The Program Specifications for the User Interface Using SAS® Software for the Worldwide Household Goods Information System for Transportation Modernization (WHIST-MOD)

Teresa L. James
Jane P. Loftis
Tai-Lun Chiang
Energy Division

THE PROGRAM SPECIFICATIONS FOR THE USER INTERFACE USING SAS® SOFTWARE FOR THE WORLDWIDE HOUSEHOLD GOODS INFORMATION SYSTEM FOR TRANSPORTATION MODERNIZATION (WHIST-MOD)

Teresa L. James*
Jane P. Lofis
Tai-Lun Chiang*

*University of Tennessee, Knoxville

Date Published: July 1991

Prepared for the Military Traffic Management Command Falls Church, Virginia 22041 under Intergency Agreement DOE No. 1405-1405-A2

Prepared by the OAK RIDGE NATIONAL LABORATORY Oak Ridge, Tennessee 37831-6285 managed by MARTIN MARIETTA ENERGY SYSTEMS, INC. for the U.S. DEPARTMENT OF ENERGY under Contract No. DE-AC05-84OR21400
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 3.2.1.</td>
<td>Hierarchy of programs called by MENU.PROGRAM</td>
<td>10</td>
</tr>
<tr>
<td>Fig. 3.2.2.</td>
<td>Primary menu program showing macrovariables used or initialized in the program</td>
<td>11</td>
</tr>
<tr>
<td>Fig. 3.2.3.</td>
<td>The MENU.PROGRAM screen</td>
<td>16</td>
</tr>
<tr>
<td>Fig. 3.2.4.</td>
<td>The MENU.PROGRAM screen with field names</td>
<td>17</td>
</tr>
<tr>
<td>Fig. 3.3.1.</td>
<td>Hierarchy of programs in the tabular program set</td>
<td>18</td>
</tr>
<tr>
<td>Fig. 3.3.2.</td>
<td>Hierarchy of tabular programs showing macrovariables set, used, or initialized in the program</td>
<td>20</td>
</tr>
<tr>
<td>Fig. 3.3.3.</td>
<td>Hierarchy of tabular programs showing permanent SAS datasets read, created, or overwritten</td>
<td>21</td>
</tr>
<tr>
<td>Fig. 3.3.4.</td>
<td>The TABULAR.PROGRAM screen</td>
<td>24</td>
</tr>
<tr>
<td>Fig. 3.3.5.</td>
<td>The TABULAR.PROGRAM screen with field names</td>
<td>25</td>
</tr>
<tr>
<td>Fig. 3.3.6.</td>
<td>The TABDATA.PROGRAM screen</td>
<td>29</td>
</tr>
<tr>
<td>Fig. 3.3.7.</td>
<td>The TABDATA.PROGRAM screen with field names</td>
<td>30</td>
</tr>
<tr>
<td>Fig. 3.3.8.</td>
<td>The TIME.PROGRAM screen</td>
<td>32</td>
</tr>
<tr>
<td>Fig. 3.3.9.</td>
<td>The TIME.PROGRAM screen with field names</td>
<td>32</td>
</tr>
<tr>
<td>Fig. 3.3.10.</td>
<td>The RANGE.PROGRAM screen</td>
<td>35</td>
</tr>
<tr>
<td>Fig. 3.3.11.</td>
<td>The RANGE.PROGRAM screen with field names</td>
<td>36</td>
</tr>
<tr>
<td>Fig. 3.3.12.</td>
<td>The SORT.PROGRAM screen</td>
<td>40</td>
</tr>
<tr>
<td>Fig. 3.3.13.</td>
<td>The SORT.PROGRAM screen with field names</td>
<td>41</td>
</tr>
<tr>
<td>Fig. 3.3.14.</td>
<td>The GROUP1.PROGRAM screen</td>
<td>45</td>
</tr>
<tr>
<td>Fig. 3.3.15.</td>
<td>The GROUP1.PROGRAM screen with field names</td>
<td>46</td>
</tr>
<tr>
<td>Fig. 3.3.16.</td>
<td>The TOPTION.PROGRAM screen</td>
<td>48</td>
</tr>
<tr>
<td>Fig. 3.3.17.</td>
<td>The TOPTION.PROGRAM screen with field names</td>
<td>49</td>
</tr>
<tr>
<td>Fig. 3.3.18.</td>
<td>The STDDEV.PROGRAM screen</td>
<td>51</td>
</tr>
<tr>
<td>Fig. 3.3.19.</td>
<td>The STDDEV.PROGRAM screen with field name</td>
<td>51</td>
</tr>
<tr>
<td>Fig. 3.3.20.</td>
<td>The TOUTOPT.PROGRAM screen</td>
<td>55</td>
</tr>
<tr>
<td>Fig. 3.3.21.</td>
<td>The TOUTOPT.PROGRAM screen with field names</td>
<td>55</td>
</tr>
<tr>
<td>Fig. 3.3.22.</td>
<td>The TREADY.PROGRAM screen</td>
<td>60</td>
</tr>
<tr>
<td>Fig. 3.3.23.</td>
<td>The BAR.PROGRAM screen</td>
<td>62</td>
</tr>
<tr>
<td>Fig. 3.3.24.</td>
<td>The BAR.PROGRAM screen with field names</td>
<td>63</td>
</tr>
<tr>
<td>Fig. 3.3.25.</td>
<td>The BARDATA.PROGRAM screen</td>
<td>66</td>
</tr>
<tr>
<td>Fig. 3.3.26.</td>
<td>The BARDATA.PROGRAM screen with field names</td>
<td>67</td>
</tr>
<tr>
<td>Fig. 3.3.27.</td>
<td>The BARDATA.PROGRAM screen with field names</td>
<td>72</td>
</tr>
<tr>
<td>Fig. 3.3.28.</td>
<td>The BARDATA.PROGRAM screen with field names</td>
<td>73</td>
</tr>
<tr>
<td>Fig. 3.3.29.</td>
<td>The TIME.PROGRAM screen</td>
<td>75</td>
</tr>
<tr>
<td>Fig. 3.3.30.</td>
<td>The TIME.PROGRAM screen with field names</td>
<td>75</td>
</tr>
<tr>
<td>Fig. 3.3.31.</td>
<td>The BMRANGE.PROGRAM screen</td>
<td>80</td>
</tr>
<tr>
<td>Fig. 3.3.32.</td>
<td>The BMRANGE.PROGRAM screen with field names</td>
<td>80</td>
</tr>
<tr>
<td>Fig. 3.3.33.</td>
<td>The BMRANGE.PROGRAM screen with field names</td>
<td>83</td>
</tr>
</tbody>
</table>

