SAS/PH-CLINICAL® SOFTWARE: STUDY STARTUP HINTS
J. Meimei Ma, Quintiles Inc.
Sandra D. Schlotzhauer, SAS Institute Inc.

INTRODUCTION
This tutorial presents assorted hints and techniques to help Product Administrators (PAS) get started smoothly with their first studies using SASIPH-Clinical software. We touch on suggestions for pre-processing considerations, HELPDEFS examples, tips for creating study definitions, backup hints, and training ideas. The advantages of planning ahead are emphasized.

The people who will learn the most from this tutorial are programmers or database managers who are, or intend to become, PAS. We assume that you have a working knowledge of clinical trials research data. General understanding of handling SAS datasets is expected. While only minimum experience with SAS/IPH-Clinical software is required to understand most of the topics, you may benefit more after gaining actual experience creating a study definition.

The techniques described apply to all installations using Release 6.08 or 6.09 of SAS/IPH-Clinical software. For simplicity, we refer to the source data for the clinical trial as clinical datasets and discuss examples that have SAS datasets as the clinical datasets.

PLANNING AND STARTUP
Although much of this tutorial stresses planning, this section focuses on planning and startup for a new Product Administrator. Before you begin to define your own study to SAS/IPH-Clinical software, we recommend:

- For Unix or other X Windows environments, your settings for the X resources will affect the appearance of the software. Refer to pages 7-8 of SAS Technical Report P-259 for a suggested set of X resources. Try out the recommended set of resources, and modify them as needed until they meet your needs. Then, you will be in a much better position to make recommendations for X resources for your users.
- Build the autoexec file. Although as a single user, you won't need this file, you do need to plan for the future when your users have access to the software as well. By building the autoexec file now, you can experiment with choices in it—and make decisions about what will best meet your users' needs.
- Check the documentation listing at the end of this tutorial. For Product Administrator tasks, you need both the SAS/IPH-Clinical Software: Administrator's Reference, Version 6, First Edition and SAS Technical Report P-259. The technical report contains new chapters, updates to existing chapters, and replacements to existing chapters. For example, to learn about defining studies, you should read Chapter 11, "Defining Studies" in P-259, not in the Administrator's reference manual or P-249.
- Gain a preliminary understanding of dataset and variable levels. These are introduced on page 93 of P-259. The impact that dataset levels have on patient group performance is explained in detail in Chapter 15 of the Administrator's reference manual; however, you may want to experiment with defining a study before reading this chapter.
- Define at least one of the sample studies. Two sample studies are shipped with the software. Using these studies to learn the interactive study definition process is an effective way to learn the study definition process without getting bogged down in any data issues specific to your site. Also, the software is shipped with a specially formatted flat files that contain the correct study definitions for these two studies. You can define the studies and then check your definitions by comparing them to the definitions obtained from using the FASTLOAD utility that reads the flat files.

PRE-PROCESSING ISSUES
The pre-processing phase is important for efficient use of SAS/IPH-Clinical studies. You should use this phase actively for planning, experimenting, and manipulating the clinical datasets. Potential benefits of thoughtful pre-processing include better performance, easier maintenance, and, most importantly, happier users. We discuss variable labels, formats, variable types, VIEWS, and the HELPDEFS program.

VARIABLE LABELS
Deliberately defined variable labels help your users to easily locate appropriate analysis variables. Although you can modify labels within SAS/IPH-Clinical software, thinking ahead allows you to create:

- consistent identifier labels
- final labels in clinical datasets
- optimal labels for all users.

Identifiers include variables that identify patients and visits, which are used to merge information, as well as variables related to investigators, sites, or visit dates. Typical variable names are PATNO, PID, PATNO, INVEST, SITE, and VISIT. Even when a consistent variable name is used, each identifier may be described in many ways. For instance, the labels for PATID might be "Patient Identification Number," "Patient," "Patient Number," or "Subject Number." Each of these labels is perfectly reasonable, but remember that labels matter as well as variable names when the list of unique variables is generated for a study definition. An advantage of creating consistent variable names and labels for identifiers in all clinical datasets is that you eliminate the need to resolve differences at study definition time.

