

# INTRODUCING SAS/PH-CLINICAL<sup>®</sup> 2.0 SOFTWARE

Sandra D. Schlotzhauer, Andrew T. Fagan, and Martin F. Michael  
SAS Institute Inc.

## INTRODUCTION

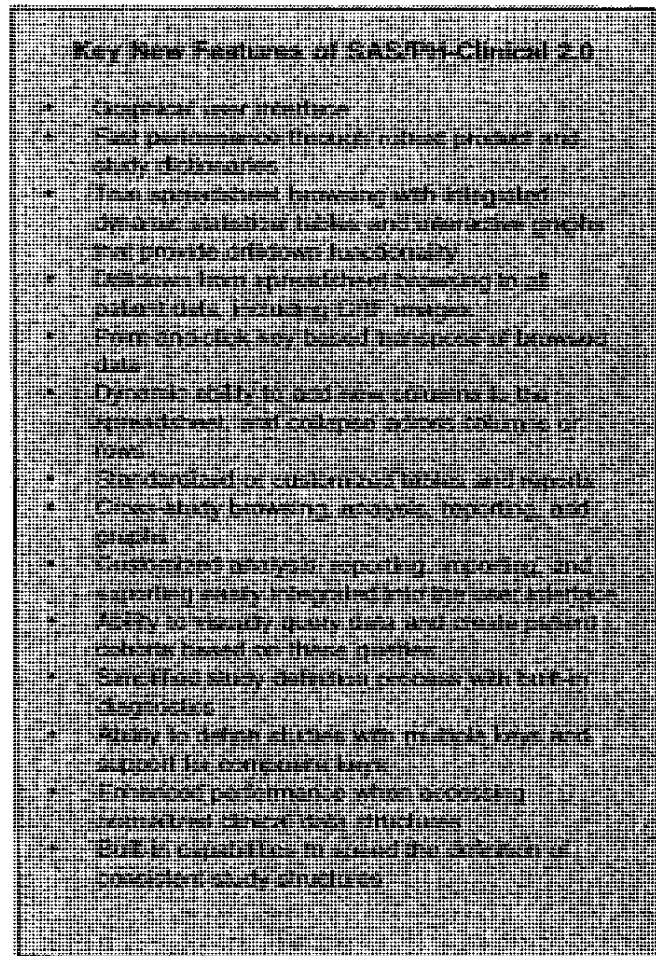
SAS/PH-Clinical 2.0 is a new, GUI-based release of the Institute's clinical data review software. This major new release of the software provides a generational leap in technology motivated by customer experiences with the first release and by the new tools available for application development in the SAS<sup>®</sup> System. SAS/PH-Clinical 2.0 is a major advancement both for internal review at pharmaceutical and biotechnology companies and for CANDAs. The result is a major step forward in facilitating the goal of enabling sponsors to use a single process for internal review and for CANDA preparation. SAS/PH-Clinical 2.0 is available with Release 6.11 on all PC and Unix desktop environments, including the Macintosh and Power PC architectures.

The study definition process is now substantially automated and data-driven. For each study, you provide minimal information to create a study dictionary, which is always synchronized with the clinical data. This ensures that queries to the data are addressed efficiently and accurately. Further, the study dictionaries are created from a product dictionary, which ensures that studies can be combined for pooled analyses even when their data structures are inconsistent. The software handles normalized data with many keys efficiently, and requires virtually no data manipulation and minimal pre-processing. The design of the new release supports access to non-patient data, such as adverse event dictionaries and lab normal range tables.

The user environment provides a sophisticated, yet simple to use, data browser that includes a true spreadsheet format, which provides a feature to open multiple spreadsheets at once. From the spreadsheet, you can browse using interactive graphs or statistical tables. You can also browse data from patient records and CRFs. Using the CRF image viewer, you can drilldown to images from the spreadsheet, graphs or statistical tables. You can build queries at any time to create new patient cohorts, and use the new visual query facility to precisely pinpoint data of interest. With the data manipulation tools in the browse spreadsheet, you can create new variables that summarize across columns or condense rows as sums or means. You can also transpose the spreadsheet, creating new spreadsheets to look at the data in a different way.

The new software provides all the reporting functionality available in the first release and substantially extends it to include, for example, adverse event occurrence reports and laboratory test transition tables. These are implemented using the powerful new Templates facility, which enables you to fully customize and extend SAS/PH-Clinical 2.0. By providing seamless hooks into the SAS/PH-Clinical 2.0 interface and dictionary systems, this feature enables you to add new functionality or to incorporate existing programs quickly and easily.

SAS/PH-Clinical 2.0 provides the next generation in clinical data review, extending the software beyond a review tool to fast, friendly, dynamic software that can perform and document virtually any analysis of clinical data.



Key enhancements for SAS/PH-Clinical 2.0 are highlighted in the box above; the remainder of the paper provides details and examples.

## A NEW USER INTERFACE

The best way to describe the new graphical user interface for SAS/PH-Clinical 2.0 is to provide examples. Figure 1 displays the new main window for SAS/PH-Clinical 2.0. (Note that all figures appear at the end of the paper.) This window illustrates a key principle for the GUI: when the software starts, you can just start to work. The new interface is easy to learn and does not surface power unnecessarily. The power and customization are available, but you can begin to work with only a few features. In Figure 1, the active study is immediately available for browsing and analysis. This is a result of your preferences, which you can change at any time.

Note that there may be some appearance differences between the production release and the figures shown in this paper. The next several sections provide additional detail on the new user interface.

