Format Cells* → Choose “*Number*” → Choose *Category* “*Custom*” →: Type: *General*. In the window where *General* is typed, type the entry: “mg/mL” just after *General* and press *OK*.
- **Format numbers.** Use the *Round* function and specify the number of decimal places to which the value should be rounded-off. For example, a formula to calculate an average with three decimals would be:

=ROUND(AVERAGE(Range), 3)

Never round to fewer than five places (including decimal places). For example, it is permissible to round the average to 87.333 but not 87.33. This criterion is used for rounding-off the values of intermediate calculations. When rounding by local laboratory policy, this policy may be used. The weighing and the chromatographic peak areas should be reported with all their digits. The final results should be rounded to the least significant digit (LSD), using the appropriate official guidance. For example the USP 24 / NF 19 page 4 provides the following rules:

- ✓ Establish the position of the LSD.
- ✓ If the digit to the right of the LSD is less than 5, eliminate all remaining digits after the LSD.
- ✓ If the next digit is larger than or equal to 5, increase the LSD by one and eliminate all the digits beyond the LSD.

The main reason for rounding the data is to provide safety in checking the displayed results against hand calculators. In the absence of the rounding function, Excel will guard additional decimals which might produce discrepancies with the hand calculator data. The Round function should be included in the formula where rounding is performed.

- **Calculations.** In the calculation and results worksheet (secondary data, attachment B3), each calculation should be based on the results of the previous calculation so that the calculations progress through the worksheet from top to bottom.
- **Add supplemental cell comments.** Use the cell comments whenever necessary for notes or instructions. To insert a comment in a cell, use *Insert* → *Comment*. When the mouse pointer moves over a cell that contains a comment, the comment becomes visible. See a comment example on attachment C2. (Cells that have a small red triangle in the upper-right corner have a comment attached).
- **Color-coded cells.** Use liberally. For example, all data entry cells could be yellow-coded, while the drop-down lists may use blue cells. This is done by using the *Format* → *Cells* → *Patterns*.

- **Add path to footer.** The workbook file name is inserted into the left footer of each worksheet. Implement with: *File* → *Page Setup*. The footer can advantageously be used to insert operating instructions for the worksheet. Also, the creation date, last modification date, and version number can be inserted in the footer.
- **Protection.** By default, *Excel* has all cells locked. First, unlock all data-entry cells in each worksheet by using the *Format* → *Cells* → *Protection* tab, and remove both checkmarks. Next, apply protection to each of **the other** worksheets in the workbook file by selecting from the *Excel* main menu bar: *Tools* → *Protection* → *Protect Sheet* and optionally entering a password. This protects the secondary calculated data from unintentional changes.
- **Relative and Absolute Reference Cells.** Press F4 repeatedly while the mouse pointer is in front of a cell address (You obtain the sequence: A1, \$A\$1, A\$1, \$A1).
- **Equation Editor.** Use to describe the formula in the analytical method. *Insert* → *Object...* → *Microsoft Equation 3.0*.
- **Statistics.** Use the following *Excel* formulas when necessary (use *Excel 2002* Office Assistant for *Excel 2002* formula definitions and descriptions):

✓ **Equations for calculating trendlines**

The equation for the straight line: $y = mx + b$
where:

m is the slope

b is the intercept

Once the slope and intercept are known, if one of the dependent variables is known, a value for the other dependent variable x' or y' can be calculated through the formulas:

$$x' = \frac{(y - b)}{m} \qquad y' = mx + b$$

Example

Use: $=(\text{Sample Response}-\text{INTERCEPT})/\text{SLOPE}$.

This formula gives the sample concentration as found by regression.

✓ **Equation for calculating the intercept**

Use: $=\text{INTERCEPT}(\text{Response range of cells}, \text{Concentration range of cells})$. For example, $\text{INTERCEPT}(A1:A6, B1:B6)$.

✓ **Equation for calculating the slope**

Use: =SLOPE(Response range of cells, Concentration range of cells). For example, SLOPE(A1:A6, B1:B6).

✓ **Equation for calculating the correlation coefficient**

Use: =CORREL(Concentration range of cells, Response range of cells). This is the correlation coefficient; for example CORREL(A1:A6, B1:B6).

✓ **Equation for calculating the average**

Use: =AVERAGE(Range of cells). For example, AVERAGE(A1:A6).

✓ **Equation for calculating the standard deviation**

Use: =STDEV(Range of cells).

For example, STDEV(A1:A6).

For the % RSD: =(STDEV/AVERAGE)*100.

Note: All of the above *Excel* equations conform to the official compendia formula descriptions (AOAC, USP, etc.).

Multiple-User Application Examples

Figures 1, 2, and 3 give examples of the multi-users spreadsheet analytical application. Figure 1 shows instructions for the user. When one of the buttons (“Instruction for the user” or Question about this application”) is pressed, a message is displayed showing instructions to aid the analyst. Figure 2 shows the secondary data (calculation and results) worksheet. The raw data sheet is not shown. A cell comment is presented, prompting the analyst to weigh the correct amount of the sample. This comment is only viewed when the mouse pointer moves over the cell that contains the comment. Figure 3 shows the standard worksheet with secondary data. The raw data comes from the instrument print-out and the standard concentrations come from the certificate of analysis results from Matreya, Inc. Cell comments are presented. Figure 4 shows the validation worksheet form for the multi-user application which is completed after the application has been validated. This form is included with the rest of the validation documentation (Figures 5A to 13).

Components of Validation Documentation

- **Validation plan** (10). The validation plan is a strategic document that should state what is to be done, the scope of approach, the schedule of validation activities, and

the tasks to be performed. The plan should also state who is responsible for performing each activity. The plan should be reviewed and approved by designated management. Some suggested components of the validation plan are the following: Purpose and scope of plan; background information; references; system overview and process description; vendor selection and management (when applicable); development/implementation methodology; configuration management for the code, system, and documentation; validation scope; assumptions, exclusions, and limitations; validation approach; high-level test plans and system acceptance criteria; roles and responsibilities; validation schedule; overview of computing environments; training and implementation strategy; system maintenance and support strategy; re-qualification criteria; and documentation maintenance.

In addition, the validation plan needs to be written as early in the project as possible to avoid additional validation costs involved by writing documentation retrospectively that should have been written at the time the activity occurred.

- **Validation procedure (10).** The validation procedure should include detailed steps on how to conduct the validation. It should describe the computer system configuration, as well as test methods and objective acceptance criteria, including expected outcomes. The procedures should be reviewed and approved by designated management. The validation procedure should be divided into an Installation Qualification (IQ) and Performance Qualification (PQ).
- ✓ **Installation Qualification.** The test cases in the IQ will establish confidence that the computer hardware and spreadsheet software is in compliance with the manufacturer and the specifications. The following test cases in the IQ for the spreadsheet applications software, workbook files and formulas include (4):
 - System documentation.
 - Computer system configuration verification.
 - List of communication cables.
 - Verification of electrical utility.
 - Hardware maintenance contracts/procedures.
 - Hardware change control.
 - List of critical equipment and spare parts.
 - Software review.
 - Software change control.
 - Standard operating procedure review.
 - Personnel training requirements and records
 - System security-physical access.
 - System security-on line program access.
 - Software formulas.

- Printer generated data.
 - Programmable functional keys (PFKs) and Displays verification.
 - Environmental conditions.
- ✓ **Performance Qualification.** PQ will thoroughly, rigorously and consistently challenge the software by testing all input/output values, data structure, process variables, and control flow logic. The following test cases in the PQ on spreadsheet applications include (4):
- On-Line Program Access Verification.
 - Field specification verification.
 - Power failure checkout.
 - PFKs verification.
 - Communication failures.
 - Electromagnetic interference (EMI) / radio frequency interference (RFI) verification.
 - Field specification verification.
 - Printout reports.
 - Performance testing.
- **Validation report** (10). The validation report should document detailed results of the validation effort, including test results. Whenever possible, test results should be expressed in quantified terms rather than stated as “pass/fail”. The report should be reviewed and approved by designated management.

Validation Procedure (Refer to Figures 1, 2, and 3).