In an ideal situation all variables would have perfect labels from the first time any data appears in a dataset. In situations where standardized CRFs exist, across protocols or across drug programs, you should strive for early labeling. Try to remind whoever is responsible for creating labels that eventually a variety of disciplines will work with those labels. For instance, if the data processing group defines labels in the data management system, then consider the value of asking the statistical and clinical groups to review draft labels before final implementation. When labels are not already
defined, or are insufficiently descriptive for your SAS/PH-Clinical software users, you should create or re-create them during pre-processing. If re-creating datasets is not appropriate, you may use the DATASETS procedure to add variable labels to existing clinical datasets.

While changing labels in a study definition is easy, in the long run you will benefit from updating labels in the clinical datasets instead. Suppose you had modified some variable labels in the study definition. Since the new clinical dataset does not contain any revised labels, you will end up repeating the changes. Also, more complex quality assurance issues arise when two versions of labels exist.

**Variable Formats**

Another pre-processing consideration is the formats for variables in the datasets. If you want users to be able to see and enter the formatted values when subsetting, you need both an informat and a format with the same name. For example, the DRUG variable may be stored in the dataset with values 1, 2, and 3; and a format $DRUG. converts these values to Placebo, Panacea, and Sasprin. If an informat named $DRUG. is not found, users see the values 1, 2, and 3 and must enter these values as well. If the informat is found, users see the values Placebo, Panacea, and Sasprin and can type in these values as well.

You need to plan the format search path. You can specify a different format search path for each study you define. Thus, you can use the $DRUG. format for different studies, and assign different values to the character codes (1, 2, 3, and so on) depending on the study. For this to be effective, you need to plan the format search path for the study. Consider which variables are formatted, which variables share a format, and what variables you are likely to omit from the users’ view. Do not worry about getting the search path exactly correct the first time since you can always modify it. However, by planning in advance, you can avoid unexpected problems once users start to work with the study.

**Variable Types & Attributes**

The selection of variable types and lengths is an important aspect of pre-processing. Keep in mind that given the variety of possible methods for organizing and storing clinical data, experimentation with your own study databases is required before the optimal choice becomes obvious.

For categorical variables, making them character has advantages. If users will use Explore, then character categorical variables automatically will be Nominal in the SAS/INSIGHT data window. Even when numeric variables are defined in a study definition as Categorical, as opposed to Analysis, they will appear as Ordinal variables in SAS/INSIGHT. Imagine the effort required to change twenty variables from Ordinal to Nominal. You may save storage space by making categorical variables character instead of numeric. For example, consider just twenty variables with values 1-5 for 2000 patients:

<table>
<thead>
<tr>
<th>Character $1</th>
<th>Numeric LENGTH=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>40,000 bytes</td>
<td>160,000 bytes</td>
</tr>
</tbody>
</table>

For completed studies, pre-processing long character variables that contain descriptive text may improve performance. For example, suppose CMNAME is 200. A programmed check reveals that the longest value is only 94. Think of the space savings in shrinking CMNAME to $95 or $100. Although SAS/PH-Clinical software allows compression for saving space, remember that performance implications may exist when using compressed clinical datasets.

Working with numeric variables brings out different issues. Using LENGTHs less than the default of eight will save space, but subtle rounding differences may result in ineffective searches. At least you should explore the source data during the pre-processing phase to fully understand the exact values that occur for all numeric variables. Descriptive statistics from the FREQ, UNIVARIATE, or MEANS procedures can prove helpful. For numeric variables with decimal places, use permanent formats in the clinical datasets to avoid values such as "1.999998." As with labels, you are allowed to change formats in the study definition, but should seriously consider the advantages of making such changes during pre-processing.

**DATA Step Views**

When you define a study, the software sometimes creates SAS DATA step views on the original datasets. One of the parameters of the software is that it does not write to the original clinical datasets. Thus, when you rename a variable as part of the study definition, the created view accomplishes the renaming without modifying the clinical datasets. Views are created when:

- variables are given a different label in study definition
- variables are renamed in study definition
- new variables are created in study definition

Ideally, you would change the original clinical datasets and revise the names or labels, and create new variables in the datasets. However, this is not always possible. Note that the views can receive the same sort of quality assurance validation that is typically performed on the original clinical datasets. The view source code is contained in the VW catalog for the study. By checking the date of the SOURCE entry, you can ensure that no modifications (new variables, or additional renaming or relabeling of variables) have been performed.