v
FIGURES, Cont'd

Fig. 3.4.15. The BOPTION.PROGRAM screen with field names ........................................... 83
Fig. 3.4.16. The BOUTOPT.PROGRAM screen ............................................................................. 86
Fig. 3.4.17. The BOUTOPT.PROGRAM screen with field names .................................................. 86
Fig. 3.4.18. The BREADY.PROGRAM screen ................................................................................. 90
Fig. 3.5.1. Hierarchy of programs in the line program set ............................................................... 92
Fig. 3.5.2. Hierarchy of line programs showing macrovariables set, used, or initialized in the programs ........................................................................................................................................ 93
Fig. 3.5.3. Hierarchy of line programs showing permanent SAS datasets read or created ............. 94
Fig. 3.5.4. The LINE.PROGRAM screen ......................................................................................... 96
Fig. 3.5.5. The LINE.PROGRAM screen with field names ............................................................ 96
Fig. 3.5.6. The LINEDATA.PROGRAM screen ............................................................................. 100
Fig. 3.5.7. The LINEDATA.PROGRAM screen with field names .................................................. 100
Fig. 3.5.8. The BRANGE.PROGRAM screen ................................................................................. 102
Fig. 3.5.9. The BRANGE.PROGRAM screen with field names ..................................................... 102
Fig. 3.5.10. The LOPTION.PROGRAM screen .............................................................................. 104
Fig. 3.5.11. The LOPTION.PROGRAM screen with field names .................................................. 105
Fig. 3.5.12. The LOUTOPT.PROGRAM screen .............................................................................. 107
Fig. 3.5.13. The LOUTOPT.PROGRAM screen with field names .................................................. 108
Fig. 3.5.14. The LREADY.PROGRAM screen ................................................................................. 111
Fig. 3.6.1. Hierarchy of programs in the pie program set ............................................................... 113
Fig. 3.6.2. Hierarchy of pie programs showing macrovariables set, used, or initialized in the programs ........................................................................................................................................ 114
Fig. 3.6.3. Hierarchy of pie programs showing permanent SAS datasets read or created ............. 115
Fig. 3.6.4. The PIE.PROGRAM screen ......................................................................................... 117
Fig. 3.6.5. The PIE.PROGRAM screen with field names .............................................................. 118
Fig. 3.6.6. The PIEDATA.PROGRAM screen ............................................................................... 122
Fig. 3.6.7. The PIEDATA.PROGRAM screen with field names ..................................................... 122
Fig. 3.6.8. The BRANGE.PROGRAM screen ............................................................................... 124
Fig. 3.6.9. The BRANGE.PROGRAM screen with field names ..................................................... 124
Fig. 3.6.10. The TIME.PROGRAM screen ..................................................................................... 126
Fig. 3.6.11. The TIME.PROGRAM screen with field names .......................................................... 126
Fig. 3.6.12. The POUTOPT.PROGRAM screen .......................................................................... 129
Fig. 3.6.13. The POUTOPT.PROGRAM screen with field names .................................................. 129
Fig. 3.6.14. The PREADY.PROGRAM screen ............................................................................... 132
### ACRONYMS AND TERMS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freelance</td>
<td>Graphics and charting software</td>
</tr>
<tr>
<td>MTPP</td>
<td>Directorate of Personal Property, MTMC</td>
</tr>
<tr>
<td>MTMC</td>
<td>Military Traffic Management Command</td>
</tr>
<tr>
<td>ORACLE</td>
<td>Relational Database Management and related software</td>
</tr>
<tr>
<td>ORNL</td>
<td>Oak Ridge National Laboratory</td>
</tr>
<tr>
<td>QA</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>SAS</td>
<td>Data Management and Statistical Analysis Toolset</td>
</tr>
<tr>
<td>SAS/AF</td>
<td>SAS's screen generation software</td>
</tr>
<tr>
<td>SAS/GRAPH</td>
<td>SAS's graphics and charting software</td>
</tr>
<tr>
<td>WHIST-MOD</td>
<td>Worldwide Household Goods Information System for</td>
</tr>
<tr>
<td></td>
<td>Transportation Modernization</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