## ADHERENCE TO WINDOWING GUIDELINES

By following available interface guidelines, the new SAS/PH-Clinical interface is easier to learn since it leverages your knowledge gained from other activities such as word processing, spreadsheets, or other familiar tools. As an example, you double-click on the desktop icons to specify details for tables and graphs. Clicking the right mouse button displays a popmenu of commonly-performed actions. These actions are also available by selecting the object and then selecting an item on the toolbar or menu bar.

New features available in SAS Software, including the FRAME object, icons, toolbars, and drag and drop, allow us to offer an interface that provides this functionality. For more detail on general new GUI features for the SAS System, see "Enhancements to the Graphical User Interface for the SAS System on Personal Computers" (Rigsbee, SUG119).

## USING FOLDERS

Another aspect of the new user interface is a reduction in the layers of windows, and an increase in the number of visible windows. Although this is in keeping with windowing guidelines, more importantly, it reflects how reviewers want to work with data during the clinical review process. Figures 1 and 2 illustrate the principle of reducing the numbers of layers of windows.

Specifically, you choose to display items—programs, output, graphs, and so on—in a way that makes sense to you. In the new release, all of these items are available from a single window as is shown in Figure 1. You can, of course, organize these items any way you wish by defining folders that best meet your needs. Related items can be placed in *folders*, which group the items together. You can switch easily between folders as you work. The ability to group items together in Public and Private folders is still available. However, this grouping may not meet your needs. For example, you may want to group all of your AE reports together in a folder named AE and all of your Efficacy reports in another folder named Efficacy as is shown in Figure 2.

The folders are not only a way to display items together, they are a way to operate on a group of items. You can move, copy, rename or delete an entire folder. This gives you an easy way to share a group of items with someone else or a quick way to copy a set of objects to another package (for example, WordPerfect®).

Equally important, items in the folders are now tightly coupled with each other. For example, a saved table or graph is integrally linked with the program that produced it and the log generated when it was produced. Any operation performed on the output or graph (move, rename, delete and so on) is automatically performed on the associated items. There is also an inherited association for each item. For example, every item in the library is integrally linked to a study; specifically the study that was active when the item was produced. This allows you to simply choose the outputs to include as part of a CANDA or CAPLA submission, and know that the related programs, logs, and notes are also automatically included in the submission.

## POWER TO CUSTOMIZE

The new point-and-click interface is easy to use, even if you work with only a few of the features. In addition, significant power to customize the interface is available to you. As an example, Figure 3 displays a computer screen with several windows from SAS/PH-Clinical 2.0. A portion of the main window is visible, and identifies the active study. The browse spreadsheet is also active, and displays the variables requested for the patients in the study. A CRF image is also visible. With this interface, you can easily cross-reference between the browse spreadsheet and appropriate images.

Finally, note the Preferences choice on the menu bar. Most windows provide you with the ability to modify the default appearance and action. One example is the main window: you choose whether the software starts with an active study. The Preferences choice on the menu bar gives you the ability either to change your preferences for the current window, or to view an index of available preferences—and change them if you like. Thus, while you can easily work with a single window and customize that window to meet your needs, you can also work with multiple active windows with each customized as you wish.

Because of the significant power of the new interface, SAS/PH-Clinical 2.0 is available on platforms that support a GUI. Providing a state-of-the-art interface requires a hardware platform that supports that interface. This means that the software is available on PCs and Unix workstations.

## BROWSING CLINICAL DATA

Both medical and statistical reviewers at sponsor firms and regulatory agencies are comfortable with viewing data in a spreadsheet format. SAS/PH-Clinical 2.0 provides a true spreadsheet browser. Figure 3 showed a portion of a browse window; Figure 4 shows the complete window. With the new browse spreadsheet, you can drag column widths to meet your needs, rearrange or delete columns, and add new columns. For example, this enables a reviewer at the FDA to create a new column that contains the reviewer's reclassification of a patient's evaluability. The software includes several pre-defined ways to create new columns, or *derived variables*. These include averaging across columns, averaging by key, and arithmetic operations between columns (adding, subtracting, multiplying, and so on). The browser also provides functionality not typically available in spreadsheets; including the ability to dynamically transpose rows and columns by a key and the ability to easily switch to a scatterplot view of selected columns. The next several sections provide details on features of the browser.

## SPREADSHEETS, GRAPHS, AND STATISTICS

Browsing data typically generates a series of quick questions that need quick answers. Looking at a column of numbers often begs for a graph for quick visual interpretation—or for a set of summary statistics. Rather than exiting from the browser to a reporting tool as is required by other software,

these answers are readily available from the spreadsheet. By clicking on a drop-down list, you can change the view of the data from the spreadsheet to a set of summary statistics, a bar chart or scatterplot (or one of several other graphs), or to a CRF image. Referring to Figure 4, suppose you click on the ECGHR and ECGPR variables, and then choose a scatterplot from the drop-down list. Figure 5 shows the results.

### DRILLDOWN TO CRFs AND IMAGES

One of the most popular features is the ability to drilldown from a listing to a more detailed CRF-browse of patient data. From any browse format (the spreadsheet, plots, or summary statistics), you can drilldown to a demographic summary window, a CRF window, a scanned image of the CRF, or a custom window of your own design.

When you drilldown to an image, the window displays the image, a counter identifying the current image number, and Next/Previous buttons that allow you to scroll through the list of images. Depending on your selection in the spreadsheet, any number of images may be available in the image window. To facilitate better review of the images, you can manipulate the viewed image. This includes darkening or lightening, mirroring, displaying as a photographic negative, and rotating. Note that this edits only the displayed view for better review; the stored image is unchanged.