The following guidelines can be used to test a spreadsheet that others will use:

- **Develop the requirements (in writing).** This includes the definition of use, definition of parameters (range of values, formulas, errors, messages, etc., from the analytical method) needed to design the spreadsheet.
- **Use dynamic testing** (10). Some testing considerations:
 - ✓ Test conditions should include not only “normal” or “expected” values, but also stress conditions (such as a high number of users accessing a network at the same time). Test conditions should extend to boundary values, unexpected data entries, error conditions, reasonableness challenges (e.g., empty fields, and date outliers), branches, data flow, and combinations of inputs. If the worksheet is set up to perform calculations using input cells, spend some time and enter very large or very small numbers and observe the effect on the formulas. If the user should enter a percentage, see what happens if a value over 100 is entered. If a positive number is expected, try entering a negative number. This also is a good way to ensure that columns are wide enough.

- ✓ Simulation tests may be performed using simulators, usually conducted off-line outside of the actual user's computing environment. These types of tests are conducted when appropriate.
- ✓ Live, user-site tests are performed in the end user's computing environment under actual operating conditions. Testing should cover continuous operations for a sufficient time to allow the system to encounter a wide spectrum of conditions and events in an effort to detect any faults that are not apparent during normal activities.
- **Use dummy data.** If a linear regression or other statistical procedures are used, try entering a "1" into each non-formula cell. This is a good way to make sure that all statistical formulas refer to the correct ranges. An incorrect formula usually stands out in this procedure.
- **Get familiar with the spreadsheet program auditing tools.** In the workbook-building process, use the auditing tool because worksheet formula errors and faulty logic can be introduced at any stage of the development process. Use the *Tool* → *Auditing* → *Show Auditing Toolbar*. See the *Excel* textbook for more information.
- **Be sure software testing includes (10):**
 - ✓ Structural testing that takes into account the internal mechanism (structure) of a system or component. It is sometimes referred to as "white box" testing. Structural testing should show that the software creator followed contemporary quality standards. This testing usually includes inspection (or walk-through) of the program code and development documents. For example, Excel source codes are written on Visual Basic for Application (VBA) and one of the development documents is the analytical method used to develop the application software.
 - ✓ Functional testing that involves running the program under known conditions with defined inputs, and documented outcomes that can be compared to pre-defined expectations. Functional testing is sometimes called "black box" testing.
 - ✓ Program build testing that is performed on units of code (modules), integrated units of code, and the program as a whole.
- **Conduct performance testing.** This verifies the spreadsheet application and the embedded formulas. It ensures that the formulas and configuration are correct and that the numbers are handled correctly. Manual calculations should be performed by a user and compared against the spreadsheet output. Manual calculations may be included in the documentation until three successful runs are made using the spreadsheet application.
- **Provide error messages.** The messages can inform the user when values are entered incorrectly or when the analyst uses the worksheet incorrectly.
- **Provide instructions for users.** This should include a step-by-step procedure so that an inexperienced user can use the worksheet correctly without assistance. Use several cells to accommodate the instructions or use the drawing tools and insert a

text box. Alternatively, use the *Insert* → *Comment* to add additional instructions for the users.

- **Add validation to data-entry cells.** Use validation criteria for a particular cell to ensure that data-entry is of the proper type. Restrict data to whole numbers, decimal numbers, or text or set limits on entries. Also, a list of acceptable entries can be specified (this is described in the safety procedure above). For example, to restrict data to decimal numbers use the sequence: *Data* → *Validation* → Choose “*Setting*” Tab → in “*Allow*” choose: “*Decimal*” → in “*Data*” → Choose “*between*” → in “*Minimum*” enter 0 and in “*Maximum*” enter 100. If the value does not fall between 0 and 100, an error message will be activated. For setting up the message, select *Data* → *Validation* → choose: *Input Message* → enter short instructions in the window *Title*. Optionally, choose: *Error Alert* and write the correction instructions in the window: *Error Message*.

Life Cycle of the Spreadsheet Application

When the applications are validated they should be preserved, controlled and maintained. These controls are fundamental to the operation of the equipment and process in a regulatory environment. Controls to support systems for software and computerized systems include (4):

- **Preventive maintenance.** Procedures should be in place to ensure that the preventive maintenance occurs throughout the life of the computer hardware.
- **Environmental control.** Adverse environmental conditions may affect the operation of the system. It is necessary to document the key environmental parameters and operational limits of the system components. Show that the system continues to reside in a location that can be maintained within the specified operational limits. A procedure should be in place for the ongoing monitoring and documentation of key environmental parameters. Environmental parameters may include: temperature, relative humidity, power levels and conditioning, static electricity, dust conditions and electromagnetic and radiofrequency interference.
- **Performance monitoring.** Testing must include a periodic review using input data and conditions that represent the range of actual operating situations. Inputs should be used for which the correct output is known.
- **Backup.** Procedures should be in place to ensure that the backups are performed on regular basis and stored in a secure location.
- **Recovery.** A system recovery procedure should be in place. This system should completely restore the system to its previously validated state. This should also include the number, frequency and location of backups based on the need for any data regeneration.

- **Training.** A procedure should be in place to ensure adequate training of the formula developers, users and administrators. Training records should be maintained throughout the system life.

Spreadsheet Verification and Validation: a Comparison

The agency defines process validation as “It establishes documented evidence which provides a high degree of assurance that a specific process will consistently produce a product meeting its pre-determined specification and quality characteristics. The final validation review assures that the developed product meets all its predefined specifications and that the documentation is complete” (12). This definition was modified by the Parenteral Drug Association (PDA) for a computerized system as “establishing documented evidence which provides a high degree of assurance that a specific computer-related system will consistently produce a product meeting its predetermined specifications”(11).

Verification is defined as “The process of evaluating the spreadsheet application for consistency and correctness of the software at each stage and between each stage of the development life cycle, to ensure compliance with the analytical method. Verification activities are in-process activities (testing and measurement), performed concurrently with spreadsheet (workbook file) development”.

It is important to mention that another definition for software validation is stated in the “Guidance for Industry of Computerized Systems Used In Clinical Trials” (24) and “FDA Draft Guidance for Industry 21 CFR Part 11; Electronic Records; Electronic Signatures Validation, August 2001” (10), which stated "Confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through the software can be consistently fulfilled."

In general, validation is concerned with generating evidence to demonstrate that the system is fit for the purpose for which you use it, it continues to be so when it is operational, and there is sufficient evidence of management control. This means that an action must be documented. Another feature of validation is to produce an auditable system with the appropriate documentation to aid any audit or inspection. Verification is concerned by the demonstration of consistency, completeness, and correctness of the software at each stage and between each stage of the development life cycle.

Validation Documentation

- Document the process using the worksheet validation form (Figure 4).
- Document any formulas, macros, and user interface elements in the spreadsheets and include copies in the validation documentation.

- Provide a “source code” copy. For *Excel* the source codes are written in VBA.
- Document data-entry validation ranges. Show which cells had data-entry restriction and what error message(s) will be activated if the value is out of range. Test the workbook sheet showing different input values.
- Use the Workbook Audit report (*Power Utility Pak v5.0*) to audit each worksheet in the workbook file and add the report to the validation documentation (12). The PUP 2000 auditing tools provides the following utility tools which are created in a new workbook automatically:
 - ✓ *Workbook Summary Report* (Figures 5A and 5B): This utility includes the *Workbook summary* (Summary information about the workbook), the *Name report* (A listing of all names defined in the workbook), the *Worksheet report* (Information about each worksheet in the workbook), and the *VBA module report* (A listing of the procedures contained in each VB Component).
 - ✓ *Workbook Link Finder* (Figure 6) is a detailed listing of all links contained in the workbook file (ideal for tracking down “phantom” links). The utility also has an option to create a report of all links. This report includes: the *Linked files* (The items that appear when Links are selected from the Edit menu); the *Formula links* (Formulas that refer to cells in a different workbook); the *Name links* (Names that are defined using a different workbook); the *XL5/95 dialog box links* (Link in the controls contained on dialog sheets); the *Chart links* (Links contained in a chart's SERIES formula or in shapes contained in the chart); *Object links* (Links contained in drawing objects or shapes); and, the *OLE links* (Links from embedded objects).
 - ✓ *Worksheet Map* (Figure 7) creates a color-coded map that summarizes the contents of a worksheet. This map can help you spot errors in your sheet -- for example, a non-formula among a group of formulas. This map displays cell entries graphically and shows the position in the worksheet of formulas, values, texts, logic and error.
 - ✓ *Compare Ranges* performs a cell-by-cell comparison of two ranges or two worksheets. It generates a report of the differences found. You can select among three options that determine if two cells are different.
 - ✓ *Date Report* (Figure 8) creates a useful report that describes all date cells in a worksheet or workbook. This report may help identify potential year-2000 problems.
 - ✓ *Name Lister* (Figure 9) allows viewing of all names in a workbook, sorted by the following categories: the *All names* (Every name defined in the active workbook), the *Workbook-level names* (Names defined at the workbook level) and the *Sheet-level names* (Names defined at the worksheet level). These names are preceded by the worksheet's name and an exclamation point (e.g., Sheet1!MyName), the *Hidden names* (Names that are hidden and do not appear in Excel's Define Name

dialog box), the *Linked names* (Names that are defined in terms of a range on a different workbook) and the *Bad names* (Names that contain an erroneous reference). These names often cause a "phantom link" problem -- Excel prompts the user to update links when the workbook is opened.