There may be a performance issue when DATA step views are created. The views add an extra step to the patient group process. In general this step is not noticeable to users, but may be noticed for very large datasets.

**Repeated Use of HELPDEFS**

The focus of this section is not on the mechanics of copying or using the program but on the strategies for using HELPDEFS. The HELPDEFS program is shipped with SAS/PH-Clinical software as an entry in the SASHELP.PHARMA catalog. To use this program, copy it to your private library as described on pages 124-125 of the Administrator’s reference manual.

Finding the Patient and Visit Identifiers

Suppose you do not know the patient or visit identifier variables. HELPDEFS can assist in identifying these variables. To do so, request the
repeated variables report by setting REPSRPT=Y in the sixth section of the HELPDEFS program. To minimize the output, set the requests for all other reports to N. This prints a report that displays the repeated variables in all the datasets for a study. The report is ordered by frequency so that the most-repeated variables appear first. Since every dataset for the study must contain the patient identifier, the first variable in the list is a likely candidate for the patient identifier. For example, in Figure 1, the PATID variable is the best candidate for the patient identifier. In this case, the variable name is an obvious clue but that is not always the case.

In some cases, the levels assigned may not be what you expected. To further examine the dataset levels, run the HELPDEFS program with ADDLEVEL=Y, FREQRPT=Y, and the patient and visit identifier variables identified. This produces a report that identifies the number of patients at each visit, and may point out problems in the data. The output in Figure 3 shows a sample report.

Identifying the Dataset Structure

Once you know the patient and visit identifiers, you can use HELPDEFS to identify the dataset structures. In fact, even if you think you know the dataset structures already, we recommend you use HELPDEFS to confirm them. Providing correct dataset structures as part of the study definition is vital to ensure correct results when creating patient groups. To use HELPDEFS to identify dataset structures, provide the patient and visit identifier variables in the HELPDEFS call and set DSSUMRPT=Y. This prints a listing like the one in Figure 2, where the last column gives the dataset level. Note that the level depends on the identifiers provided; if they are not supplied correctly, the dataset levels may not be correct.

First, the DEMOGP dataset contains one observation for each of the 100 patients and does not contain the visit identifier variable (VISIT). The MEDHIST dataset also contains one observation for each patient and does contain the VISIT variable—which has the value 0 for all patients. The LABS dataset contains one observation per patient at each visit and thus has level V. Notice that for visits 3 and 4, only 99 of the 100 patients have lab information. The EYES dataset is a VE dataset, with 2 observations per patient for most patient-visit combinations. You may want to examine the patients with only one observation at visits 1 and 3. Since the dataset is normalized to have one observation per patient-visit-eye combination, you would want to ensure that there were no missing records for patients with only one observation. The CONMED2 and ADVERSE2 datasets are also VE, with patients reporting multiple observations as needed at each visit. In these sorts of datasets, look for unusual values. For example, if the report contained a line "1 patient had 24 visits = 4" then you would want to investigate the records for the patient. The patient may, in fact, have reported 24 concomitant medications at visit 4; or, the patient may have reported far fewer concomitant medications and the number of records were inadvertently replicated.

Revising Variables

Another step in study definition is to ensure that the only two variables repeated in the clinical datasets are the patient and visit identifiers. The repeated variables report, shown earlier, assists with this process.
One problem that sometimes occurs is a variable with the same name--

**TERM**--character variable in each dataset. The VlSDATE variables within SASIPH-Clinical software or in the original data might appear in both the ADVERSE2 and CONMED2 datasets and is a character variable in each dataset.

By using this report and browsing the datasets, you can decide if the repeated variables contain different information in the different datasets. For example, the VISDATE variable is likely to differ in the different datasets, but the AGE, SEX, RACE, INVEST, and SITE variables are likely to contain the same information—and the same values for a given patient—in the different datasets. To eliminate repeated variables, you will need either to rename the VISDATE variables within SASlPH-Clinical software or in the original datasets. To eliminate the repeated instances of the other variables, the best solution is to delete the replicated variables from the original datasets during pre-processing. Or, you can delete the variables from all but one of the datasets as part of the study definition. This is explained in the administrators reference information on pages 96-114 of P-259.