The slideshow feature provides another way to look at images in a given order. You can subset the current list of images and then display them one-by-one with either manual or automatic image advancement. For example, suppose all images for patient 32 are displayed in the image window initially, but you only want to see CRF images for the first three visits. You can easily subset the list of images and view just those for visits 1-3.

### MANIPULATING THE DATA - EXAMPLE

No single structure for a given set of clinical information is ideal. For example, consider lab data. A structure where a patient's visits are recorded on different rows of the spreadsheet facilitates plotting RBC against HGB for each of the patient visits, for example. Suppose you want to plot each patients' average values for these two lab tests against each other. Imagine you have a dataset stored and displayed in the spreadsheet as follows:

PATIENT	VISIT	LABTEST	RESULT
1	1	RBC	6.7
	1	HGB	12.6
	2	RBC	5.4
	2	HGB	13.3
2	1	RBC	6.2
	1	HGB	13.6
	2	RBC	6.0
	2	HGB	12.9

You want to plot the average RBC value against the average HGB value for each patient. The SAS/PH-Clinical 2.0 browser enables you to do this in just three point-and-click operations.

First, *flip* (transpose) the spreadsheet on RESULT. The underlying study dictionary knows that this column is keyed to LABTEST (more on this later). No other selections are necessary to obtain the new spreadsheet shown below:

PATIENT	VISIT	RBC	HGB
1	1	6.7	12.6
	2	5.4	13.3
2	1	6.2	13.6
	2	6.0	12.9

Second, obtain the average of the new RBC and HGB variables. The dictionary already knows that these are keyed to VISIT, so after a second mouse-click, the data appears as follows:

PATIENT	MEANRBC	MEANHGB
1	6.05	12.95
2	6.10	13.25

Third, choose the MEANRBC and MEANHGB columns to plot.

Suppose, instead of generating a plot, your goal is to evaluate a patient's change in blood chemistry over time. In this situation, you want to see the various lab tests in a single row of the spreadsheet. This makes it easier to read across the row and visualize how the patient's information has changed. This structure is also easiest for creating difference variables; to compare the patients' initial and final values, for example. You can generate this structure from the dataset shown above. Additional features for data manipulation in the browser enable you to create new spreadsheets, generate summary statistics, and export the data to other spreadsheet or word-processing software. By providing this ability to perform dynamic manipulations of the browsed data without affecting the underlying clinical datasets, SAS/PH-Clinical 2.0 helps reduce the need for parallel data structures.

### REGENERATIVE AUDIT TRAIL

In the example above, the two data manipulation steps are both saved in an audit trail. This audit trail, not only documents (in English, not computerese) the manipulations you've done, but also enables the manipulations to be regenerated by you, or another user in your QA or Regulatory Affairs Department. You can also save the new columns and spreadsheets in the dictionary so that you can use them in later sessions without regenerating them.

The data manipulation tools are not only designed for the data reviewer. As clinical data is collected and managed, it is often arranged in a manner more ideal for data storage than for data review. The data manipulation tools can also be used by statisticians or data analysts to organize and transform the data into a form suitable for tabulation, summary or statistical analysis. These restructuring tasks often require a significant time investment, since analysts must first verify that complex programs are restructuring the data correctly and then must verify the results of an analysis. In addition, the data are typically duplicated as a result of restructuring, which adds more validation and QA tasks each time the data change. These two labor-intensive operations are substantially reduced by SAS/PH-Clinical 2.0 software.

# TEMPLATES FOR STANDARD AND AD-HOC REVIEWS

The concept of a library of reports, analyses, graphs and so on for each study is closely tied to the concept of templates. Reviewers—at sponsor firms and at the FDA—typically want to generate (or re-generate) a set of reports, analyses, and graphs. Some of these review tasks are standard in that they change little from study to study. Other tasks are ad-hoc in that they are customized review tasks for a given study, or are tasks suggested by earlier reviews of the clinical data for a study. As another alternative, these review tasks may be comprised of a set of standard reports prepared by a sponsor. Templates provide a straightforward way to integrate these standard reports directly into the software to contribute to shared knowledge across groups at the sponsor company, or as part of a CANDAs or CAPLA.

## OVERVIEW

SAS/PH-Clinical 2.0 combines standard and ad-hoc review tasks in the Templates facility. The software includes a library of supplied templates, which encompasses all the functionality for ad-hoc review tasks provided in the Tables, Graphs, and Analysis components in the first release. In addition, new templates provide standard reports that extend the functionality. For example, one template provides a histogram with a fitted normal curve and summary statistics displayed on the graph. To use this template, you choose the variable to plot. Another template provides a standard report summarizing demographic information for patients at the different centers for the study.

The complete set of templates is a reflection of increased communication between sponsors, regulatory agencies, and the Institute about the types of review tasks performed by medical and statistical reviewers. The set of templates provided also reflects the increased public communication about the needs for CANDAs and CAPLA systems. As both sponsors and agency reviewers present their experiences with these systems, the industry as a whole gains an increased awareness of the common functionality desired across all drug review systems. The efforts of the PMA-FDA CANDA working group to provide and update the *CANDA Guidance Manual* have also been significant in disseminating the global set of needs for a clinical data review product.

Although the set of Institute-supplied templates reflects this increased communication, it cannot encompass all types of analyses and reports performed on clinical data. Thus, you can add new templates to the set of Institute-supplied templates. For example, as part of a CANDA, you may choose to add templates based on existing programs that perform efficacy analyses, generate the integrated safety summary, provide cross-study comparisons, and more. The key for a CANDA—and for internal review—is the seamless integration of your own reporting programs.