The authors would like to thank the staff at the Directorate of Personal Property, Military Traffic Management Command for their input and support. The authors would also like to thank Robin Noe for her professional contribution to the preparation and publication of this document.
ABSTRACT

The Directorate of Personal Property of the Military Traffic Management Command (MTMC) asked Oak Ridge National Laboratory (ORNL) to design and prototype a decision support system, the Worldwide Household Goods Information System for Transportation Modernization (WHIST-MOD). This decision support system will automate current tasks and provide analysis tools for evaluating the Personal Property Program, predicting impacts to the program, and planning modifications to the program to meet the evolving needs of military service members and the transportation industry. The system designed by ORNL consists of three application modules: system dictionary applications, data acquisition applications, and user applications. The development of the user applications module is divided into two phases. Round 1 is the data selection front-end interface, and Round 2 is the output or back-end interface.

This report contains the design specification for the back-end output interface of the user applications module, which was prototyped using SAS, a data management and statistical analysis toolset. This report discusses our goals for the back-end interface and describes how we met these goals. The prototype design fulfills the users' needs for a flexible interface that can be used to produce a variety of tabular and graphical output.
1. INTRODUCTION

1.1 BACKGROUND

The Worldwide Household Goods Information System for Transportation Modernization (WHIST-MOD) being designed and prototyped by Oak Ridge National Laboratory (ORNL) is a decision support system for the various organizations of the Military Traffic Management Command (MTMC) that establish and implement the Personal Property Movement and Storage Program. The decision support system will benefit the staff of the Personal Property Program in their tasks of program evaluation and policy setting. This system is designed to access a distributed database through a powerful set of information management tools.

The prototype system offers users, even those with minimal computer experience, easy access to a large selection of data elements and the ability to formulate complex queries. In addition, users may perform special studies and one-time-only \textit{ad hoc} queries. WHIST-MOD will be a dynamic system that evolves to meet the changing needs of the Directorate of Personal Property MTMC (MTPP) staff.

User requirements for the WHIST-MOD system were identified during the analysis phase of the project. During this phase, ORNL identified three modules for prototyping to aid decision support activities at MTPP. These modules include applications to provide a description of components of the system (system dictionary applications), applications to acquire data (data acquisition applications), and applications to retrieve and display data from the database (user applications).

1.2 THE APPLICATIONS MODULE

The user applications module was prototyped in two design and development phases. Round 1, the front-end interface, was prototyped using a relational database management system (RDBMS), ORACLE, and its associated toolset. The front-end interface includes screens that allow users to choose, retrieve, and store a subset of data. These data are then passed to the back-end interface. For more details on the design of the front-end interface see the "User Interface in ORACLE for the Worldwide Household Goods Information System for Transportation Modernization (WHIST-MOD)" document.  

During Round 2, the back-end interface was prototyped using SAS, a data management and statistical analysis toolset. The back-end interface allows the user to specify report types and formats and to produce output based on the dataset passed to SAS from the front-end interface.