**Restructuring Candidates and Patient Group Sizes**

The output from HELPDEFs also identifies candidates for restructuring. Briefly, reducing the number of Patient Event (PE) and Visit Event (VE) datasets in a study can result in performance gains in the patient group process and produce more compact patient groups. (For more detail on this issue, see Chapter 15 in both the PA reference manual and P-259.) After resolving issues with repeated variables, run the HELPDEFs program again with ADDLEVEL=Y, SIZEPT=T, and the patient and visit identifier variables identified. This produces an output similar to the one in Figure 4.

In this situation, there are no PE datasets, so the reduction in patient group size can occur only by restructuring the VE datasets to V. Little performance improvement is gained due to the small size of the datasets in the study. From the first part of the output, note that the three VE datasets are the smallest datasets in the study. In a more typical situation, the VE or PE datasets are among the largest datasets in the study. In that case, restructuring the PE and VE datasets will result in significantly larger values for Percentage Reduction in the output.

**STUDY DEFINITION ISSUES**

This section discusses several tips for study definition, including when to provide the general information, updating the default group names, using the checks for completeness, and building a custom screen.

**Providing General Information**

When defining a study, you generally want to provide the completed study definition to users as quickly as possible. The first step in defining a study is to provide general information about the study—the sort of information typically contained in the study protocol. Compiling this information may take time, so you may speed the definition process by defining only the required information. Of course, to help your users, you should later update the study and provide the general information. The information we recommend you provide at study definition falls into three categories: required, security group, and update dates.

**Required Information**

The only required fields for the general information for a study definition are the study name and the baseline visit. Users see the study name when building patient groups, so be sure to provide a unique name that clearly identifies the study. The baseline identifier is used internally; if you do not know the exact value in study definition, simply supply a best-guess value at this time. Later, update the definition with the correct value.

**Security Group Information**

The first screen of general information provides a field where you enter the security group for the study. If you plan to restrict access to the study to a limited group of users, enter the security group name. Note that the security group does not need to be defined at this point. However, you will want to define the security group and group membership before giving users access to the completed study. As always, planning ahead is the secret.

**Definition and Data Update Dates**

The first screen of general information also contains fields for the study definition update date and the study data update date. This information is not required but it can be very useful, both for you as an administrator and for your users.

The Study Definition Update Date identifies the last time the study was updated. At study definition, this is set to the current date. Later, you can manually update this date when you update the study. When users select saved patient groups for the study, the software checks the study update date. If this date occurs after the date when the patient group was saved, users receive a warning message that recommends they recreate the patient group. For example, suppose you update the study on 20MAY94 and enter the update date in the general information. Any user who accesses a saved patient group (for this study) that was created before 20MAY94 receives a message that the study definition has changed.
The Study Data Update Date is similar. If the study data is updated, then the saved patient group datasets that users have created may be incorrect. For example, suppose a user saves a patient group dataset for this study on 25MAY94. The study datasets are updated on 27MAY94. Any time the user accesses the saved patient group after 27MAY94, she receives a message that the saved patient group data may be out of date.

**Revising the Default Group Names**

When providing details on the datasets for a study, evaluate the default group names, which are set to the dataset names by the software. Unless your dataset names will clearly identify the groups to users, you should change the default names. A default group name can be up to 12 characters long and contain embedded blanks; dataset names are restricted to 8 characters and cannot contain embedded blanks. For example, the dataset AESEVR contains information on serious adverse drug reactions. Rather than AESEVR as a group name, you might consider using "AORs-Severe".

**Updating PATID and VISID Lists**

When providing details on the datasets for a study, you identify which datasets contain the complete list of patients and which datasets contain the complete list of visits. The datasets with the Patient ID list set to Yes are used to create the complete list of patients. When users create a patient group and select patients by patient ID, this list is displayed. When defining a study, all datasets initially have the Patient ID list set to Yes. Thus, the software searches through all datasets to construct the complete list of patient IDs. If a subset--or one--of the datasets contains the complete list of patients, leave the Patient ID list set to Yes for those datasets and change it to No for the remaining datasets. This results in better performance in the patient group process, since fewer datasets are searched.

