Tools to Facilitate the Creation of Pooled Clinical Trials Databases

Patricia Majcher, Johnson & Johnson Pharmaceutical Research & Development, L.L.C., Raritan, NJ

ABSTRACT
Data collected from individual clinical trials offer critical information. Of perhaps even greater importance is information that can be gleaned when data from individual trials are pooled. Such pooled databases can be used for several purposes: to answer health authority questions more efficiently, to support publications, and for data mining. However, data from different studies, particularly for long-term drug projects, may utilize different data model structures. Mergers between companies can also introduce variability to a company’s data model standards. At Johnson & Johnson Pharmaceutical Research & Development, L.L.C., the Data Warehousing department supports several SAS® macros, which can be used to facilitate the creation and verification of pooled databases. These macros utilize Base SAS® and SAS® Macro Language and can execute on both UNIX and Windows. This paper will present an overview of these macros.

INTRODUCTION
Pharmaceutical companies have a potential treasure trove of information contained within their clinical trials databases. In order to enable knowledge discovery, pooled databases are created from individual trials. Pooled databases facilitate the defense of current indications relative to health agency questions and help uncover additional indications for future study. These databases can also be used as the source for Integrated Summaries of Safety and Efficacy.

However, for many reasons, data from individual trials may not utilize the same data structure. In a long-term project, the case report forms used to collect the data may change over time, causing differences in code lists. Companies may not have—or may not enforce compliance with—standard data models. Mergers between pharmaceutical companies generally result in changes for at least one of the companies involved. Another factor to consider is the growing acceptance of the Clinical Data Interchange Standards Consortium (CDISC) standards, including recent endorsements by the Food and Drug Administration. Companies complying with CDISC standards will face a new data model to which legacy data may need to be converted.

Clinical trials data must be made consistent (ie, standardized to a single data model) before they are pooled. Otherwise, the resulting database will not provide meaningful information. At Johnson & Johnson Pharmaceutical Research & Development, L.L.C., one of the Data Warehousing department’s objectives is to support the conversion of legacy data to a new standard structure. To achieve that goal, the department supports two sets of SAS macros.
1) The first set, the Investigation Macros, provides a systematic and organized way to examine the original data in preparation for its conversion to a standard data model.
2) The second set of macros, the Conversion Macros, actually transforms the original data to the target—or new—data structure.

Both sets of programs utilize Base SAS and SAS Macro Language; the Macros execute on both UNIX and Windows.

After the data from different trials have been made consistent, the data can be combined into a pooled database. The Investigation and Conversion Macros can both be used to verify the resulting database.

INVESTIGATION
The following examples illustrate why it is important to ensure consistency in a pooled database.

1) Code lists
Suppose Trial 1 uses the following code list for adverse event severity
   1 = MILD
   2 = MODERATE
   3 = SEVERE
while Trial 2 uses this code list:

1 = MILD  
2 = MODERATE  
3 = MARKED

A user subsetting the data for SEVERE adverse events would inadvertently omit the MARKED events, due to the differences in the decodes.

2) Variable Names

a) Different variable names can be used to represent the same data attribute in different trials. For example, Trial 1 uses the variable AEONSET to represent the date an adverse event occurred. However, Trial 2 uses variable name AESTART.

b) Conversely, the same variable name can be used to represent different data attributes. In Trial 1, the variable CHGBASE represents the change from baseline at Day 28, according to its protocol. However, in Trial 2, the protocol calls for change from baseline to be calculated at Day 14. The difference in meanings for this variable would cause issues in the analysis.

The above examples demonstrate how data may not be pooled in a useful manner if they are not uniform. If the data are not consistent within the pooled database, then each user of the data must make the required transformations in every analysis and reporting program. This would be inefficient and could lead to inconsistencies and errors.

Once the decision has been made to transform the original data – also called the source data – to a new data structure, the source data should be investigated for consistency in two major areas.

- Are the data sets themselves consistent? Have the same data set names and labels been used?
- Are the variables within the data sets consistent in terms of code lists and attributes such as variable name, label, type, and length?