1.3 OVERVIEW

This report presents the design specifications for the postprocessing back-end interface. It contains a discussion of the goals of the interface and a description of the actual prototype. ORNL has not been tasked with implementing the entire design, but rather with prototyping significant pieces that will serve as the basis from which MTPP staff will implement and expand the system.

Section 1 provides an introduction, and section 2 discusses the goals for the interface. Section 3 describes the SAS interface programs and screens prototyped to meet these goals.
2. GOALS FOR THE INTERFACE

We identified the goals for the back-end interface during the requirements analysis phase of the WHIST-MOD project. The goals are based on our analysis of current user needs and our vision of the current and future needs of the users and the system itself. The following is a list of goals for the back-end interface.

- The interface must be modular to support modifications and expansion.
- The interface must be generic enough to be used to produce a variety of reports.
- The interface must be easy to use.

The following sections discuss each of these goals.

2.1 MODULAR DESIGN

Because of the evolving nature of the personal property business, the interface must be designed to easily incorporate modifications and expansions. The need for modifications and expansions may come from a variety of sources. As MTPP staff monitor the Personal Property Program they will identify the need for changes and additions to the program. External sources, such as Congress, the carrier industry, and service members, may also create the need for changes in the program. For example, MTPP is currently in the process of consolidating separate quality assurance (QA) programs, international QA and domestic QA, into one standardized QA program. Once this new QA program is implemented, the WHIST-MOD system, including the back-end interface, will have to be modified to support this new program. Due to the changing needs of service members, MTPP recently expanded the definition of personal property to include privately owned pleasure boats. This change added a new category of personal property that needs to be supported by the system.

To remain useful, the system must be designed to easily accommodate these frequent and often unpredictable types of changes. In addition, the modular design of the back-end interface will be easier for programmers to understand than one long program and, therefore, less time-consuming to maintain.

The prototype back-end interface is modular. Each output type (i.e., tabular report, bar chart, line chart, or pie chart) constitutes a different program set, and within each program set, each function is a separate entity. For example, there is one program to support the selection of an output destination in the pie program set and a separate program to support the selection of an output destination in the bar program set. Another program supports the selection of variables that will be used to sort the output. Because these program sets and programs are designed as separate entities, it will be easy to add program sets and/or programs to support additional functions. For example, MTPP might require a program set that will produce three dimensional bars or a program that provides regression options, such as confidence intervals, which can easily be added without significantly affecting existing functions. In addition, because functions exist as separate entities, it will be easy for the MTPP maintenance and modification staff to identify and locate specific programs where modifications are desired.
2.2 GENERIC DESIGN

Another goal for the back-end interface design is that it be generic enough to be used to produce a variety of reports. The interface is generic on three levels to support the production of reports.

- It supports the production of 36 report types.
- It supports the production of four output types for each report type.
- It supports the selection/manipulation of data for each output type.

2.2.1 Variety of Report Types

Staff at MTPP currently produce different reports based on their work division. There are three divisions of work at MTPP: QA and Operations, Rates, and Management Support. Some staff assigned to work in the QA and Operations Division monitor carrier performance. They often produce reports that identify carriers who miss delivery dates, carriers who have been disqualified from the Personal Property Program, or carriers who damage shipments in transit. Staff who work in the Rates Division monitor the changes in the carrier rates that were bid for moving and storing personal property. The staff who work in the Management Support Division monitor the overall program and produce reports for external agencies such as Congress and the Armed Services.

During the requirements analysis phase of WHIST-MOD, 36 report types that encompass the needs of these three work divisions were identified. For example, there is a Missed Required Delivery Date Report that is frequently produced by the QA and Operations staff. Rates staff need to produce a Change in Rate Levels Report. All three divisions produce a Number of Shipments Report and a Net Weight Shipped Report.

The prototype front-end interface completed by ORNL in March 1990 was designed to retrieve data for each of these 36 report types. The back-end interface is generic enough to accept the data retrieved by the front-end and produce output for each of the 36 report types.

2.2.2 Variety of Output Types

For each of these report types the staff at MTPP may need, depending on the purpose, to produce different output types. They need bar charts, line charts, and pie charts to represent trends in the Personal Property Program and tabular output to report detailed information. These output types may be used internally as an analysis tool or externally to provide information for other agencies. For example, if MTPP staff are analyzing trends in shipment weights for specific carriers, or if Congress requests shipment weight information, MTPP staff need the ability to quickly produce tabular or graphical reports.

The back-end interface is generic enough to support the production of tabular as well as a variety of graphical output for each report type. The prototype back-end interface supports the production of the following output types:

- tabular reports,
- horizontal bar charts,
- vertical bar charts,
• line charts, and
• pie charts.