Similarly, the datasets with a Visit ID list set to Yes are used to construct the complete list of visits. When users select visits in the Analysis component, the software searches through all datasets with the Visit ID list set to Yes. When defining a study, all datasets initially have the Visit ID list set to No. This differs from the Patient ID list since all studies must contain a patient ID, but are not required to contain a Visit ID. If a subset of the Visit (V) or Visit Event (VE) datasets contains the complete list of visits, set the Visit ID list to Yes for those datasets and leave it set to No for the remaining datasets. The safest course of action is to set the Visit ID list to Yes for all V or VE datasets. However, this action also requires the software to search through multiple datasets. Reducing the number of datasets involved in the search results in an increase in performance.

**Using the Checks for Completeness**

When you have provided all the information for the study definition, the software performs a number of checks for completeness. If all checks are successfully passed, the study is marked as complete, and users can use it to create patient groups. If one or more checks are not passed, the software displays a message window with details on the problems and marks the study as incomplete.

As you define the study, the software performs checks at each step in the process. For example, when providing information on the variables, the software checks that the attributes of all variables are either A, C, or N (since these are the only valid choices for the attribute). In addition to these steps, the overall checks for completeness are:

- Has a visit identifier been assigned when the study definition does not contain any V or VE datasets?
- Does the visit identifier variable appear in a Patient (P) or Patient Event (PE) dataset?
- Have all datasets been given valid organizational levels (P, PE, V, or VE)?
- Do any variables other than the patient and visit identifier variables appear in more than one dataset?
- Do all computed variables have a list of components?

If any one of these checks fails and you force the study definition to be complete, users will receive unpredictable--and possibly incorrect--results when creating a patient group. As one example, suppose the variable DATE appears in both the DEMOG and ADR datasets. If users choose other variables from both those datasets when creating a patient group, the resulting patient group contains a single variable DATE, and users will not know which dataset the value for DATE is from. To solve this problem, use the variable in both datasets, to DATEDMG and DATEADR, for example.

The key to the checks for completeness is that they point out problems with the study definition, and these problems should be resolved before giving users access to the study. After updating the study definition, you can re-run the checks for completeness. If all checks are successfully passed, then you can mark the study definition as complete. In addition, any time you make significant changes in a study definition, you should re-run the checks for completeness. This functionality is available in the Other tasks window in the Admin: Study List window.

**Building Test Patient Groups**

Okay, so now you have a study definition completed, what next? The answer is not to build a patient group with all patients and all variables! You should approach this testing process like any other complex programming task: start small, follow planned logical steps, and document as you go. Remember that an important part of PA's responsibility is to support users after the study definition is completed and released. You will learn valuable lessons about using the data defined in SAS/IPH-Clinical software as you build increasingly complex patient groups.

Suppose we have defined a new, relatively small study, only 100 patients. In this case, using a subset of patients during initial testing is not important. The data are organized as follows:

- **P** Demographics
- **V** Efficacy parameters
- **V** Lab results
- **PE** Adverse events
- **VE** Concomitant medications
- **P** Termination variables

1411
Here is one way to get started. Remember we want to test the study definition, as well as ensure that users will find the performance of the system acceptable. We would build a variety of patient groups, which include all patients, by selecting the following combinations of variables:

1. identifiers and demographics (P)
2. primary efficacy variables (V)
3. demographics, termination, and all efficacy (P,V)
4. adverse events (PE)
5. lab results (V)
6. concomitant medications and AEs (VE,PE).

The final step for the first stage of testing is to work with these patient groups as a user. The entire process is iterative as problems and issues are discovered. We may need to return to the pre-processing phase and/or redo parts of the study definition in order to produce the best study.

For your own studies, after building each test patient group, experiment with atlas patients is practical. Will the dataset be too large for the PA learns during testing process should lead to documentation for users who are allowed to build their own patient groups.

Creating a Custom Screen

By providing a custom screen, your users can browse a patient group either using the default screen or one that organizes the variables more effectively for review purposes. In addition, they can drill down to either the default screen or the custom screen. Some hints for creating a custom screen are given below:

- Design the custom screen with users in mind. Since each custom screen is associated with a clinical trial, find out who the primary users are for that trial. Ask these users for input. Which variables are most important? Which variables do they want to see first? Which variables must appear on the first page of a multi-page screen? Which variables are unimportant?
- Look at the CRFs for the study. Since one goal is to design the custom screen as a better match with the CRFs, look at how information is grouped in the CRF. As an example, if your users tell you that demographic information is important and must appear near the top of the first page, the CRF will show how that information is organized. Communication with your users is important here as well—they may not like the CRFs and you can change the organization to better suit your users’ needs.
- Consider displaying only a subset of variables. The custom screen need not display all variables for the study. Users can always view all the variables for the current patient group by browsing records with the default screen. If users generally do not build a patient group with all variables, consider supplying a custom screen that displays only commonly-requested variables. Again, communication with your users will help you design a screen that displays the most important information first, and secondary information later—if at all.