As a brief technical summary, the Templates facility provides you with the ability to create generic reports that can run against any study without making any changes to the template. The Templates facility incorporates a program editor with a process that allows all dataset and variable names to be coded as *substitution fields* rather than requiring you to supply the actual dataset and variable names. When the program runs, these fields are replaced with the correct dataset and variable names for the current study, resulting in a correct program for the current set of clinical data. Further, if your program requires several variables to be in the same dataset and this

is not the case, the software dynamically builds the dataset containing the required variables. This is possible because of the underlying study dictionary.

Templates also enable you to associate a user window with each program. This is especially useful when you want to add new ad-hoc reviewing tasks to the software. You can define the template so the user window displays when users select the template from the desktop. In the window, users select variables and supply other details for the ad-hoc review task. For SAS programmers, writing a template consists of supplying the SAS procedure code and designing the user window. However, this design process does not require you to know SCL. Note that user windows are not required; you can build templates that simply run when a user double-clicks on the icon in the desktop.

• Standard and ad-hoc review tasks
• Capable to construct integrated programs to integrate
• Flexible reports based on user choices
• Ability to build customized reports
• Easy to update and consistent files
• Task windows can exist in the same window or in separate
• Customized substitution of all keywords
• New templates can be defined

The table above summarizes major features of the Templates facility.

## EXAMPLE

To provide more information about templates, we will first use an example and then discuss more technical details. Figure 6 displays a user window for a template that creates histograms of one or more variables. Figure 7 displays the results of choosing the AGE variable and then choosing the Run button in the user window, and Figure 8 displays the SAS log generated when the template is run. (Recall that all figures appear at the end of the paper.)

## BUILDING TEMPLATES

Figures 6-8 display how templates appear to every user, including those users who build templates. For the template builder, however, the underlying source code must be entered, and substitution fields defined. This enables the template to run across any study with any variable. Or, by explicitly specifying a clinical dataset and variables for the SAS procedures, you can build templates that generate study-specific reports.

Consider a simple printing task. Suppose the program for printing adverse event information from a given study is as shown at the top of the next page:

```

proc print data=mylib.adverse noobs ;
  id patnum ;
  var trt aecode aedesc severity related
    recur hosp death ;
  format aecode $whocode. trt $TRT. ;
  title 'Adverse Events Summary' ;
run;

```

This program may work for one or more studies, but perhaps not for all studies, given the inconsistencies in variable names, dataset structures, and the like. Or, you may want to add other variables to the report, such as additional demographics or related concomitant medications. If you were to revise the program above to use macro variables, it might look as shown below.

```

proc print data=&data noobs ;
  id &patid ;
  var &trtvvars &codvar &dvar &othvars ;
  format &codvar $whocode. &trtvar $TRT.;
  title &rptitl ;
run;

```

This is **exactly** the same code that converts the program to a template. The familiar &-variables are not macro variables, but instead indicate *substitution fields*, whose values are supplied from the underlying dictionary.

Incorporating your existing programs into templates is an easy process: copy them into the program editor and change all of the hard-coded dataset and variable names to substitution fields, and then create a user window if you want one. If the programs are written as macros, conversion is even easier because the substitution fields used in templates resemble macro variables.

If you have study-specific programs where you want to hard-code the datasets and variables, this is still possible. Substitution fields are not required, but they are recommended for the most complete use of the Templates facility. By using substitution fields, you maintain a dynamic library of programs. If a variable name in the underlying clinical dataset changes, the templates do not need updating, since this is handled by the dictionary.

## BUILDING THE USER WINDOW

Another advantage of the Templates facility is all ad-hoc reviewing tasks—those supplied with SAS/PH-Clinical 2.0 and those you create yourself—use the same interface. After providing the SAS code for a template, the Templates facility enables you to build a custom window where users supply information. This is not required; you can simply define the template to run when a user double-clicks on it. In this case, you can still use substitution fields and thus ensure that the resulting report contains the same information, regardless of any cross-study differences in variable naming, dataset structures, and the like. The Institute-supplied templates are all built using the same interface tools available to you, thus ensuring a seamless integration of your own templates with those already available.

By using folders, you can group the Institute-supplied templates and your own templates together, or not. Regardless of your choice, all templates have the same interface and go through the same process. This allows you to use the same process for internal review and for CANDAs or CAPLAs

preparation. You may have a larger set of templates, particularly if you use the software to review several drug types, than you want to supply for a given submission. By building a set of folders for the submission, you can simply subset a portion of the larger set and provide the regulatory reviewers with the standardized reports and ad-hoc review tools relevant for the submission. This accomplishes the goal of using a single process to perform internal clinical data review and to prepare a CANDA or CAPLA.

## VALIDATING TEMPLATES

Validating templates is no different from the process of validating SAS programs used for clinical data review and analysis. By comparing the results from the template with those from another SAS program, you can quickly assure yourself that the template is performing as expected. A key advantage of templates is that once the validation is performed for a given study, it is performed for all studies. And, because of the underlying dictionaries, if the variable names change, you do not need to re-validate the templates.

## MODIFYING INSTITUTE-SUPPLIED TEMPLATES

All the templates provided with SAS/PH-Clinical 2.0 are shipped with user windows and with source code. You can copy these templates and modify them to meet your own special needs. This process is the same as building your own new templates. After copying the template, you simply modify the source code and user window as you wish.