A query in the front-end selects data from the appropriate database table based on the parameters the user chooses. Some of the data selected and the specific database tables used for the selection of data are dependent on the report type the user chooses. However, these data are stored in a single data file that is used by the back-end, regardless of the report type the user wants to produce. This design is generic because it does not require multiple dataset files or different sets of back-end programs for each report type.

2.2.3 Selection/Manipulation of Data

For each report type, staff at MTPP need the ability to produce reports that display different data. For example, users may wish to produce a Net Weight Shipped Report that shows the weight shipped for specific carriers for a particular year. They may also need to produce the same report type that shows the weight shipped by particular codes of service for the same year.

Therefore, another goal of the design of the back-end interface is to provide the ability for users to select different data columns from the dataset file created in the front-end interface. The dataset file prototyped in our front-end interface is created from a query of the database and contains a prescribed set of data columns, depending on the report type the user has chosen to produce. From this set of data columns, the user may choose specific columns for a report type.

Not only do the users need to select data columns from the dataset file, they also need the ability to manipulate these data. For a tabular report the users need to choose data columns for sorting the output. For example, they may need the tabular output sorted by codes of service, or by carriers, or by one of the other data columns. They also need to perform different calculations on these data. They need to select the particular data column to use for the calculation, and they need to be able to specify the calculation type. For example, the user may want to calculate an average on the weight column for a Net Weight Shipped Report.

The prototype back-end interface is generic enough to support interactive sorting and the interactive selection of calculation variables. It also supports the following calculation types:

• average,
• percent, and
• total.

2.3 EASE OF USE

MTPP staff also need an interface that is easy to use because there is a high rate of staff turnover and they have disparate computer skill levels. MTPP frequently has new employees and could lose experienced staff. And although some MTPP staff currently are experienced computer users, others have minimal computer experience. All MTPP staff must produce reports.
The prototype back-end interface is easy to use so that even inexperienced computer users can quickly and easily produce a variety of reports. The interface protects the user from the sophistication and complexity of the application development software. Each screen allows users to select and manipulate data by simply pressing a key, and extensive interactive help messages are incorporated into the interface.
3. THE SAS BACK-END

3.1 OVERVIEW

The SAS/AF programs written for the WHIST-MOD prototype produce screens that allow
the user to manipulate the data selected from the ORACLE front-end. The user may manipulate
these data for reporting purposes by choosing a variety of options such as the following:

1. the output type (i.e., a tabular report, a bar chart, a line chart, or a pie chart),
2. the data columns to be included on a report/chart,
3. the data column on which to perform arithmetic operations,
4. the type of arithmetic operation to be performed, and
5. the destination for the output.

Other options, depending on the type of report or chart the user has chosen to produce, may also
be available. All options will be discussed in this document.

The SAS/AF and SAS/GRAPH programs have been designed to support the flexible,
generic nature of the WHIST-MOD user applications prototype. These programs are generic and
flexible because they allow the user to produce approximately 36 different types of reports. For
example, using these SAS/AF and SAS/GRAPH programs in conjunction with the ORACLE
programs, the user may, among others, produce a "Tonnage" report, a report on "Average Weight
Shipped," or a "Missed Pickup" report. The SAS programs have additional flexibility because for
any specific report type, the user may produce a tabular report, a pie chart, a line chart, or a bar
chart. A third type of flexibility is built into the programs that allows the user to choose a variety
of data for any report type without having to revisit the ORACLE front-end screens.

For example, once the user has selected data parameters from the ORACLE front-end,
he/she may select specific columns for the report to be produced. The user, therefore, may select
codes of service, shipment weights, and carriers for one report or chart and, using the same data,
produce a second report that displays shipment pickup dates, carriers, and destination areas.

Besides being flexible and generic, the SAS/AF prototype programs are designed to be
modular to allow for easy expansion. Currently there are five program sets: a menu program,
programs that produce tabular reports, programs that produce bar charts, programs that produce
line charts, and programs that produce pie charts. Additional program sets that produce other
charts could easily be added to the prototype.

3.1.1 Screen Types

There are two basic types of screens that the user will see: SAS/AF program screens and
SAS function screens. The names of all program screens end with the .PROGRAM suffix. The
main purpose of this document is to describe these program screens. The program screens,
however, may call SAS function screens. There are four types of SAS function screens that have
been used in the back-end prototype:

The SAS Title window,
The SAS Legend window,
The SAS Help window, and
The SAS List Variable window.

Section 3.X.X.5 (where X may be any subsection number) contains a list of the SAS window functions called by each program. Any SAS functions windows that are called by a program are also discussed in the structured English code for that program, Sect. 3.X.X.7.