- On personal computers (PCs), set the workspace size both when creating the custom screen and when invoking SAS/IP-Clinical software. This ensures that the screen you develop appears in the same way for all users. The simplest—and best—approach is to maximize the size. To do so, use the AWSDEF invocation option. This option is executed as SAS is being invoked. The syntax to maximize the size is: -AWSDEFS 000100 100.

As a final, but important point, you should create the custom screen outside of SAS/IP-Clinical software and then move it to the PHADMIN.USRSORT catalog when completed. Creating the custom screen with the FSEDIT procedure is essentially a programming effort, and it is best done in the familiar programming environment of the SAS Display Manager System (DMS). Here is a suggested strategy to follow:

Preparation:

1. In SAS/IP-Clinical software, note the study code in the Admin: Study List window. Select the study and then select the Other menu.
2. When the Other Tasks window displays, select Produce Study Report.
3. In the Admin: Study Report window, make choices to run the Variables report interactively and then select OK.
4. When the report is generated, we recommend you both print it and save it. These selections are available from the Other menu.
5. If you plan to create a custom screen with most of the variables from the study, you may want to create a patient group that contains those variables. This is a judgement call. Creating the patient group will facilitate design of the screen, but a very large patient group may take time to run.
6. Exit to DMS.
7. Create a SASUSER.TEMP directory. This is a permanent SAS directory and will be saved across SAS sessions. Do not develop the custom screen in a WORK directory, as the information will not be retained across sessions.
8. If you have not created a patient group, create a temporary dataset to be used with FSEDIT. The information in this dataset is not important; some suggested code is shown below, but you can use any code that creates a small dataset. Ideally, the variable names should differ from those in your study.

```
data temp;
  a=1;
  output;
  run;
```

Creating the Screen

9. On a command line, type:

```
fsedit temp sasuuser.temp.work<XXX>.screen
```

and press ENTER. Replace <XXX> with the study code for your study. By using study codes, you can work on multiple custom screens in this directory and easily identify which screen is associated with each study. This command starts the FSEDIT procedure on the TEMP dataset and creates the SCREEN entry.
10. In the FSEDIT window, type MOD on the command line, and then select 2 in the list that appears. You can now modify the screen. We recommend you turn line numbering on to assist with screen design. Use the output from the Variables report to assist you with screen design. This report shows the variable name, type (character or numeric), label, and length.

11. Save the screen occasionally as you work. Not only does this ensure you save your work, but it provides you with an opportunity to check the work you have completed.

12. When you are satisfied with the completed screen, type CAT PHADMIN.USRSCRN on a command line and press ENTER. This displays the catalog that contains the custom screens. In the CAT window, type

COPY sasuser.temp.work<AXXX>.screen
<AXXX>.screen

where <AXXX> is replaced with the study code. The custom screen will now be available to users.

13. As a final check, create an appropriate patient group from the study. This patient group should include all the variables displayed in the custom screen. In the Browse component, choose to browse patient records with a custom screen. Check to ensure that the screen appears as you would like.

Finally, we recommend you follow the same procedure when modifying a custom screen for a study. Copy the existing screen to a permanent SAS library, modify it, and then copy it back to the PHADMIN.USRSCRN catalog. To avoid the need to re-create a previous custom screen for a study, we recommend you store the old versions of a custom screen either in the PHADMIN.USRSCRN catalog or in another catalog. If you store the backup custom screens in the PHADMIN.USRSCRN catalog, be sure to use names that will not automatically be used for other studies. One solution is to use the study name followed by an extension such as BK; for example, A320BK.SCREEN.

CHECK THE FORMAT SEARCH PATH

As mentioned before, a study can have multiple format libraries. If you have defined multiple format libraries to the product, check to ensure that the format search path gives users the formats they expect. If you don’t define a search path, the software uses the default format search path. This is defined at startup and may not give users what they expect.