## SHIPPING NEW TEMPLATES

One more bonus is the freedom templates give us to ship enhancements. As new templates are developed at the Institute, they can be made available to all licensed customers quickly. Windows within the software do not change and you do not need to install a new copy of the software. Simply copy the new template and it is ready to be used. This has important implications for submissions as well. If a regulatory reviewer asks for a new standard report or for new ad-hoc reviewing tools, you can develop templates and supply them to both internal and agency reviewers quickly. No modifications of the existing study or the existing interface are required, and validating the source code is straightforward since every user can view the code.

## ACCESS AND DEFINITION OF CLINICAL STUDIES

One of the major design goals for SAS/PH-Clinical 2.0 is to provide simple, direct access to the clinical data so that all users—including the Product Administrator (PA)—can work with the clinical data as it exists with minimal pre-processing. To facilitate this goal, the study definition process is now substantially automated and data-driven.

## DICTIONARIES - OVERVIEW

Data from a clinical trial are typically stored in several datasets. Each dataset has a number of keys, such as the patient identifier, visit number, laboratory test, and so on. These keys are used to identify records in the

datasets and to enable datasets to be joined. The software is designed to access virtually any dataset and key structures. With equal ease, you can define parallel-group and crossover studies, single-center and multi-center trials, and trials with simple or composite keys.

Normalized datasets and their associated keys are the preferred data structures. Pre-processing of clinical data into an optimal structure is no longer necessary. Datasets can be defined as they exist with no performance penalty for common queries. Since datasets are only joined as requested, defining a study that includes many datasets with multiple keys does not adversely affect performance.

## DICTIONARIES - DETAILS

The software contains one or more *product dictionaries*. A product dictionary contains default names and attributes for *functions*. A function can be a type of dataset, a key, a field, a CRF, and so on. Defining a study to the software primarily involves assigning datasets, keys, and variables to functions in the product dictionary—thus forming the *study dictionary*. The study dictionary contains the subset of functions in the product dictionary that map to the clinical data structures for a given study. If naming conventions for datasets, variables, and keys are consistent between studies, then defining a new study is reduced to identifying the datasets and the keys for each dataset. The process is designed to require a minimum of input from you to create a robust study dictionary that controls and facilitates interaction with the underlying clinical data.

By using the functions you assign to datasets, variables, and keys, the software is able to perform analyses across studies even when there are discrepancies in naming conventions between the study datasets and variables. For example, suppose the adverse event dataset in one study is AEDATA and the adverse event dataset for another study is ADRS. If you assign the function "Adverse Events Dataset" to both these datasets, the dictionary recognizes that these datasets are the ones to join to provide a combined summary of adverse events for the two studies. As another example, suppose the variable containing adverse event codes is ADRCODE in one dataset and AECODE in another dataset. If you assign the function "Adverse Event Code" to both variables, then a template that includes the "Adverse Event Code" function variable will produce correct results for both studies, or for the combined summary.

Other additions to the dictionary include support for format libraries, lab normal range datasets, and images. A format library can be associated with each study and is accessed whenever the study is used. Images (for example, scanned copies of the CRFs) can be defined in much the same manner as the clinical data. The images are linked to the clinical data and are accessible while browsing data.

An important aspect of both the product and study dictionaries is that they are dynamic. Whenever users create a new variable (perhaps by adding a new column to the browse spreadsheet and saving the results), the dictionary retains new information just as if it had been supplied during the study definition process. Thus, medical reviewers can essentially enhance the clinical datasets without modifying the original underlying data structures. For example, users can add a new variable that reclassifies patients' evaluability status—by first copying the existing evaluability variable and then modifying the status for individual patients. Both evaluability variables are defined to the study dictionary, and can be used in subsequent review tasks.

As another example, users can transpose a spreadsheet view of the data and save the transposed view. Both the original and transposed spreadsheets are now available for subsequent review tasks.

Perhaps most importantly, dictionary management is done without unnecessarily surfacing the underlying technical details on the structure of the clinical datasets to users. The study dictionary stores enough information to ensure that data manipulations required to complete user requests are performed intelligently and efficiently. Also, the dictionary stores commonly-used information about the data. This information might include: a list of patients, summary statistics for selected columns, adverse event counts for each patient, and so on. Access to this information can be performed very quickly because it is stored in the dictionary and does not need to be calculated each time.

## DIFFERENCES FROM PREVIOUS RELEASE

Since SAS/PH-Clinical 2.0 uses a completely different dictionary structure and a data-driven study definition process, it is difficult to compare the study definition with the previous release. For Product Administrators who are familiar with the previous study definition process, this section briefly discusses some differences.

A major implication of the SAS/PH-Clinical 2.0 study definition process is that patient groups are no longer needed. All variables for all patients are always available—without the overhead and time required to build a patient group. The software joins the clinical datasets only as needed for the task, and bases these joins on information contained in the study dictionary.

One important consequence of no longer using patient groups is that significant pre-processing of your clinical data is not required. The only requirement on the clinical datasets is that they are sorted by the keys. Minimizing the need for pre-processing helps move toward the goal of eliminating duplication of processes for internal and regulatory review. Since the data are typically already stored in SAS datasets, you can work directly against those existing datasets. By using one parallel process for internal and regulatory review, you also eliminate the increased validation and QA tasks associated with duplicate data.

## FUTURES

Many of the changes incorporated into the study definition process and the underlying dictionary structures are being made in order to provide the necessary support for data entry and validation features planned for the future. These features include, but are not limited to: double-key data entry, remote data entry, clinical database design, rules-based validation checks, and normal range checks.