The Investigation Macros currently supported by the Data Warehousing department facilitate this data inspection.

Two macros comprise the Investigation Macros. The first macro queries the DICTIONARY.COLUMNS metadata and extracts the attributes of all columns in the specified data sets. For each variable, it runs a PROC FREQ and extracts up to 20 values for the variable. The variable attributes and values for all data sets are then saved to a SAS data set. In general, this macro is executed once for each protocol to be investigated. The saved SAS data set contains the results for all protocols. The second macro generates a listing using the data set created by the first macro. The listing contains the protocol number; data set name; variable name, label, type, and length; and variable values or decodes/formats for all records in the data set.

Macro parameters provide flexibility for the user, including the ability to:

- Specify the location of the input data sets.
- Create a temporary or permanent data set from the first macro.
- Examine multiple data sets simultaneously.
- Control the number of values per variable that are saved in the data set and displayed on the detailed listing.
- Read data that reside on a platform different than the one where the Macros are executing.
- Include only certain variables or, conversely, exclude certain variables from the outputs.
- Specify the name of the variable that represents the protocol identifier (this is needed when the incoming data set contains multiple protocols).
- Control the line size and page size of the outputs.

The user creates a program to invoke the Macros, one program for each type of data set (e.g., adverse events or demography) to be investigated. The Macros can examine multiple data sets simultaneously; there is no pre-defined limit to the number of protocols that can be reviewed in a program. The outputs produced are:

- For each protocol, a PROC CONTENTS of each source data set.
- For each protocol, cross-tabulations of codes and decodes.
- A listing of each protocol and its source data sets that were examined by the Investigation Macros.
- For all protocols combined, cross-tabulations of codes and decodes. See Figure 1 below.
- A detailed listing of each variable in the source data sets with the following information: variable name; protocol identifier; data set name; variable label, type, and length; and up to 20 variable values. See Figure 2.

The next two figures show parts of the outputs produced by the Investigation Macros, examining demography data for three protocols. Among the variables in the demography data set are two variables for sex, one representing the coded value (SEX) and the other representing the decode or formatted value (SEXF). As can easily be seen from the simple example in Figure 1, the three protocols have different values for variables SEX and SEXF. Protocols 101 and 202 are consistent in coding males as 1 and females 2; however, their decodes differ. Protocol 303’s decodes
match 101’s, but the codes differ.

Figure 1: Cross-tabulations for all protocols combined

<table>
<thead>
<tr>
<th>Code</th>
<th>Decode</th>
<th>Count</th>
<th>Protocol</th>
<th>Data Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>MALE</td>
<td>5</td>
<td>P303</td>
<td>DEMO</td>
</tr>
<tr>
<td>1</td>
<td>FEMALE</td>
<td>5</td>
<td>P303</td>
<td>DEMO</td>
</tr>
<tr>
<td>1</td>
<td>M</td>
<td>5</td>
<td>P202</td>
<td>DEMOG</td>
</tr>
<tr>
<td>1</td>
<td>MALE</td>
<td>5</td>
<td>P101</td>
<td>DEMOG</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>5</td>
<td>P202</td>
<td>DEMOG</td>
</tr>
<tr>
<td>2</td>
<td>FEMALE</td>
<td>5</td>
<td>P101</td>
<td>DEMOG</td>
</tr>
</tbody>
</table>

Figure 2 presents the detailed listing of all variables to illustrate other potential inconsistencies in the source data. Here, in addition to the SEX and SEXF variables seen in the previous output, variables AGE, RACE, and RACEF are also present. RACE and RACEF are the coded and decode values of race, respectively. In this listing, the user can clearly see where variable labels and lengths differ. The user can also see some of the same differences in values that were displayed in Figure 1. Values for all variables are displayed in the last column of the listing (Source Decode/Format).