- ✓ *Formula Report* (Figure 10) creates a listing of all formulas in a worksheet or workbook, including values, and locked /unlocked status for each worksheet.
- ✓ *VBA Project Summary Report* (Figure 11 and 12) generates a listing of all VBA procedures contained in a workbook. Optionally, the report includes a list of all controls contained in all UserForms.
- ✓ *VBA macro code* (Figure 13) prints the VBA macro codes and can be performed by PUP 2000 or by Excel it self using the VBA project explorer.
- Identify the server drive location of the spreadsheet file. A server location is preferred over a local hard drive, because it provides wider access by users and ensures that the backup of the program is automated.
- Demonstrate to the supervisor, reviewers and users that the spreadsheet performs correctly and obtain supervisor and director signatures to release the validated spreadsheet for general use.
- Include copies of the formulas used from the analytical procedure and check them against the formulas used in *Excel*. Software formulas must be identified, reviewed and accepted by QA. The list of all algorithms or formulas used in the spreadsheet should be on file and documented.
- Provide copies of test results and procedures, and of error messages that may result upon testing. Final specification / requirements document.
- Correctly paginate (for example, page X of Y). Validation documentation is a controlled document, be sure it is signed by author, authorized by two other people (technical and supervisory), and distributed to specified individuals.
- Provide on a 3.5-inch floppy a copy of the final spreadsheet application, the validation results of the raw data set, a copy of the worksheet validation form (Figure 4), and a copy of the user's instruction to the supervisor or person designated by management to maintain validation documentation.
- Create the worksheet validation form on an *Excel* workbook file (Excel Validation Forms.xls) as a fill-in-the-blank form; the file can be send via e-mail upon request.

Operational Change Control (15)

During their operational use changes to spreadsheet applications must be controlled. This change control must be implemented from the point in the development life cycle when the user starts to use the application, and will continue thereafter through the life of the application. These changes include the controlled documentation, application software, operating system software, firmware, hardware and system, master and configuration data for system in use, or during qualification and acceptance in the user environment. All changes must be reviewed, impact and risk assessed, authorized, documented, tested, and approved before implementation. Any

proposed change should be requested by completing the request for change section of the change request form (Figure 14). Each change request should be assigned a unique reference number and logged in the change request index (Figure 15). If more than one controlled item is to be changed as a result of the change request then each change should be tracked to completion. The use of a change note form (Figure 16) for each item should be considered, defining the particular changes to be made to that item. A change plan may be required for multiple activities and resources and, if produced, should be attached to the change request and cross-referenced from it. When all changes have been implemented and re-testing is completed, the change request should be passed to the supervisor for final review and approval. When the change request form is completed it should be filed and the change request index (Figure 15) updated.

Managing Workbook File Applications

Each laboratory section should maintain a listing of all validated spreadsheets, their revision numbers, and corresponding analytical procedures. When the spreadsheet application is approved, an electronic version of the spreadsheet file must be placed in a dedicated directory on a network drive. This directory will be protected, with read-only access designated for end-users of the application file. Access to the network directory where the spreadsheet application is installed should be granted only to those laboratory sections that will use the application.

A separate, dedicated network directory shall be established for saving filled-in or used spreadsheets. After completing a given set of data analyses, the analyst must save the filled-in spreadsheet in this directory. The analyst may not re-use any filled-in spreadsheets for subsequent analyses. All new data sets must be calculated using the write-protected, blank spreadsheet template from the network directory. All filled-in spreadsheet applications using the old spreadsheet version must also immediately be removed and archived in a protected location.

Conclusions

Currently, spreadsheets for analytical applications are widely used in the FDA analytical laboratories and are starting to emerge in other areas such as the microbiology laboratories. For these reasons, applications need to be standardized within all FDA laboratories through the use of standard operation procedures (SOP's) to minimize design and validation discrepancies among the applications. Basic principles of software engineering techniques must be applied in the construction of spreadsheet applications. If spreadsheet applications are used without proper control (SOPs, training), the operation may go out of control creating serious errors among the applications. Also it is important to use the operational change control (15) forms, which maintain the analytical applications in a validated stage.

This document is based on a system that consists of both, electronic and paper records. For example, when a workbook file is completed and printed by the original analyst and verified by the second analyst, the verified printed sample worksheets are used for the regulatory documentation and not the electronic workbook file, which is saved and maintained in the

laboratory where the sample analysis was performed. This saved workbook file can be used for future audits against the completed printed worksheets.

Each laboratory must train several analysts to design and validate spreadsheet analytical applications. These trained analysts can, in turn, train other analysts. The advantage of training analysts in creating spreadsheet applications is definitely positive for the FDA as was demonstrated in the Atlanta Center for Nutrient Analysis (ACNA) in which all analysts are using the *single-user* spreadsheets (discussed in Part II) as well as the multi-user applications. All analysts cooperate on improving the *multi-user* applications as well as the *single-user spreadsheets*. Therefore, the standardization of ACNA analytical spreadsheets improved reports and saved considerable analyst time. The authors currently maintain and up-grade the multi-user applications by improving the spreadsheet applications (method changes, errors, method reference changes and improvement of the instructions for the users as requested by the analysts) and re-validations.

Use of the Power Utility Pak, Version 5.0 or equivalent is recommended, which is a unique collection of add-ins for Microsoft Excel for Windows. Add-ins are designed to save time, ease the use of functions and generate validation documentation. It is important to mention that the degree of validation depends on the complexity of the spreadsheet analytical application. When a LIMS approach is used to manage the raw data (data from balances and analytical instrumentation), the raw data can be sent to the existing spreadsheet applications. The spreadsheet applications should be tested and validated with its documentation before and after the raw data from the LIMS is sent to the applications. Where application files are locally installed or on a network by means of a LIMS, it is important that the LIMS and the spreadsheet applications are secured by implementation of the safeguards in 21 CFR 11 (electronic records and electronic signature regulations).

Management commitment is needed in order to implement policies and SOPs necessary for future laboratory accreditation. Implementing controls over the design and development of analytical spreadsheet applications serves to protect the integrity of the data and offers some assurance regarding the usefulness of the information of the applications generated. Currently, spreadsheet development in FDA laboratories is not standardized by means of SOPs, and training is not normally given to reduce errors.

Finally, it was found that spreadsheet error rates (13) are completely consistent with error rates found in other human activities (speech, typing, programming, driving, industrial accidents, even commercial aircraft disasters). Overconfidence tends to blind people to the need for taking steps to reduce risk. In “spreadsheets” formal testing and adherence to tedious disciplines (such as training and SOPs) can save time and avoid errors. The ability to catch errors installs confidence.