## SUMMARY

SAS/PH-Clinical 2.0 provides a major advancement in clinical data review technology. Key features include a friendly interface; multi-functional browser; dynamic queries; fast data access; a rich set of reports, analyses, and graphs; and automated study definition. Further, you can incorporate additional functionality and can customize the interface according to your preferences.

Since the first release of the software, pharmaceutical and biotechnology companies have gained more experience with considering clinical data review as a process that includes a CANDA or CAPLA submission; and various speakers at industry conferences have identified the goal of using a single process for internal review and CANDA preparation. SAS/PH-Clinical 2.0 facilitates achievement of this goal. Key aspects of the software functionality and their relationship to the goal of achieving a single process are:

- The dictionary technology facilitates using a single data structure for all review tasks, rather than duplicating datasets to generate different reports and analyses.
- The user interface provides a common tool that meets the diverse interface needs of medical reviewers, statisticians, statistical programmers, data analysts, and regulatory reviewers; rather than requiring different tools for different reviewers.
- The browser includes true spreadsheet functionality, drilldown, integrated links to CRF images, and a visual query facility, which eliminates the need to anticipate all different views of the data that any reviewer might want to investigate.
- The browser also provides the ability to dynamically transpose the spreadsheet, add new columns to the spreadsheet, switch to a graphical view of columns in the spreadsheet, and obtain summary statistics on columns in the spreadsheet. Again, this helps eliminate the need to anticipate derived variables of interest to reviewers; and, since restructuring is dynamic, eliminates the need to anticipate the varying structures that help different reviewers attain a comfort level with the clinical data.
- The dictionary provides increased robustness in handling more types of dataset structures, including those with multiple levels of keys, which facilitates the ability to use dataset structures more typical of a database management system.
- Perhaps most importantly for CANDA systems, the Templates facility provides not only a rich set of reports, analyses, and graphs but provides the ability for sponsors to add to the set. This enables each sponsor to provide customization to SAS/PH-Clinical 2.0. And, since the same user interface is used for Institute-supplied and custom templates, all reviewers (internal reviewers and those at the agency) need only learn one interface to generate all types of reports, analyses, and graphs.
- The Templates facility also provides a way for sponsors to leverage their own investment in custom reporting programs. By enabling sponsors to integrate their own custom reporting programs into SAS/PH-Clinical 2.0, the software can be fully customized to the needs of specific reviewers at the sponsor firm or the FDA.

SAS/PH-Clinical 2.0 provides the next generation in clinical review, by providing fast, friendly, dynamic software that can perform virtually any task of interest to reviewers at sponsor firms and regulatory agencies.

## Contact Addresses

Sandra D. Schlotzhauer, Andrew T. Fagan, and Martin F. Michael  
SAS Institute Inc., SAS Campus Drive, Cary, NC 27513

## Acknowledgements

The authors would like to thank other members of the SAS/PH-Clinical development team for providing input and reviewing drafts of this paper.

SAS and SAS/PH-Clinical are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © Indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.