Figure 2: Detailed listing of variable attributes and values

<table>
<thead>
<tr>
<th>Source Variable</th>
<th>Source Protocol</th>
<th>Source Data Set</th>
<th>Source Name</th>
<th>Source Label</th>
<th>Source Type</th>
<th>Source Length</th>
<th>Source Decode/Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGE</td>
<td>P101</td>
<td>DEMOG</td>
<td>AGE</td>
<td>NUM</td>
<td>8</td>
<td>22,44,46,54,55,58,65,66,72,78</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P202</td>
<td>DEMOG</td>
<td>AGE</td>
<td>NUM</td>
<td>8</td>
<td>22,38,44,46,54,62,66,74,82</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P303</td>
<td>DEMO</td>
<td>AGE</td>
<td>NUM</td>
<td>8</td>
<td>17,34,48,51,55,58,65,68,72,75</td>
<td></td>
</tr>
<tr>
<td>RACE</td>
<td>P101</td>
<td>DEMOG</td>
<td>RACE CODE</td>
<td>NUM</td>
<td>8</td>
<td>0,1,2,3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P202</td>
<td>DEMOG</td>
<td>RACE CODE</td>
<td>NUM</td>
<td>8</td>
<td>1,2,3,4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P303</td>
<td>DEMO</td>
<td>RACE (CODE)</td>
<td>NUM</td>
<td>8</td>
<td>1,2,3</td>
<td></td>
</tr>
<tr>
<td>RACEF</td>
<td>P101</td>
<td>DEMOG</td>
<td>RACE</td>
<td>CHAR</td>
<td>5</td>
<td>ASIAN,BLACK,OTHER,WHITE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P202</td>
<td>DEMOG</td>
<td>RACE</td>
<td>CHAR</td>
<td>5</td>
<td>ASIAN,BLACK,OTHER,WHITE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P303</td>
<td>DEMO</td>
<td>RACE</td>
<td>CHAR</td>
<td>5</td>
<td>BLACK,OTHER,WHITE</td>
<td></td>
</tr>
<tr>
<td>SEX</td>
<td>P101</td>
<td>DEMOG</td>
<td>SEX CODE</td>
<td>NUM</td>
<td>8</td>
<td>1,2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P202</td>
<td>DEMOG</td>
<td>SEX CODE</td>
<td>NUM</td>
<td>8</td>
<td>1,2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P303</td>
<td>DEMO</td>
<td>SEX (CODE)</td>
<td>NUM</td>
<td>8</td>
<td>0,1</td>
<td></td>
</tr>
<tr>
<td>SEXF</td>
<td>P101</td>
<td>DEMOG</td>
<td>SEX</td>
<td>CHAR</td>
<td>6</td>
<td>FEMALE,MALE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P202</td>
<td>DEMOG</td>
<td>SEX</td>
<td>CHAR</td>
<td>2</td>
<td>F,M</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P303</td>
<td>DEMO</td>
<td>SEX</td>
<td>CHAR</td>
<td>6</td>
<td>FEMALE,MALE</td>
<td></td>
</tr>
</tbody>
</table>

Both of the above figures highlight the cross-study inconsistencies and help the user to define the transformations needed in order to pool these three protocols into a useful database. Once the transformations have been identified, the user is ready to convert the data.

CONVERSION

There are a number of ways to convert the data; the most appropriate method depends on the particular situation. The Conversion Macros are a good choice if the transformations required consist primarily of variable attribute changes and recoding of categorical variables.
There are two main programs for the Conversion Macros, one to generate the conversion code and the second to execute verifications. The functionality of the Conversion Macros includes:

- Renaming source variables to the target data model.
- Applying formats to source categorical variables to recode values to match the target code lists.
- Combining multiple source data sets to create a single target data set.
- Executing built-in checks to flag inconsistent or potentially invalid mappings.
- Producing output that documents the mapping and lists the converted data.
- Creating a permanent SAS file of the code used to convert the data. This file can be modified and reused, as might be needed for special cases where user intervention is required.

Like the Investigation Macros, the Conversion Macros provide macro parameters for added flexibility. Parameters enable users to:

- Specify the location of the mapping table, input data, output data, and the reports and conversion code generated.
- Specify the data model of the source data set.
- Read data that reside on a platform different than the one where the Macros are executing.
- Write data to a platform different than the one where the Macros are executing.
- Indicate whether the conversion code that is generated is executed immediately or created and saved for later execution.