References:

1. Title 21 Code of Federal Regulations, Part 211, "Current Good Manufacturing Practice for Finished Pharmaceuticals", Revised as of April 1, 2000; pages 115-135. <http://www.fda.gov>
2. Title 21 Code of Federal Regulations, Part 58, "Good Laboratory Practice for Non-clinical Laboratory Studies," Revised as of April 1, 2000; pages 297-311. <http://www.fda.gov>.
3. Ludwig Huber: Validation of Computerized Analytical Systems, Interpharm Press, Inc., Buffalo Grove, IL, ISBN: 0-935184-75-9, 1995.
4. Lopez, Orlando, Journal of Validation Technology, Computer Validation Doctor: Spreadsheet Qualification Applications, Vol. 3, No. 3. 1997, pages 260-266.
5. AOAC International (2000), ISO 17025 and the Laboratory – An Introduction to Laboratory Accreditation, Gaithersburg, MD. <http://www.aoac.org>
6. 21 CFR Part 11, "Electronic Records; Electronic Signatures, Final Rule," Federal Register 62 (1997) 13430-13466, World Wide Web http://www.fda.gov/ora/compliance_ref/Part11/.
7. Rajalingham, Chadwick, Knight, and Eduards, Proceedings of the 33rd Hawaii International Conference on System Sciences-2000, Quality Control in Spreadsheets: A software Engineering-Based Approach to Spreadsheet Development. Available at the internet address: <http://panko.cba.hawaii.edu/ssr/HICSS33/HICSS33-Rajal-SoftEngr.pdf>
8. Institute of Chartered Accountants in England and Wales (1994), Briefing No. 6
9. Walkenbach, John (1996), Excel 97 Bible, IDG Books Worldwide, Inc.
10. FDA Draft Guidance for Industry 21 CFR Part 11; Electronic Records; Electronic Signatures Validation, August 2001. Internet address: http://www.fda.gov/ora/compliance_ref/part11.htm.
11. Parenteral Drug Association Technical Report 18, 1995, Validation of Computer-Related System.
12. JWalk & Associates: Auditing report (Power Utility Pak 97 or 2000), <http://www.jwalk.com>, P.O. Box 12861, La Jolla, CA 92039-2861, (1997).
13. Panko R. R., Journal of End User Computing's, Vol. 10, No. 2. 1998, pp. 15-21. (www.cba.hawaii.edu/panko).
14. FDA: Glossary of Computerized System and Software Development Terminology, The Division of Field Investigations, Office of Regional Operations, Office of Regulatory Affairs, (1995).
15. GAMP 4 Good Automated Manufacturing Practice (Guide for Validation of Automated Systems), 2001
16. Spreadsheet Detective: (for auditing spreadsheets). Southern Cross Software , Queensland. <http://www.uq.net.au/detective>
17. Cantellops, Dennis (1997), LIB No. 4119
18. Cantellops, Dennis (1994), LIB No. 3946
19. Diamond, D.; Hanratty, and Venita C. A. (1997), Spreadsheet Applications in Chemistry Using Microsoft Excel, John Wiley & Sons, NY.
20. Compliance Policy Guide 7153.17 for enforcement of 21 CFR 11: Electronic Records and Electronic Signatures," Federal Register 64 (1999) 41442-41443.

21. D'Eramo, Paul N.; Budihandojo Rory (1999), *Pharmaceutical Technology*, 23, No. 3, 78-88.
22. John R. Joyce: *Computer / Software Trends*. "jrjoyce@briefcase.com", *Inside Laboratory Management*, AOAC International, August, 1999.
23. Information about ISO17025: <http://ts.nist.gov/ts/htdocs/230/235/qualtemp.htm>
24. Information about ISO in general: <http://www.isogroup.iserv.net/basic10.htm>
25. Computerized Systems Used In Clinical Trials
(http://www.fda.gov/ora/compliance_ref/bimo/ffinalcct.htm)
26. R.D. Mcdowall, February 2001. *Validation of Spectrometry Software, Part I, Spectroscopy Magazine*.

Additional Information:

Panko, R. R. (1997b). *Spreadsheet Research (SSR)* is a repository for research on spreadsheet development, testing, use, and technology. Website: (<http://www.cba.hawaii.edu/panko/ssr/>). Honolulu, Hawaii: University of Hawaii.

Panko, R. R. (1997a). *Human Error Website*:

(<http://www.cba.hawaii.edu/panko/papers/ss/humanerr.htm>) . Honolulu, HI: University of Hawaii.

Acknowledgments

The authors wishes to thank Ronald R. Eiternmiller, Ph.D., Basil Vassos Ph.D., Luis Burgos, Chemist, Cecilia Romero, Chemist and ACNA chemists for useful comments.

Special appreciation is given to Paul J. Motise, CSO for the review of this document and clarifying some 21 CFR Part 11 issues, which have been included in the body of this paper.

Special appreciation is given to David Irizarry, Computer Specialist, for clarifying some of the computer related security issues.

Fatty Acids Methyl Ester (FAME's) Spreadsheet Reporting v5.0

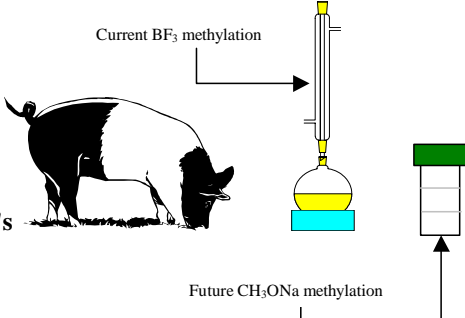
Instructions for the user Show Toolbar Questions about this application

>> Sample and results page (total fat, saturated fat, polyunsaturated fat, and monounsaturated fat)

>> Standard page

>> Instrument parameters (GC)

>> Equations used to calculate FAME's



Current BF_3 methylation

Future CH_3ONa methylation

Figure 1. Menu and instructions worksheet of a multi-user spreadsheet application used in Atlanta Center for Nutrient Analysis (ACNA) for the determination of fatty acids (FAME's) in food.

Clear data-entry cells on current page

Show Toolbar

Show Menu

Product: SRM1846 (Infant Formula Powder)				Sample No.: SRM 1846			
Serving Size: <u>100 g</u>				Analyst: DC			
				Date: 02/08/99			
Method: AOAC 16th ed. 32.2.02A (Methylation / Quantitation)							
Fat Extraction: AOAC 16th ed. 32.2.02A, Fat in Cereals							
Net Wt: <u>1.2963 g</u>		Enter the sample weight in gram. For SRM1846 weigh approx. 1.3 g (this is equivalent to approx. 350 mg of fat for BF ₃ methylation). For regular samples weight equivalent to approx. 350 mg of fat for BF ₃ methylation.				(C13:0): <u>5.004 mg/mL</u> Standard: <u>1.00 mL</u> Standard: <u>27369.60</u>	
Fatty Acid	Peak Area (Asi)	Res. Factor Ri	FAME wt. (g) Fi	Conv. Factor FAME → TG	Conv. Factor FAME → FA	Triglyceride wt (g) F (i, TG)	Fatty Acid wt (g) F (i,FA)
4:0				0.9868	0.8627		
6:0				0.9897	0.8923		
8:0	26160.90	0.8776	0.00545	0.9915	0.9114	0.00540	0.00497
10:0	27098.10	0.9397	0.00527	0.9928	0.9247	0.00523	0.00488
12:0	251269.00	0.9867	0.04656	0.9937	0.9346	0.04627	0.04351
13:0				0.9941	0.9386		
14:0	117172.00	1.0132	0.02114	0.9945	0.9421	0.02103	0.01992
14:1				0.9944	0.9417		
15:0	774.60	1.0302	0.00014	0.9948	0.9453	0.00014	0.00013
15:1				0.9947	0.9449		
16:0	230872.00	1.0607	0.03979	0.9950	0.9481	0.03960	0.03773
16:1 cis	1361.89	1.0489	0.00024	0.9950	0.9478	0.00024	0.00022
17:0	1831.65	1.0499	0.00032	0.9953	0.9507	0.00032	0.00030
17:1				0.9952	0.9503		
18:0	235333.00	1.0876	0.03956	0.9955	0.9530	0.03938	0.03770
18:1 trans	13063.03	1.0638	0.00225	0.9955	0.9527	0.00223	0.00214
18:1 cis	738911.00	1.0854	0.12447	0.9955	0.9527	0.12391	0.11858
18:2	371516.22	1.0677	0.06362	0.9954	0.9524	0.06332	0.06059
18:3	7199.01	1.1358	0.00116	0.9954	0.9520	0.00115	0.00110
18:4				0.9954	0.9517		
20:0				0.9959	0.9570		
20:1	4860.69	1.1058	0.00080	0.9959	0.9568	0.00080	0.00077
20:5				0.9958	0.9557		
20:6				0.9957	0.9554		
22:0	4680.04	1.1217	0.00076	0.9962	0.9604	0.00076	0.00073
22:1				0.9962	0.9602		
22:5				0.9961	0.9593		
22:6				0.9961	0.9591		
Total Fat (as triglycerides):						27 %	N/A
Saturated Fat (as fatty acids):						N/A	11.6 %
Polyunsat. Fat (as fatty acids):						N/A	4.76 %
Monounsatur. Fat (as fatty acids):						N/A	9.22 %
Total Fat / Serving		<u>27 g</u>					
Saturated Fat / Serving		<u>11.6 g</u>					
Polyunsatur. Fat / Serving		<u>4.76 g</u>					
Monounsatur. Fat / Serving		<u>9.22 g</u>					

(Modification date: 12/14/99 / v5.0 Revised)

Figure 2. Calculation and results worksheet (secondary data). The raw data sheet is not shown.