To define the format search path, choose the study in the Admin: Study List window and then choose the Other menu. In the Other Tasks window, choose Update Format Lookup Table. In the window that displays, you provide the study code, physical name, catalog name, and sort order for each format library. You can check this or change it at any time.

BACKUP AND SELF-PROTECTION

When in doubt, make a backup. Sound familiar? Have you done so lately? If not, take the time to read about FASTDUMP and FASTLOAD. We will assume that your clinical datasets are being backed up automatically. The other aspects of using SAS/PH-Clinical software that cry out for backup procedures are the study definitions and the datasets in PHADMIN.

USING FASTDUMP

You can benefit from using FASTDUMP to back up a study definition in a variety of situations. Dumpfiles are useful backups for the following reasons:

- Testing study definitions
  - Work in progress
  - Experimenting
- Production study definitions
  - Completed
  - Updating

Since creating a study definition for a complex trial may take more than a few minutes, after doing a substantial amount of work; protect yourself by using FASTDUMP. The interactive study definition process allows you to create a study definition in stages. Once you are almost done, but would like to test alternatives, saving the current study definition as a dumpfile can give you confidence to experiment with new options.

Once a study definition is completed, maintaining a current dumpfile is the best backup strategy. A dumpfile allows you to re-create a damaged study, or move one to another platform or site, or to quickly build a new study with a similar structure. As a new PA, become comfortable using FASTDUMP and FASTLOAD before releasing the first study to your users. You should define a standard naming convention and location for dumpfiles. We suggest creating a protected directory to store these files; do not just put them in the PHADMIN directory.

Once you have a dumpfile, you may want to make changes there instead of using the SAS/PH-Clinical software interactive method. Be careful! Read the documentation very carefully before changing anything. People often forget that unlike a SAS program, a dumpfile is case sensitive. Also remember to put any text changes in quotes. When you have planned your changes and are ready to start ... have you made a backup of the dumpfile?

BACKUP YOUR PHADMIN LIBRARY

In addition to backing up the study definitions with the FASTDUMP utility, you should consider backing up other items in your PHADMIN library. The simplest approach is to backup the entire library. This includes datasets and catalogs for:

- general study information
- each study. (This information is also included in a FASTDUMP file for the study.)
- the Public Library, which includes saved patient groups, saved patient group datasets, saved programs, saved output, saved graphs, and saved notes.
- items in the request queue for the Public Library. (Generally, you will want to process any items in the request queue before performing the backup.)
users and security groups for users
- devices, which includes monitors, printers, and plotters
- batch setup
- defining custom menu systems accessed from SAS/PH-Clinical software.

If, however, you want to back up only selected datasets or catalogs from the PHADMIN library, Figure 5 shows which datasets and catalogs are associated with each general PA task.

Whether you choose to back up the entire PHADMIN library or only a portion of it, be sure to store the backup copy in a different directory. On a PC with multiple hard disks, you may want to save the backup copy on a different hard disk.

**Hints for Single-User PCs**

In the special case of single-user PCs, you know the user who will be working with SAS/PH-Clinical software on that particular PC. This provides you with the ability to customize the software to a certain extent. It can be helpful not only for individual users at your company, but as part of a CANDA for a given FDA reviewer.

**Providing the Userid on Startup**

For example, on PC hosts the initial window in the software is one where users specify their userid and password. (This is necessary on PC hosts because the software cannot obtain the userid from the operating system, as it can with other hosts such as MVS, VMS, CMS, and Unix.) Since you know the userid and can obtain their SAS/PH-Clinical password, you can configure the software so that this information is provided in the autoexec file. This saves users a step on startup, and the main window for SAS/PH-Clinical appears. To do so, add the following statements to the autoexec file:

```plaintext
%let phjobid = <userid> ;
%let phpasswd = 0 ;
```

where the `<userid>` is the user's id for SAS/PH-Clinical software. This provides the userid information to the software.