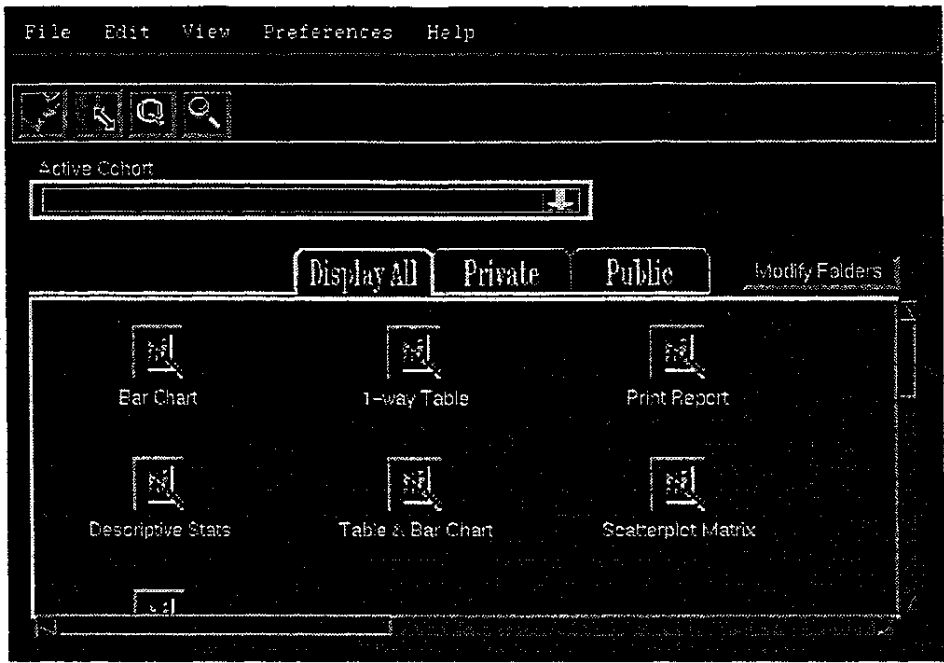


Figure 1

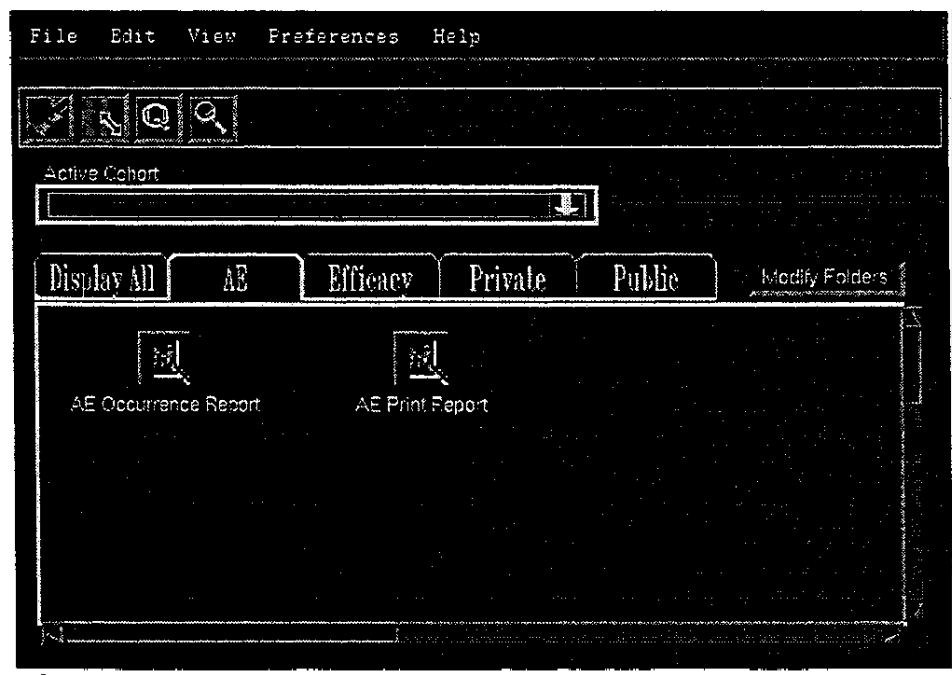


Figure 2



The screenshot displays a software interface with a menu bar (File, View, Actions, Preferences, Help) and a toolbar. A table of observations is visible, with columns for OBS, PROTOCOL, PATIENT, VISIT, AEDATE, ABCODE, and AETEXT. Below the table, a window titled 'PHC2.0 Test Study' contains a form with fields for 'Address Event', 'Patient ID Number', 'Patient Name', and 'Vid Number'. A small table within the form lists 'Study Site' with columns for 'Date', 'Status', and 'Location'.

OBS	PROTOCOL	PATIENT	VISIT	AEDATE	ABCODE	AETEXT
1	ABC123	1	1	05AUG99	01090001	HEADACHE
2	ABC123	1	2	06AUG99	04550001	EDEMA
3	ABC123	2	1	05MAY99	02790010	GASTRIC DISCOMFORT
4	ABC123	2	2	06MAY99	01540010	FEVER
5	ABC123	3	1	26JUL99	00240003	PRURITUS
6	ABC123	3	2	27JUL99	01090001	HEADACHE
7	ABC123	3	3	28JUL99	01090001	HEADACHE
8	ABC123	3	4	29JUL99	02050001	DIARRHEA
9	ABC123	3	5	30JUL99	02790011	INDIGESTION
10	ABC123	3	6	31JUL99	02040001	CONSTRICTION
11	ABC123	3	7	01AUG99	02050001	DIARRHEA
12	ABC123	3	8	02AUG99	02050001	DIARRHEA
13	ABC123	3	9	03AUG99	01090001	HEADACHE
14	ABC123	3	10	04AUG99	02280001	NAUSEA
15	ABC123	3	11	05AUG99	02280001	NAUSEA
16	ABC123	3	12	06AUG99	04550001	EDEMA

Figure 3

OBS	PROTOCOL	PATIENT	VISIT	ECGHR	ECGPR	ECGQRS	ECGQTINT	ECGQTC	ECGDATE
1	ABC123	1	1	84	200	80	420	395	05AUG99
2	ABC123	1	2	60	100	60	420	420	06AUG99
3	ABC123	2	1	90	100	80	380	441	27SEP99
4	ABC123	2	2	110	160	60	360	457	29SEP99
5	ABC123	3	1	90	140	60	320	392	04MAY99
6	ABC123	3	2	108	120	60	320	425	06MAY99
7	ABC123	3	3	85	110	40	340	410	05JUL99
8	ABC123	3	4	90	160	60	360	441	27JUL99
9	ABC123	3	5	79	200	60	380	420	04MAY99
10	ABC123	3	6	84	180	70	350	414	06MAY99
11	ABC123	3	7	88	100	77	327	391	04AUG99
12	ABC123	3	8	64	100	60	400	410	07AUG99
13	ABC123	3	9	72	140	60	380	416	06JUL99
14	ABC123	3	10	84	160	60	360	426	07JUL99
15	ABC123	3	11	60	160	60	400	407	04JUN99
16	ABC123	3	12	81	160	60	400	405	06JUN99
17	ABC123	3	13	65	160	60	380	395	15APR99
18	ABC123	3	14	63	160	40	380	389	17APR99
19	ABC123	3	15	61	140	80	400	400	19APR99