Clear data-entry cells on current page

Show Toolbar

Show Menu

This is the concentration of the internal standard (C13:0) in mg/mL of the standard mix (C_{S13}).

Sample No.: **SRM 1846**
 Analyst: DC
 Date: 02/08/99

Fatty Acids	Peak Area (Ai)	FAME Conc. (mg/mL) Fsi	Response Factor Ri
4:0			
6:0			
8:0	16721.90	0.3000	0.8776
10:0	29841.80	0.5000	0.9397
12:0	62666.30	1.0000	0.9867
13:0		0.5000	
14:0	32175.50	0.5000	1.0132
14:1	19334.00	0.3000	1.0147
15:0	19628.20	0.3000	1.0302
15:1			
16:0	134731.00	2.0000	1.0607
16:1 cis	66615.20	1.0000	1.0489
17:0	33340.60	0.5000	1.0499
17:1			
18:0	69074.50	1.0000	1.0876
18:1 trans	27024.70	0.4000	1.0638
18:1 cis	206805.00	3.0000	1.0854
18:2	135629.00	2.0000	1.0677
18:3	72138.10	1.0000	1.1358
18:4			
20:0	9852.70	0.3000	0.5171
20:1	21069.60	0.3000	1.1058
20:5			
20:6			
22:0	21372.70	0.3000	1.1217
22:1	21246.70	0.3000	1.1151
22:5			
22:6			

Fatty Acids Mixed Standard

Conc. Internal Std. (C13:0): 0.5000 mg/mL

Area Internal Std. (C13:0): 31756.00

Enter the area of the internal standard (C13:0) in the standard mix (A₁₃).

Notes:

- Standard mix used: Kel-Fim-Fame-5, Total FAME's 15.5 mg/mL, catalog No. 4210, Matreya, Inc., Pleasant Gap, PA.
- Excel 97 v8.0 computerized spreadsheet program, was used to developed this multi-user application.

(Modification date: 12/14/99 / v5.0 Revised)

Figure 3. Standard worksheet with secondary data.

Spreadsheet / Worksheet Validation Form for Multi-User Application					
<p>1. Name and version of the operation system (Windows 95, 98, etc.) and the spreadsheet program used to develop the spreadsheet / worksheet (workbook file) application.</p> <p>The operation system used is Windows 95 and the application was developed with Microsoft Excel 97 v8.0</p>					
<p>2. Author (s) / Developer (s) of the spreadsheet / worksheet (name, signature and date).</p> <p>Dennis Cantellops, Anne P. Reid and Thomas Hamill</p>					
<p>3. Name of the workbook file (version number and the date) and its intended purpose. Cite official methods as appropriate.</p> <p>Method (s): AOAC 16th ed. 32.2.02A (Methylation / Quantitation)</p> <p>"FAMEs Calc.xlt" v5.0, for the calculations of Fatty Acid Methyl Esters analyses (Attach. A1, A2, & A3).</p>					
<p>4. Identify & define the formulas that the application uses. Are the formulas the same as cited in the official method?</p> <p>See Attach. C, D, & E, and the formulas are the same as the official method.</p> <p>Audit report on Attachments, B1-B3 (Menu), C1-C6 (calc. & results), & D1-D4 (response factor).</p>					
<p>5. Program interacts with the following other systems.</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">System Name</th> <th style="text-align: left;">Type of Interaction</th> </tr> </thead> <tbody> <tr> <td>N/A</td> <td>N/A</td> </tr> </tbody> </table>		System Name	Type of Interaction	N/A	N/A
System Name	Type of Interaction				
N/A	N/A				
<p>6. User instructions for this application include the following:</p> <table style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td style="width: 50%;">(Instructions for normal use)</td> <td style="width: 50%;">(Problem and help instructions)</td> </tr> <tr> <td>Menu and instructions for normal used. Attach. A1</td> <td>Dialog box with letter help.</td> </tr> </tbody> </table>		(Instructions for normal use)	(Problem and help instructions)	Menu and instructions for normal used. Attach. A1	Dialog box with letter help.
(Instructions for normal use)	(Problem and help instructions)				
Menu and instructions for normal used. Attach. A1	Dialog box with letter help.				
<p>7. Spreadsheet / Worksheet security (cells and visual basic project properties protection) for inadvertent changes.</p> <p>(Protect worksheet for Contents, Objects, Scenarios) (VBA Project Properties - Lock project for viewing)</p> <p>Provide password (s): "xxxxxxx" for both each worksheet and VBA project properties (see Director below).</p> <p>Each worksheet protected for contents, objects, scenarios and VBA Project Properties - Locked for viewing</p>					
<p>8. Workbook file (worksheets) has been tested for normal and problem cases and all formulas were verified by manual calculations by (signature and date).</p> <p>Thomas Hamill</p>					
<p>9. Spreadsheet / worksheet released for validated use by (Supervisor's / Director's signatures and dates).</p> <p>Anne P. Reid</p>					
<p>10. Server location / Individual PC's of the spreadsheet / worksheet application.</p> <p>Installed on drive Q, Folder: ACNA Spreadsheet Applications, Folder: FAME's and select > FAME's Calc.xls</p>					
<p>11. Document data-entry validation ranges. Shows which cells had data-entry restriction and what error message will be activated if the value is out of range.</p> <p>See Attachment F for testing out-put (Screen print out showing the results for the testing).</p>					
<p>12. Validation documentation (3.5 inch floppy copy of the final workbook file, the validation results of the raw data set, a copy of spreadsheet validation form and a copy of the user's instructions) are provided to the supervisor.</p> <p>Author's signature and date: Dennis Cantellops, 00/00/00</p> <p>Supervisor signature and date: Anne P. Reid, 00/00/00</p>					

Figure 4. Validation worksheet form for multi-user application (FAMEs Calc.xlt).

Workbook Summary Report for FAME's Calc.xls*Report generated: Monday, April 09, 2001 11:49:12 AM***File Information**

Workbook name: FAME's Calc.xls
 Workbook path: C:\~\ACNA Spreadsheet Applications
 Author: Dennis Cantellops
 File format: Excel 97/2000 Workbook
 File creation date: Wednesday, January 24, 1996 2:01:02 PM
 File size: 470,528 bytes
 Read-only: No

Workbook Protection

Structure protected: No
 Windows protected: No

Sheet Information

Total sheets: 6
 Hidden sheets: 0
 Worksheets: 6
 Chart sheets: 0
 Excel 5/95 Dialog sheets: 0
 Excel 4 XLM Macro sheets: 0

Workbook Settings

Date system: 1900 Date System
 Precision as displayed: No
 Accept labels in formulas: Yes

Linked Files

(no linked files)

Name Report for FAME's Calc.xls*Report generated: Monday, April 09, 2001 11:49:12 AM*

Name	Refers To	Type	Hidden
ClearSp1	=Fat1!\$D\$3,Fat1!\$D\$4,Fat1!\$D\$9,Fat1!\$I\$3,Fat1!\$I\$4 ,Fat1!\$I\$5,Fat1!\$J\$8:\$J\$10,Fat1!\$D\$15:\$D\$19,Fat1!\$ D\$21:\$D\$42,Fat1!\$C\$6:\$H\$7	Workbook Level	No
ClearStd1	=Fat2!\$I\$12,Fat2!\$D\$10:\$D\$14,Fat2!\$D\$16:\$D\$37	Workbook Level	No
Fat1!PrintArea	=Fat1!\$B\$2:\$L\$52	Sheet Level	No
Fat2!PrintArea	=Fat2!\$B\$2:\$K\$57	Sheet Level	No
Menu!PrintArea	=C\$4:\$G\$17	Sheet Level	No
Fat1!Print_Area	=Fat1!\$B\$2:\$L\$53	Sheet Level	No
Fat2!Print_Area	=Fat2!\$B\$2:\$K\$58	Sheet Level	No
Formulas_1!Print_Area	=Formulas_1!\$E\$63:\$G\$77	Sheet Level	No
Inst_1!Print_Area	=Inst_1!\$B\$2:\$K\$51	Sheet Level	No
Menu!Print_Area	=Menu!\$E\$63:\$G\$77	Sheet Level	No

Figure 5A. Summary Audit Report for worksheets on Figure 1, 2 and 3. This report is generated with the Power Utility Pak v5.0. If PUP v5.0 is installed in *Excel 2002*, Select *PUP v5* from the main menu: → *Auditing Tools*. From Figures 5A to 12 are the auditing reports for the worksheet in Figures 1, 2 and 3, showing the calculations and results for FAME's.