3.1.2 SAS Dataset Files

There are two types of dataset files that are used by the SAS back-end programs, permanent and temporary files. The front-end interface creates two initial permanent dataset files, data1 and data1a. Other permanent and temporary dataset files are created depending on the functions the user selects. All dataX dataset files are used by the SAS List Variable function window. All dataXa dataset files are used by the SAS calculation procedures. Section 3.X contains a figure that lists all permanent dataset files read, created, or overwritten by that program set. Any SAS dataset files that are read, created, or overwritten by a program are also discussed in the structured English code for that program.

3.1.3 Organization of the SAS Back-End Document

This section of the WHIST-MOD Design Document is organized into six parts. Section 3.1 is an overview which serves to explain the logic of the design of the SAS back-end and gives high-level information about the SAS/AF and SAS/GRAPH programs that will be useful to the WHIST-MOD implementation and maintenance and modification teams.

The remaining sections of this document give detailed information about the programs that are components of the SAS back-end prototype. The sections are organized around program sets. There are five program sets in the SAS prototype:

the primary menu program set (see Sect. 3.2),
the tabular report program set (see Sect. 3.3),
the bar chart program set (see Sect. 3.4),
the line chart program set (see Sect. 3.5), and
the pie chart program set (see Sect. 3.6).

Each of these program set subsections consists of descriptions of the programs that are components of that subset. For example, Sect. 3.3 describes each SAS program that is used to produce tabular reports. Section 3.6 describes each SAS program that is used to produce pie charts. Though the program screens for the production of pie charts, for example, may look to the user like the program screens used for the production of bar charts, the programs that produce the screens are different programs.

Each subsection that describes a program set also contains three hierarchy charts. The first hierarchy chart shows all of the programs in that program set and the interaction of these programs (i.e., which programs call other programs). The second hierarchy chart shows all macrovariables that are used, initialized, or set by the programs within that subset. The third hierarchy chart shows all dataset files that are created, used, or overwritten by the programs within that subset.
3.2 THE PRIMARY MENU PROGRAM

3.2.1 Overview

The primary menu screen, MENU.PROGRAM, is called by the front-end ORACLE interface. It is automatically displayed for the user after the ORACLE query is performed. This screen allows the user to call one of the following screens:

   TABULAR.PROGRAM,
   BAR.PROGRAM,
   LINE.PROGRAM, or
   PIE.PROGRAM.

Figure 3.2.1 shows a hierarchy structure for MENU.PROGRAM and the programs it calls. Figure 3.2.2 shows the macrovariables that are created, initialized, or set by MENU.PROGRAM. No dataset is read, overwritten, or written by this program.

3.2.2 MENU.PROGRAM

3.2.2.1 Program purpose

The MENU.PROGRAM allows the user to select options for the type of report she/he wants to produce. The user's choice determines the appropriate program set (i.e., tabular, bar, line, or pie) to call. The user may choose to produce a tabular report, a vertical bar chart, a horizontal bar chart, a line graph, or a pie chart.

3.2.2.2 Calling program

The front-end interface (Sect. 1.2)

3.2.2.3 Programs called

   TABULAR.PROGRAM
   BAR.PROGRAM
   LINE.PROGRAM
   PIE.PROGRAM

3.2.2.4 Program returns to

May exit to system prompt or to the ORACLE front-end screens
Fig. 3.2.1. Hierarchy of programs called by MENU.PROGRAM
Fig. 3.2.2. Primary menu program showing macrovariables used or initialized in the program.
3.2.2.5 SAS window functions called by program

N/A

3.2.2.6 Macrovariables

Used - N/A

Initialized -

afstr2 initialized to blank

_g_calopt initialized to blank

_g_calvar initialized to blank

_g_date initialized to blank

_g_group initialized to blank

_g_sort1 initialized to blank

_g_sort2 initialized to blank

_g_sort3 initialized to blank

_g_str1 initialized to blank

_g_str2 initialized to blank

_g_tot initialized to blank

_g_xvar initialized to blank

Set -

_g_bartyp contains a string identifying the type of bar chart the user wants to produce if the user chooses to produce a bar chart (i.e., the value will be 'hbar' for horizontal bars and 'vbar' for vertical bars)