Figure 4

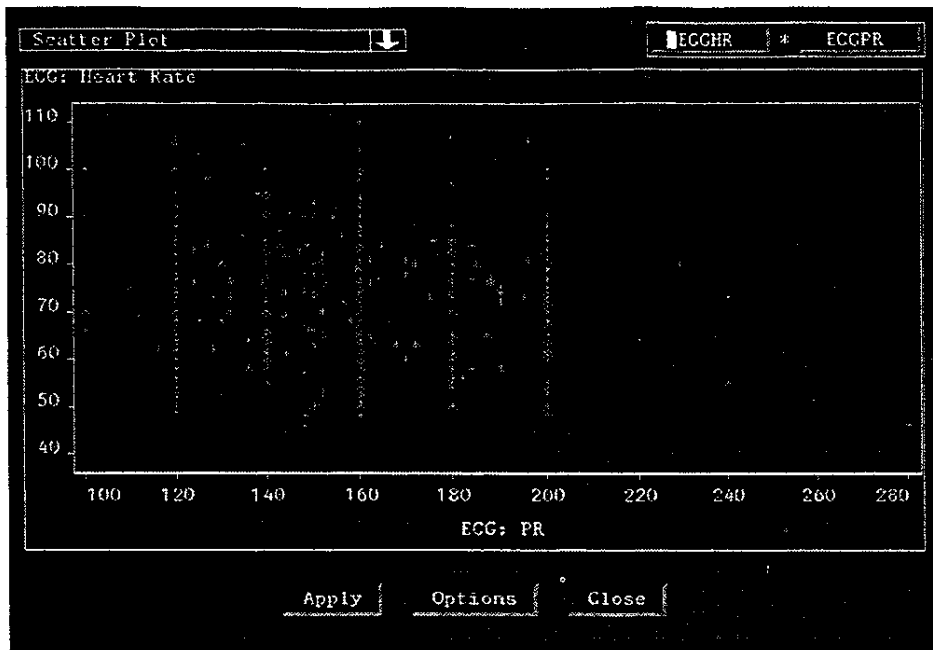


Figure 5

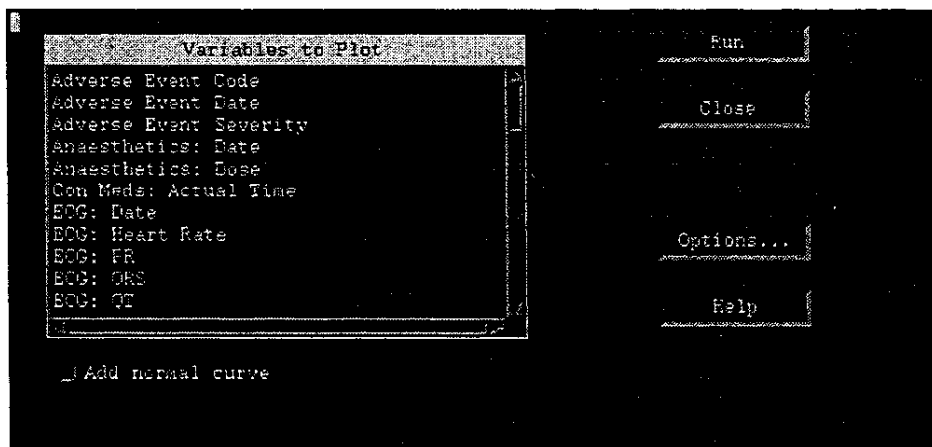


Figure 6

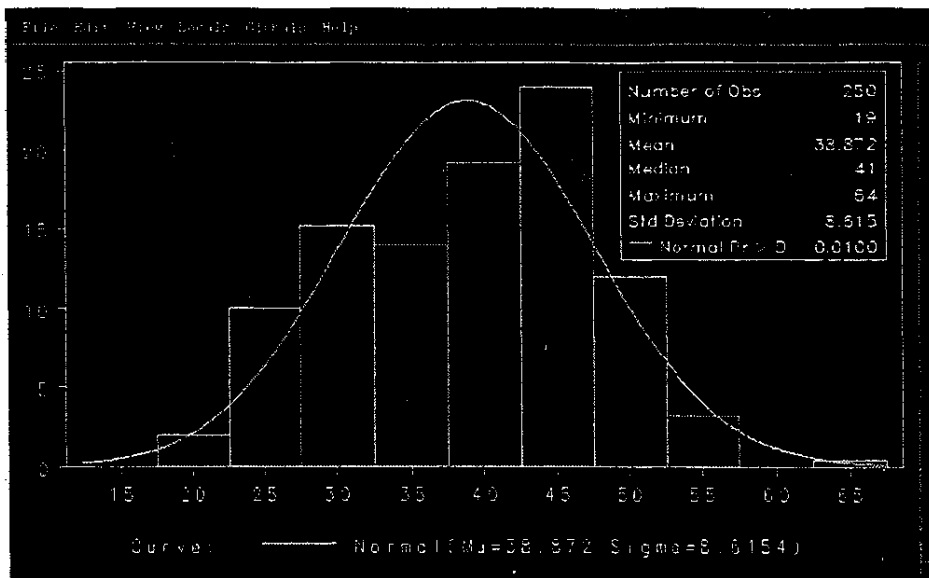


Figure 7

```

File Edit View Globals Help
-----
150 proc capability data= WORK_FTH01SD      graphics:      /* omit state */
151 /*
152 var AGE;                                /* list graphs */
153 histogram AGE                          /* plot histogram on var listed */
154 / normal(mu=0 color=blue)              /* add normal
distribution curve */
155 /*
156 /* with width of 2, color of blue */
157 width=gray                             /* color for bars
158 color=black                             /* color for axis
159 style=black                             /* color for text
160 fill=zebra                              /* fill pattern for bars
161 font=duplex                             /* font for text
162 haxis=median                             /* axis options for haxis
163 vaxis=axis2                             /* axis options for vaxis
164
165 inset                                  /* add inset table with stats
166 none min mean median max               /* number obs, minimum, median
167 sd (5.3)                               /* mean, maximum, std. deviation
168 normal(k=3) p(0.001)                   /* p-value for K-S normality test
169 / font=duplex                          /* font for inset stats
170 position=none                          /* location of inset stats box
171
172 /* values in parens are formats
173

```

Figure 8