Several files are required to execute the Conversion Macros.

- A mapping table, which shows how each source variable maps to its target variable.
- PROC FORMAT code that provides data set labels and, if needed, formats that can be used to recode categorical variables.
- A program that initializes the relevant parameters and invokes the Conversion Macros.

Standard mapping tables and format programs have been created for users to copy and adapt for their individual studies. If there are several, similar trials to be converted, then the mapping table can be reused. However, as it is rare that a study follows a standard exactly and that any two studies are identical, users need to review the mapping tables and format programs with each use.

Figure 3 shows a mapping table created to convert two types of data: demography and adverse events. The mapping table is a text file; its values are read using column input. The first line presents the column headings. The first three columns present information about the source data: the data set name, the variable name, and, if applicable, the variable’s conversion format. The user would specify in the third column the format needed to recode categorical variables. The next five labelled columns are information about the target data: the data set name and the variable’s name, type and length, format, and label. There is a column without a heading between new data set and variable name: the value in this column denotes whether the variable is required (R) or optional (O). Multiple data sets can be referenced in a single mapping table, as in Figure 3. The first several lines after the headings represent demography; information for adverse events is presented after the dotted line.
Figure 3: Mapping table

<table>
<thead>
<tr>
<th>old_ds</th>
<th>old_vr</th>
<th>old_fmt</th>
<th>new_ds</th>
<th>new_vr</th>
<th>new_lg</th>
<th>new_fmt</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>all</em></td>
<td>protocol</td>
<td><em>all</em></td>
<td>R studyid</td>
<td>$18</td>
<td>Study ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>all</em></td>
<td>subjid</td>
<td>z8.</td>
<td><em>all</em></td>
<td>R subjid</td>
<td>$8</td>
<td>Subject Number</td>
<td></td>
</tr>
<tr>
<td><em>all</em></td>
<td><em>all</em></td>
<td></td>
<td><em>all</em></td>
<td>R usubjid</td>
<td>$31</td>
<td>Unique Subject Id</td>
<td></td>
</tr>
<tr>
<td>demog</td>
<td>age</td>
<td>kdemog</td>
<td>R age</td>
<td>8</td>
<td>Age</td>
<td></td>
<td></td>
</tr>
<tr>
<td>demog</td>
<td>racef</td>
<td>kdemog</td>
<td>R race</td>
<td>$9</td>
<td>Race</td>
<td></td>
<td></td>
</tr>
<tr>
<td>demog</td>
<td>race</td>
<td>race.</td>
<td>kdemog</td>
<td>R racce</td>
<td>8</td>
<td>Race Code</td>
<td></td>
</tr>
<tr>
<td>demog</td>
<td>sexf</td>
<td>kdemog</td>
<td>R sex</td>
<td>$6</td>
<td>Sex</td>
<td></td>
<td></td>
</tr>
<tr>
<td>demog</td>
<td>sex</td>
<td>kdemog</td>
<td>R sexc</td>
<td>8</td>
<td>Sex Code</td>
<td></td>
<td></td>
</tr>
<tr>
<td>demog</td>
<td>siteid</td>
<td>kdemog</td>
<td>R siteid</td>
<td>$3</td>
<td>Site</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| ae      | aestn   | kae     | R aestdt  | 8      | date9. | Start Date of Event          |
| ae      | aeseq   | kae     | R aeseq   | 8      |        | AE Sequence Number           |
| ae      | aeactc  | aaeactc.| kae       | R aeacttrc| 8    | Action Taken Code            |
| ae      | aeact   | $aeact.| kae       | R aeacttrt| $40  | Action Taken                 |
| ae      | aebody  | kae     | R aebodays| $100  | Body System                    |
| ae      | aecode  | kae     | R aecode  | $11    | AE Dictionary Code            |
| ae      | aepref  | kae     | R aedecod| $40    | Preferred Term                |
| ae      | aeincl  | kae     | R aedecod1| $40   | Included Term                 |
| ae      | aeenn   | kae     | R aeendt  | 8      | date9. | End Date of Event            |
| ae      | aeendt  | kae     | R aeendtc | $10    | End Date of Event (char)      |
| ae      | aeout   | kae     | R aeout   | 40     | Outcome of Event              |
| ae      | aeoutc  | kae     | R aeoutc  | 8      | Outcome of Event Code         |
| ae      | aesser  | kae     | R aesser  | 40     | Seriousness Criteria          |
| ae      | aescrc  | kae     | R aescrc  | 8      | Seriousness Criteria Code     |
| ae      | aesevty | $aesev.| kae       | R aesev | $10   | Severity of Event            |
| ae      | aesevtyc| kae     | R aesevc  | 8      | Severity of Event Code        |
| ae      | aestdt  | kae     | R aestdtc| $10    | Start Date of Event (char)    |
| ae      | aeverb  | kae     | R aeterm  | $200   | Reported Term                 |
|         |         |         | kae       | R visit | 8      | Visit                         |
| ae      | visitnum| kae     | R visitnum| 8      | Visit Id                       |
| ae      | siteid  | kae     | R siteid  | $3      | Site                           |