3.3.2.7 Structured English for program code

Initialize fields tabular, vbar, hbar, line, pie, and clear to blank
Initialize macrovariables afstr2 and g_calvar to blank
Assign numeric values to each month name
When user presses <return> while in tabular field
If there is not a '*' or a blank in the tabular field then do
   If the vbar, hbar, line, and pie fields are blank then do
      Put a '*' in tabular field
      Put a blank in the clear field
      Call TABULAR.PROGRAM
   Else do
      Display message 'You must delete the other option before choosing again.'
      Highlight the tabular field
      Sound the alarm
      Position the cursor in the tabular field
      Put a blank in the tabular field
   Else do
      Position cursor in the vbar field
      When user presses <return> while in the vbar field
         If there is not a '*' or a blank in the vbar field then do
            If the tabular, hbar, line, and pie fields are blank then do
               Put a '*' in the vbar field
               Put a blank in the clear field
               Create a variable string to hold the phrase 'vbar'
               Create a macrovariable g_bartyp as the value in the variable string
               Call BAR-PROGRAM
            Else do
               Display message 'You must delete the other option before choosing again.'
               Highlight the vbar field
               Sound the alarm
               Position the cursor in the vbar field
               Put a blank in the vbar field
         Else do
            Position cursor in the hbar field
            When user presses <return> while in the hbar field
               If there is not a '*' or a blank in the hbar field then do
                  If the tabular, vbar, line, and pie fields are blank then do
                     Put a '*' in the hbar field
                     Put a blank in the clear field
                     Create a variable string to hold the phrase 'hbar'
                     Create a macrovariable g_bartyp as the value in the variable string
                     Call BAR-PROGRAM
                  Else do
                     Display message 'You must delete the other option before choosing again.'
                     Highlight the hbar field
                     Sound the alarm
                     Position the cursor in the hbar field
                     Put a blank in the hbar field
               Else do
                  Display message 'You must delete the other option before choosing again.'
                  Highlight the hbar field
                  Sound the alarm
                  Position the cursor in the hbar field
                  Put a blank in the hbar field
Else do
  Position cursor in the line field
When the user presses <return> while in the line field
  If there is not a '*' or a blank in the line field then do
    If the tabular, vbar, hbar, and pie fields are blank then do
      Put a '*' in the line field
      Put a blank in the clear field
      Call LINE.PROGRAM
    Else do
      Display message 'You must delete the other option before choosing again.'
      Highlight the line field
      Sound the alarm
      Position the cursor in the line field
      Put a blank in the line field
  Else do
    Position cursor in the pie field
When the user presses <return> while in the pie field
  If there is not a '*' or a blank in the pie field then do
    If the tabular, vbar, hbar, and line fields are blank then do
      Put a '*' in the pie field
      Put a blank in the clear field
      Call PIE.PROGRAM
    Else do
      Display message 'You must delete the other option before choosing again.'
      Highlight the pie field
      Sound the alarm
      Position the cursor in the pie field
      Put a blank in the pie field
  Else do
    Position cursor in the clear field
When the user presses <return> while in the clear field
  If there is not a '*' or a blank in the clear field then do
    Put a '*' in the clear field
    Put a blank in the following macrovariables:
      g_sort1
      g_sort2
      g_sort3
      g_group
      g_calopt
      g_calvar
      g_xvar
      g_date
      afstr2
      g_str1
      g_str2
      g_tot
Put a blank in the following fields:
  tabular  
  line  
  vbar  
  hbar  
  pie  
Position the cursor in the tabular field  
Put a blank in the clear field  
Set titles to null  
Else position cursor on the command line

3.2.2.8 Notes

This is the initial interface screen. It is a required screen. Only one report type may be chosen at a time.

3.2.2.9 Program screens

Figure 3.2.3 represents the screen the user will see. Figure 3.2.4 contains the field names in parentheses on the program screen.

3.3 TABULAR PROGRAMS

The tabular programs are called when the user specifies 'Tabular Report' on the MENU.PROGRAM screen. The first screen the user will see is TABULAR.PROGRAM. This screen allows the user to call one of the following screens:

  TABDATA.PROGRAM,  
  SORT.PROGRAM,  
  TOPTION.PROGRAM,  
  The SAS titles function window, or  
  TOUTOPT.PROGRAM.

Figure 3.3.1 is a hierarchy chart of the programs that are components of the tabular program set. TABDATA.PROGRAM allows the user to choose variables from the original dataset for inclusion in the report. It is a mandatory screen. TABDATA.PROGRAM may also call TIME.PROGRAM and/or RANGE.PROGRAM. If the user chooses the variable 'PICKUP DATE' in TABDATA.PROGRAM, TIME.PROGRAM is called. TIME.PROGRAM allows the user to specify the way he/she wants to look at time (i.e., by year; by month and year; or by day, month, and year). RANGE.PROGRAM allows the user to specify a range of values to be used in the report for any numeric variable chosen in TABDATA.PROGRAM.