**Avoid the Superuser ID**

The software is shipped with the SUPERUSER id predefined so that you as a Product Administrator can use this id on startup. When configuring a single-user PC, it is important to define an id for the user. The SUPERUSER id gives Product Administrator functionality, which you generally do not want to provide to end-users since PA functionality enables users to change the study definitions.
BUILDING A PATIENT GROUP ON STARTUP

Since you know the user, you may also know which study they work with the most. In addition, the user may have a patient group that they typically build when invoking the product. For example, suppose the user is focusing on the Panacea study and starts the SAS/IP-Clinical session by building a patient group with all patients and all variables for the study. You can perform this task for the user by performing the following steps:

1. Ask the user the name of the patient group they want built when SAS/IP-Clinical starts.
2. In SAS, define a LIBNAME statement that references the user’s PHUSER library. The libref for this statement is not important. However, if you have exited from SAS/IP-Clinical software to the Display Manager System, do not use PHUSER as the libref for the user’s library. The PHUSER libref is already assigned to your private library, and overwriting this assignment can cause problems.
3. View the user’s PHMLEST dataset with the FSVIEW or FSBROWSE procedure. Find the patient group that the user wants built on startup and write down the values for the following variables: STDYCODE, NAME, DESC, ECODE, DATE, and TIME.
4. On the user’s PC, add the following statements to the autoexec file for SAS/IP-Clinical software:

   ```
   %let stdycode = <stdycode> ;
   %let viewname = <name> ;
   %let viewdesc = %str(<descr>) ;
   %let viewcode = <ecode> ;
   %let viewdate = <date> ;
   %let viewtime = %str(<time>) ;
   data work.phdata(compress=no protect=ABC123) ;
   set phuser.d<ecode without leading PR> ;
   run ;
   data work.varlist(compress=no protect=ABC123) ;
   set phuser.v<ecode without leading PR> ;
   run ;
   ```

   where the values in <> are those values you wrote down from the user’s PHMLEST dataset.

As an example, consider a patient group for the Panacea study with all patients and all variables for the study. The following code builds the patient group on starting SAS/IP-Clinical software:

   ```
   %let stdycode = A011 ;
   %let viewname = Panacea ;
   %let viewdesc = %str(Panacea: All Patients, All Vars) ;
   %let viewcode = PRO000002 ;
   %let viewdate = 06JAN94 ;
   %let viewtime = %str(11:00) ;
   data work.phdata(compress=no protect=ABC123) ;
   set phuser.d000002 ;
   run ;
   data work.varlist(compress=no protect=ABC123) ;
   set phuser.v000002 ;
   run ;
   ```

   In this example, the patient group name is PANACEA, and the ECODE value is PRO000002. Notice that the ‘PR’ for the ECODE value is omitted in the SET statements for the PHDATA and VLIST datasets.

RESOURCES FOR NEW PAs

New PAs can learn from a number of resources. The written documentation includes:

- SAS/IP-Clinical manuals (see the “References” section)
- Usage Notes
- Alert Notes
- Update memos from SAS Institute developers.

The Alert Notes and update memos are distributed with production and maintenance releases of the software. These are sent directly to PAs.

As you investigate your first studies, remember to explore the online Help facility along with printed reports on security groups, study definitions, and so on. HELPDEFS output is another source of information. After reading about the PHADMIN library, consider browsing a few of the datasets.

THE PAYOFF

The variety of topics we covered in this tutorial demonstrates that the startup process involves many aspects. Planning ahead and allowing time for experimentation and documentation is an important time investment. You will enjoy the payoff when you find yourself supporting users with well designed study definitions and fewer questions.

---

**Study Startup**

**Plan, Experiment, Document**

**For Happier Users**

---

1415
REFERENCES

MANUALS

The manuals for Product Administrators for SAS/IPH-Clinical software are:


Both manuals are needed. The technical report updates information from the reference manual and provides new information, but it does not replace the manual. The manuals for all users for SAS/IPH-Clinical software are:

SAS/IPH-Clinical® Software: Your Clinical Data Review System, Version 6, First Edition


Both the reference manual and P-248 are needed to provide complete user reference information. Additional information is available in the host companions for specific operating systems and in manuals for related products. As a Product Administrator, the manuals you will almost certainly need are those for SAS/FSP and SAS/INSIGHT software. If your users will exit to SAS/ASSIST software, those manuals will be helpful as well.

RECOMMENDED READING


CONTACT ADDRESSES

Dr. J. Meimei Ma
Quintiles
P. O. Box 13979
Research Triangle Park, NC 27709

Sandra D. Schlotzhauer
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513

SAS, SAS/ASSIST, SAS/FSP, SAS/INSIGHT and SAS/IPH-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.