Note that for some lines, the keyword "_all_" is used in place of data set names. This tells the Conversion Macros that these variables should exist in all data sets created. Sometimes standard variables have no corresponding source variables; such an example is variable VISIT (towards the bottom of the table). In this case, the Macros will create the variable with the attributes specified and missing values. Other keywords and characters are available to provide shortcuts to tell the Macros how to convert data.

The next figure, Figure 4, presents PROC FORMAT code for the demography and adverse events conversions. The first format, $dslabel, provides the standard data set labels for the converted data sets. The remaining formats, which were all referenced in the above mapping table, are used to recode categorical variables. For example, in study P101, the race decodes match the standards. However, the coded values are 0, 1, 2, and 3, not the standard values of 1, 2, 3, and 4. This was seen in Figure 2, in the detailed listing from the Investigation Macros. For source variable AEACT (action taken with regard to the adverse event), both the codes and decodes differ from the standard. Format $AEACT will modify the decodes, while format AEACTC changes the codes to match the target data model.
Figure 4: PROC FORMAT code

```
proc format;
  value $dslabel
    'KAE     ' = 'ADVERSE EVENTS                 '
    'KDEMOG  ' = 'DEMOGRAPHICS                   '
  ;
  value race
    0='1'
    1='2'
    2='3'
    3='4'
  ;
  value $aesev
    'MILD'='MILD'
    'MODERATE'='MODERATE'
    'MARKED'='SEVERE'
  ;
  value $aeact
    'NONE'='NONE'
    'DOSE CHANGE'='DOSE ADJUSTED'
    'DOSE ADJUSTED'='DOSE ADJUSTED'
    'TEMPORARY STOP'='TEMPORARY STOP'
    'PERMANENT STOP'='PERMANENT STOP'
  ;
  value aeactc
    1='1'
    2,99='2'
    3='3'
    4='4'
  ;
```

The Macros produce a number of reports as well as SAS log messages to aid the user during the conversion process; details are below.

- A listing of the source and target variables and data sets (similar to the mapping table).
- Cross-tabulations of source and target categorical variables for source variables that the user recoded (see Figure 5).
- Listings of potential problems, including (a) variables in the new data set which had no corresponding source variables, (b) variables in the source data set(s) which are not in the new data set, and (c) variables present in the mapping table, which do exist in the source data set(s).
- PROC CONTENTS of the new data set.
- PROC PRINT of the first 20 observations of the new data set.

In Figure 3, the user specified in the mapping table a format to recode categorical variable RACE to RACEC; Figure 4 displayed the format. Figure 5 below shows how the Conversion Macro has processed this information. This listing allows the user to confirm that the categorical variables have been recoded correctly, both in terms of the translation (ie, value 0 to 1) and the frequency counts.

Figure 5: Cross-tabulations of source and target categorical variables

<table>
<thead>
<tr>
<th>Variable Mapping</th>
<th>Incoming Value</th>
<th>Outgoing Value</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>RACE --&gt; RACEC</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

Messages to the SAS log provide additional information about potential issues and conflicts with the conversion input files the user provided. Information is also contained in the permanent file of SAS code created by the Macros to
convert the data. The user should carefully review all files to confirm that the conversions have been executed correctly.