Worksheet Report for FAME's Calc.xls*Report generated: Monday, April 09, 2001 11:49:12 AM*

Sheet Name	Index	Visible	Contents Protected	Objects Protected	Scenarios Protected	Used Range	Used Rows	Used Columns	Cells in Used Range	Hidden Rows	Hidden Columns
Menu	1	Yes	No	No	No	A1:H77	77	8	616	0	0
Fat1	2	Yes	No	No	No	A1:IU2205	2205	255	562275	0	0
Fat2	3	Yes	No	No	No	A1:BF226	226	58	13108	0	0
Inst_1	4	Yes	No	No	No	A1:BT195	195	72	14040	0	0
Formulas_1	5	Yes	No	No	No	A1:AD197	197	30	5910	0	0
HelpSheet	6	Yes	No	No	No	A1:M98	98	13	1274	0	0

Sheet Name	Formula Cells	Constant Cells	Empty Cells	Chart Objects	Shapes	Comments	Scenarios	Pivot Tables	Query Tables	1-2-3 Eval	1-2-3 Entry
Menu	0	35	614	0	15	0	0	0	0	No	No
Fat1	124	194	561986	0	16	10	0	0	0	No	No
Fat2	33	102	13001	0	7	3	0	0	0	No	No
Inst_1	5	52	14203	0	2	0	0	0	0	No	No
Formulas_1	5	69	5840	0	14	0	0	0	0	No	No
HelpSheet	0	10	1264	0	0	0	0	0	0	No	No

VBA Module Report for FAME's Calc.x*Report generated: Monday, April 09, 2001 11:49:1*

VB Component	Code Lines	No. Procedures
PrintMod	27	3
ThisWorkbook	31	2
Sheet3	3	1
Sheet11	0	0
Sheet21	3	1
Sheet31	0	0
Module1	202	18
Module2	15	2
Sheet1	0	0
Sheet2	0	0
AboutBox	8	2
Module11	7	2
FormHelp	89	8
HelpMod	5	1

Figure 5B. VBA module report for the file FAME's Cal.xls. The macros and code lines that run the worksheets on Figures 1, 2, 3 are included.

Link Report for: C:\~ACNA Spreadsheet Applications\FAME's Calc.xls*Report generated: 4/9/01 11:51:36 AM***FileLinks**

(none)

Formula Links

(none)

Name Links

(none)

XL5/95 Dialog Links

(none)

Chart Links

(none)

Object Links

(none)

OLE/DLL Links

(none)

* Indicates a hidden object

Note: Any linked chart titles or chart data labels are not included in this report.

Figure 6. Link Summary Report for worksheet of Figure 2 generated with PUP v5.0.

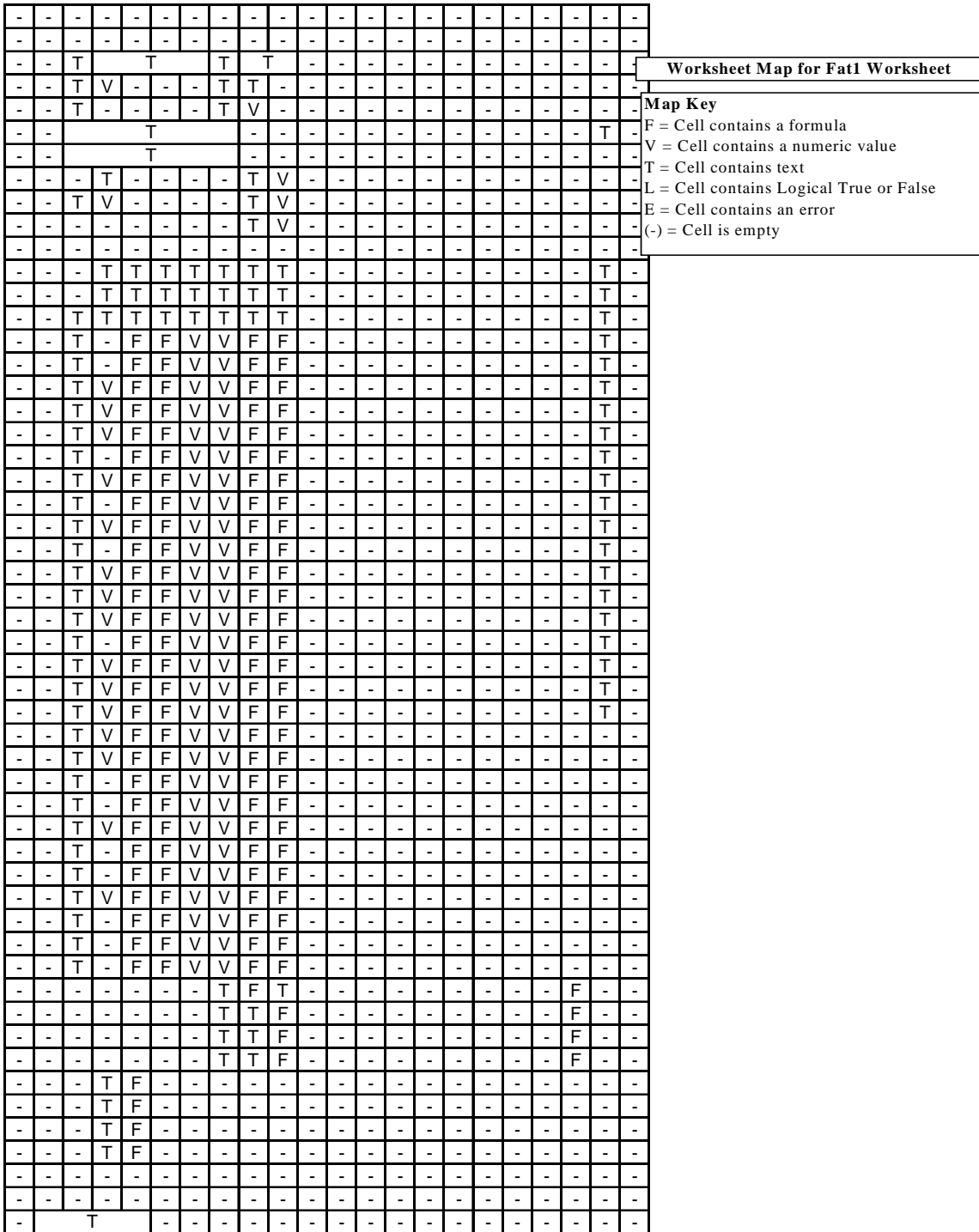


Figure 7. Map report for worksheet in Figure 2 generated with PUP v5.0. The graphic display shows where formulas, values, texts, error and logic are in the audited worksheet.