CREATION OF THE POOLED DATABASE

Thus far, the user has reviewed the source data for several clinical studies using the Investigation Macros. Based on the findings from the Investigation Macros, the user created the appropriate input files to enable the Conversion Macros to convert the data to a single, consistent standard. To take maximum advantage of this effort, the next step is generally to combine the data across the drug compound. There could be one physical database for each indication or one combined database for all indications. If indication-level databases are created, care should be taken to ensure consistency across these databases so they can be pooled across indications, if necessary. Below are a few tips for creating pooled databases.

- Convert a trial’s data as soon as the individual trial has been locked. Conversion is easier while the data and their idiosyncrasies are still fresh in everyone’s minds.
- Document the required transformations. This will enable users of the data to understand the meaning of the variables and provide information about how to handle similar variables that may be encountered in future studies.
- Wait until the conversion of the data has been verified before adding the new data to the pooled database. Issues identified during the verification process could necessitate the recreation of the converted data set.
- Focus on variables that are meaningful across the compound. If a variable appears in only a few trials, it may not be needed when all trials for the compound are pooled.

VERIFICATION

In the process of converting and pooling data, many input data sets may be combined. Some variables are transformed (e.g., categorical variables that are recoded) while others have no changes applied to them. Still other variables may be new variables that do not exist in the source data but are derived for the new data set. Thus, discrepancies between the pooled database and previously created reports using individual data sources are to be expected. However, an explanation for each discrepancy should exist and be consistent with the transformation or derivation applied.

The procedure for verifying pooled databases involves generating frequency counts and summary statistics on both the source data and the target database and then comparing the outputs to make sure they are either consistent given the transformations applied or identical if no transformations were applied. Both the Conversion and Investigation Macros support this process.

- The Conversion Macros’ listing of the source and target variables and data sets provides clear documentation of how each variable was mapped. This listing is similar to the mapping table, except that it also contains messages for the user about the conversion. The mapping table itself and the PROC FORMAT code also provide mapping information.
- Figure 5 above displayed the Conversion Macros’ report on recoding categorical variables. As previously mentioned, this output allows the user to confirm that the categorical variables were recoded correctly.
- The PROC CONTENTS produced by the Conversion Macros allows the user to check a number of items about the new data set: the variables and their attributes (length, type, label); the data set label; and the number of observations, which should be the same as in the source data set.
- The Conversion Macros’ PROC PRINT of the first 20 observations allows the user to check a random sample of values. This is recommended, especially for newly derived variables.
- The Investigation Macros can be run on the pooled database, just as they are executed on the individual trials at the start of this process. Running the Macros on pooled data provides another way to verify the final data values present in the database (via the cross-tabulations and detailed listing) and the data set structure (via the PROC CONTENTS).

If “empty” data sets of the standard target data model exist (i.e., data sets with variables defined but no observations), they can be used as a separate protocol in the execution of the Investigation Macros. In this way, the user can assess the uniformity of the source data variables’ attributes directly against the target data model. This can be done both before and after conversion: before, to define the needed transformations and after, to verify the results.

CONCLUSION

Pooled databases of clinical trials data have the potential to provide a tremendous amount of information. In order for this information to be meaningful, the data structure must be consistent across trials. Converting data to a single structure consumes time and personnel resources. However, the monetary benefits of such an initiative can be significant. The use of the Investigation and Conversion Macros described in this paper help enhance the cost-benefit ratio of data conversion and pooling.
ACKNOWLEDGEMENTS
The author thanks Karen Grippo of R-Square, Inc. for her suggestions regarding this paper and her support of the Macros.

CONTACT INFORMATION
Any questions or comments? Please contact the author:
Patricia Majcher
Johnson & Johnson Pharmaceutical Research & Development, L.L.C.
920 Route 202
Raritan, NJ 08869
Phone: 908-704-4165
Fax: 908-526-5196
Email: pmajcher@prdus.jnj.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.