Date Report for: C:\~ACNA Spreadsheet Applications\FAME's Calc.xls

Workbook date system: 2000

Worksheet name: Fat1

Report generated: 4/9/01 12:29:37 PM

Address	Row	Col	Formula	Display	No. Format	Serial No.	Year	Month	Day
I5	5	9	(none)	02/08/99	mm/dd/yy	36199	1999	2	8
C15	15	3	(none)	4:0	General	0.17	1999	2	8
C16	16	3	(none)	6:0	General	0.25	1999	2	8
C17	17	3	(none)	8:0	General	0.33	1999	2	8
C18	18	3	(none)	10:0	General	0.42	1999	2	8
C19	19	3	(none)	12:0	General	0.5	1999	2	8
C20	20	3	(none)	13:0	General	0.54	1999	2	8
C21	21	3	(none)	14:0	General	0.58	1999	2	8
C22	22	3	(none)	14:1	General	0.58	1999	2	8
C23	23	3	(none)	15:0	General	0.63	1999	2	8
C24	24	3	(none)	15:1	General	0.63	1999	2	8
C25	25	3	(none)	16:0	General	0.67	1999	2	8
C27	27	3	(none)	17:0	General	0.71	1999	2	8
C28	28	3	(none)	17:1	General	0.71	1999	2	8
C29	29	3	(none)	18:0	General	0.75	1999	2	8
C32	32	3	(none)	18:2	General	0.75	1999	2	8
C33	33	3	(none)	18:3	General	0.75	1999	2	8
C34	34	3	(none)	18:4	General	0.75	1999	2	8
C35	35	3	(none)	20:0	General	0.83	1999	2	8
C36	36	3	(none)	20:1	General	0.83	1999	2	8
C37	37	3	(none)	20:5	General	0.84	1999	2	8
C38	38	3	(none)	20:6	General	0.84	1999	2	8
C39	39	3	(none)	22:0	General	0.92	1999	2	8
C40	40	3	(none)	22:1	General	0.92	1999	2	8
C41	41	3	(none)	22:5	General	0.92	1999	2	8
C42	42	3	(none)	22:6	General	0.92	1999	2	8

Figure 8. Displayed date reported and date format.

*Name report for C:\~ACNA Spreadsheet Applications\FAME's Calc.xls
Report generated 4/9/01 12:36:12 PM*

Name	Refers to	Sheet level	Defined in	Hidden	Linked	Bad
Menu!Print_Area	=Menu!\$E\$63:\$G\$77	TRUE	Menu	FALSE	FALSE	FALSE
Fat1!Print_Area	=Fat1!\$B\$2:\$L\$53	TRUE	Fat1	FALSE	FALSE	FALSE
Fat2!Print_Area	=Fat2!\$B\$2:\$K\$58	TRUE	Fat2	FALSE	FALSE	FALSE
Inst_1!Print_Area	=Inst_1!\$B\$2:\$K\$51	TRUE	Inst_1	FALSE	FALSE	FALSE
Formulas_1!Print_Area	=Formulas_1!\$E\$63:\$G\$77	TRUE	Formulas_1	FALSE	FALSE	FALSE
ClearSpl1	=Fat1!\$D\$3,Fat1!\$D\$4,Fat1!\$D\$9,Fat1! \$I\$3,Fat1!\$I\$4,Fat1!\$I\$5,Fat1!\$J\$8:\$J\$1 0,Fat1!\$D\$15:\$D\$19,Fat1!\$D\$21:\$D\$42 ,Fat1!\$C\$6:\$H\$7	FALSE	FAME's Calc.xls	FALSE	FALSE	FALSE
ClearStd1	=Fat2!\$I\$12,Fat2!\$D\$10:\$D\$14,Fat2!\$D \$16:\$D\$37	FALSE	FAME's Calc.xls	FALSE	FALSE	FALSE
Fat1!PrintArea	=Fat1!\$B\$2:\$L\$52	FALSE	FAME's Calc.xls	FALSE	FALSE	FALSE
Fat2!PrintArea	=Fat2!\$B\$2:\$K\$57	FALSE	FAME's Calc.xls	FALSE	FALSE	FALSE
Menu!PrintArea	=\$C\$4:\$G\$17	FALSE	FAME's Calc.xls	FALSE	FALSE	FALSE

Figure 9. Display of the named cells and its cell references including the page name.

Formula Report for: C:\~ACNA Spreadsheet Applications\FAME's Calc.xls

Worksheet name: Fat1

Number of formulas: 124

Report generated: 4/9/01 12:40:16 PM

Address	Row	Column	Display	Number Format	Formula	All Precedents	Direct Precedents
E15	15	5		0.0000	=IF(D15="", "", ROUND(+Fat2!F10,4))	1	1
F15	15	6		0.00000	=IF(D15="", "", (D15*\$J\$8/1000)/(\$J\$10*E15) *(\$J\$9))	5	5
E16	16	5		0.0000	=IF(D16="", "", ROUND(+Fat2!F11,4))	1	1
F16	16	6		0.00000	=IF(D16="", "", (D16*\$J\$8/1000)/(\$J\$10*E16) *(\$J\$9))	5	5
E17	17	5	0.8776	0.0000	=IF(D17="", "", ROUND(+Fat2!F12,4))	1	1
F17	17	6	0.00545	0.00000	=IF(D17="", "", (D17*\$J\$8/1000)/(\$J\$10*E17) *(\$J\$9))	5	5
E18	18	5	0.9397	0.0000	=IF(D18="", "", ROUND(+Fat2!F13,4))	1	1
F18	18	6	0.00527	0.00000	=IF(D18="", "", (D18*\$J\$8/1000)/(\$J\$10*E18) *(\$J\$9))	5	5
E19	19	5	0.9867	0.0000	=IF(D19="", "", ROUND(+Fat2!F14,4))	1	1
F19	19	6	0.04656	0.00000	=IF(D19="", "", (D19*\$J\$8/1000)/(\$J\$10*E19) *(\$J\$9))	5	5
E20	20	5		0.0000	=IF(D20="", "", ROUND(+Fat2!F15,4))	1	1
F20	20	6		0.00000	=IF(D20="", "", (D20*\$J\$8/1000)/(\$J\$10*E20) *(\$J\$9))	5	5
E21	21	5	1.0132	0.0000	=IF(D21="", "", ROUND(+Fat2!F16,4))	1	1
F21	21	6	0.02114	0.00000	=IF(D21="", "", (D21*\$J\$8/1000)/(\$J\$10*E21) *(\$J\$9))	5	5
E22	22	5		0.0000	=IF(D22="", "", ROUND(+Fat2!F17,4))	1	1
F22	22	6		0.00000	=IF(D22="", "", (D22*\$J\$8/1000)/(\$J\$10*E22) *(\$J\$9))	5	5
E23	23	5	1.0302	0.0000	=IF(D23="", "", ROUND(+Fat2!F18,4))	1	1
F23	23	6	0.00014	0.00000	=IF(D23="", "", (D23*\$J\$8/1000)/(\$J\$10*E23) *(\$J\$9))	5	5
E24	24	5		0.0000	=IF(D24="", "", ROUND(+Fat2!F19,4))	1	1
F24	24	6		0.00000	=IF(D24="", "", (D24*\$J\$8/1000)/(\$J\$10*E24) *(\$J\$9))	5	5
E25	25	5	1.0607	0.0000	=IF(D25="", "", ROUND(+Fat2!F20,4))	1	1
F25	25	6	0.03979	0.00000	=IF(D25="", "", (D25*\$J\$8/1000)/(\$J\$10*E25) *(\$J\$9))	5	5
E26	26	5	1.0489	0.0000	=IF(D26="", "", ROUND(+Fat2!F21,4))	1	1
F26	26	6	0.00024	0.00000	=IF(D26="", "", (D26*\$J\$8/1000)/(\$J\$10*E26) *(\$J\$9))	5	5
E27	27	5	1.0499	0.0000	=IF(D27="", "", ROUND(+Fat2!F22,4))	1	1
F27	27	6	0.00032	0.00000	=IF(D27="", "", (D27*\$J\$8/1000)/(\$J\$10*E27) *(\$J\$9))	5	5
E28	28	5		0.0000	=IF(D28="", "", ROUND(+Fat2!F23,4))	1	1
F28	28	6		0.00000	=IF(D28="", "", (D28*\$J\$8/1000)/(\$J\$10*E28) *(\$J\$9))	5	5
E29	29	5	1.0876	0.0000	=IF(D29="", "", ROUND(+Fat2!F24,4))	1	1
F29	29	6	0.03956	0.00000	=IF(D29="", "", (D29*\$J\$8/1000)/(\$J\$10*E29) *(\$J\$9))	5	5

Figure 10. Formula addresses (Generated with PUP v5.0) for worksheet shown in Figure 2. This is only an example and not all the formulas are shown.

VBProject: hlpdemo8.xls (FAME's Calc.xls)

Report Generated 4/9/01 1:00:33 PM

VB Component	Component Type	Procedure Name	Procedure Type	Start Line	Total Lines	Procedure Declaration
PrintMod	Standard Module	PrintForms	Sub	2	17	Sub PrintForms()
PrintMod	Standard Module	EditData	Sub	19	5	Sub EditData()
PrintMod	Standard Module	ReturnToForm	Sub	24	5	Sub ReturnToForm()
ThisWorkbook	Document Module	workbook_open	Sub	1	7	Private Sub workbook_open()
ThisWorkbook	Document Module	Workbook_BeforeClose	Sub	15	24	Private Sub Workbook_BeforeClose(Cancel As Boolean)
Sheet3	Document Module	Worksheet_SelectionChange	Sub	1	3	Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range)
Sheet11	Document Module	(no procedures)	(no procedures)	(no procedures)	0	(no procedures)
Sheet21	Document Module	Worksheet_SelectionChange	Sub	1	3	Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range)
Sheet31	Document Module	(no procedures)	(no procedures)	(no procedures)	0	(no procedures)
Module1	Standard Module	Auto_Open	Sub	2	23	Sub Auto_Open()
Module1	Standard Module	DeleteToolbar	Sub	25	14	Sub DeleteToolbar()
Module1	Standard Module	SiblingWorkbooks	Function	38	24	Function SiblingWorkbooks(WorkbookName, NumberHurdle)
Module1	Standard Module	ShowToolbar	Sub	62	5	Sub ShowToolbar()
Module1	Standard Module	HideToolbar	Sub	69	7	Sub HideToolbar()
Module1	Standard Module	SampleMacro	Sub	75	11	Sub SampleMacro()
Module1	Standard Module	GoTo_Fat1	Sub	86	8	Sub GoTo_Fat1()
Module1	Standard Module	GoTo_Fat2	Sub	94	8	Sub GoTo_Fat2()
Module1	Standard Module	GoTo_Menu	Sub	102	8	Sub GoTo_Menu()
Module1	Standard Module	GoTo_Spl1	Sub	110	8	Sub GoTo_Spl1()
Module1	Standard Module	GoTo_Std	Sub	118	7	Sub GoTo_Std()
Module1	Standard Module	GoTo_Inst_1	Sub	125	7	Sub GoTo_Inst_1()
Module1	Standard Module	GoTo_Formulas_1	Sub	132	7	Sub GoTo_Formulas_1()
Module1	Standard Module	GoTo_Top2	Sub	143	12	Sub GoTo_Top2()
Module1	Standard Module	GoTo_Top1	Sub	151	8	Sub GoTo_Top1()
Module1	Standard Module	ClearCells1	Sub	159	14	Sub ClearCells1()
Module1	Standard Module	ClearCells2	Sub	173	11	Sub ClearCells2()
Module1	Standard Module	ClearCells3	Sub	184	20	Sub ClearCells3()
Module2	Standard Module	Print_Page_Fat1	Sub	1	7	Sub Print_Page_Fat1()
Module2	Standard Module	Print_Page_Fat2	Sub	9	8	Sub Print_Page_Fat2()
Sheet1	Document Module	(no procedures)	(no procedures)	(no procedures)	0	(no procedures)
Sheet2	Document Module	(no procedures)	(no procedures)	(no procedures)	0	(no procedures)
AboutBox	Form	Label2_Click	Sub	2	4	Private Sub Label2_Click()
AboutBox	Form	OKButton_Click	Sub	6	4	Private Sub OKButton_Click()
Module11	Standard Module	ShowAboutBox	Sub	1	3	Sub ShowAboutBox()
Module11	Standard Module	ShowInstructionsBox	Sub	5	4	Sub ShowInstructionsBox()
FormHelp	Form	CommandButton1_Click	Sub	12	9	Private Sub CommandButton1_Click()
FormHelp	Form	LabelText_Click	Sub	21	4	Private Sub LabelText_Click()
FormHelp	Form	UserForm_Initialize	Sub	25	13	Private Sub UserForm_Initialize()
FormHelp	Form	UpdateForm	Sub	38	25	Private Sub UpdateForm()
FormHelp	Form	ComboBoxTopics_Click	Sub	63	6	Private Sub ComboBoxTopics_Click()
FormHelp	Form	PreviousButton_Click	Sub	69	8	Private Sub PreviousButton_Click()
FormHelp	Form	NextButton_Click	Sub	77	8	Private Sub NextButton_Click()
FormHelp	Form	ExitButton_Click	Sub	85	6	Private Sub ExitButton_Click()
HelpMod	Standard Module	ShowHelp	Sub	3	4	Sub ShowHelp()

Figure 11. Display of the Visual Basic project for the workbook file.

VBProject: hlpdemo8.xls (FAME's Calc.xls)*UserForm Report Generated 4/9/01 1:00:33 PM*

UserForm Name	Control Type	Control Name	Left	Top	Width	Height	Container
AboutBox	Label	Label1	12.0000	30.0000	198.0000	54.0000	AboutBox
AboutBox	CommandButton	OKButton	168.0000	90.0000	42.0000	20.0000	AboutBox
AboutBox	Label	Label2	12.0000	6.0000	66.0000	18.0000	AboutBox
FormHelp	CommandButton	ExitButton	12.0000	162.0000	49.5000	19.5000	FormHelp
FormHelp	CommandButton	NextButton	216.0000	162.0000	49.5000	19.5000	FormHelp
FormHelp	CommandButton	PreviousButton	156.0000	162.0000	49.5000	19.5000	FormHelp
FormHelp	Label	LabelText	4.0000	3.8500	212.0000	122.1500	Frame1
FormHelp	Frame	Frame1	12.0000	30.0000	252.0000	120.8000	FormHelp
FormHelp	ComboBox	ComboBoxTopics	12.0000	6.0000	252.0000	18.0000	FormHelp
FormHelp	CommandButton	CommandButton1	72.0000	162.0000	72.0000	20.0000	FormHelp

Figure 12. Display of the Visual Basic project for the workbook file user form names, which control the dialog boxes of the application.

Sub DeleteToolbar()

```
' Delete the toolbar from the collection
' This keeps the EXCEL5.XLB file smaller
' Cancel the toolbar display
If IsNull(SiblingWorkbooks(BookName, 1)) Then
  Application.OnWindow = ""
  With ThisWorkbook
    .OnSheetActivate = ""
    .OnSheetDeactivate = ""
  End With
  Toolbars("Fatty Acids Toolbar").Delete
End With
End If
```

End Sub

Figure 13. Example of a VBA macro in the Module 1 composed of nine code lines (Module 1 is composed of a total of 202 code lines) that must be included in the validation documentation.

Request for Change		
Analytical Application: FAMEs Calc.	Change No.: 1	
Location (Branch): ACNA	Telephone no: 404-253-1200, x5217	
Originator: James Martin		
Reason for change: Correct formula	Required by date: 05/07/02	
Change Disposition & Authorization		
Disposition: Accepted		
Authorized / Rejected		
Signature: _____	Job title: Analyst	Date: 05/07/02
Signature: _____	Job title: Director	Date: 05/07/02
Change Details		
Comments: Formula in cell K19, page "FAME Calc", the rounding formula is incorrect.		
Number of Change Notes attached: 1		
Change Completion and approval		
Completed signature: _____	Job title: Analyst	Date: 05/07/02
Approved by signature: _____	Job title: Supervisor	Date: 05/07/02
Signature: _____	Job title: Director	Date: 05/07/02

Request for Change Form v1.0

Figure 14. Example for the request for change form. This example is for correcting a formula.

Change Note	
Analytical Application: FAMEs Calc.	
Type of Change: Correct formula	
Note Number: 1	
Application Version: FAMEs Calc. V3.0	
Details of Change to perform: Formula in cell K19, page "FAME Calc", the rounding formula is incorrect.	
Implemented by: James Martin	Print name:
Signature:	Date: 5/7/02

Change Note Form v1.0

Figure 16. Example for change note form. This form is use when more than one change is requested. Use one separate change note form for each item to be change.