SORT.PROGRAM allows the user to choose up to three variables for sorting the data for the report. It is not a mandatory screen. SORT.PROGRAM calls GROUP1.PROGRAM. GROUP1.PROGRAM allows the user to subtotal by the first or the first and second sort variable(s) chosen in the SORT.PROGRAM.
Output Menu

- Tabular Report
- Vertical Bar Chart
- Horizontal Bar Chart
- Line Chart
- Pie Chart
- Clear Previous Output Selections

Enter 'X' beside your choice.
Press RETURN.

Press F10 to exit this screen.

Fig. 3.2.3. The MENU.PROGRAM screen
Output Menu

- (tabular) Tabular Report
- (vbar) Vertical Bar Chart
- (hbar) Horizontal Bar Chart
- (line) Line Chart
- (pie) Pie Chart
- (clear) Clear Previous Output Selections

Enter 'X' beside your choice.
Press RETURN.
Press F10 to exit this screen.

Fig. 3.2.4. The MENU.PROGRAM screen with field names.
Fig. 3.3.1. Heirarchy of programs in the tabular program set
TOPTION.PROGRAM allows the user to choose a calculation type for the report. It is a mandatory screen. The choices are 'AVERAGE,' 'TOTAL,' and 'PERCENT.' If the user chooses 'AVERAGE,' STDDEV.PROGRAM and AVG.PROGRAM are called. STDDEV.PROGRAM allows the user to specify whether or not to include the standard error of the mean in the output. AVG.PROGRAM creates a dataset file that contains the averages of the data and the standard error if the user has chosen to include the standard error. The SAS titles function window is optional. The user may specify up to ten lines for a title to be included in the report.

TOUTOPT.PROGRAM allows the user to direct the output. The output may be directed to a printer or to the screen.

TREADY.PROGRAM sends the output to the destination specified by the user. It also does the calculations if the user chooses 'PERCENT' in the TOPTION.PROGRAM window.

Each of these programs may use macrovariables that have been initialized or set from another program or the programs may themselves initialize or set macrovariables that may be used by subsequent programs. Figure 3.3.2 shows the macrovariables that are initialized, set, or used by each of the programs that makeup the tabular program set. Figure 3.3.3 shows the dataset file(s) that is used as input and the dataset file(s) that is written when the program is executed. In some cases no dataset file is read and/or no new dataset file is written during execution of a program.

3.3.1 TABULAR.PROGRAM

3.3.1.1 Program purpose

TABULAR.PROGRAM allows the user to select options to generate a tabular report. From this screen the user chooses to call other screens for the purpose of selecting data columns for a report, selecting sort columns for ordering data, selecting a calculation type, and selecting the output destination.

3.3.1.2 Calling program

MENU.PROGRAM

3.3.1.3 Programs called

TABDATA.PROGRAM
SORT.PROGRAM
TOPTION.PROGRAM
TOUTOPT.PROGRAM

3.3.1.4 Program returns to

MENU.PROGRAM
Fig. 3.3.2. Hierarchy of tabular programs showing macrovariables set, used, or initialized in the programs.
Fig. 3.3.3. Hierarchy of tabular programs showing permanent SAS datasets read, created, or overwritten.
3.3.1.5 SAS window functions called by program

SAS Titles Window

3.3.1.6 Macrovariables

Used - afstr2 contains user chosen variables from TABDATA.PROGRAM

Initialized - g_group initialized to blank
g_vars initialized to blank

Set - N/A

3.3.1.7 Structured English for program code

Initialize g_group and g_vars to null
When user presses <return> while in first field - var2
  If there is not a '*' or a blank in the var2 field
  then do
    Put a '*' in the var2 field
    Call TABDATA.PROGRAM
    Copy value in afstr2 to the local variable str2
  Else position cursor in var3 field
When user presses <return> while in second field - var3
  If there is not a '*' or a blank in the var3 field
  then do
    If str2 is blank then do
    Sound alarm
    Display message 'You must select data before sorting.'
    Put a blank in var3
  Else
    Put a '*' in var3
    Call SORT.PROGRAM
Else position cursor in var4 field
When user presses <return> while in third field - var4
  If there is not a '*' or a blank in the var4 field
  then do
    Put a '*' in the var4 field
    Call TOPTION.PROGRAM
  Else position cursor in var5 field
When user presses <return> while in fourth field - var5
  If there is not a '*' or a blank in the var5 field
  then do
    Put a '*' in the var5 field
    Display message 'Press RETURN and fill in titles.'
Press F10 when finished.
Call the SAS Titles Window
Else position cursor in var6 field
When user presses <return> while in the fifth field - var6
If there is not a '*' or a blank in the var6 field
then do
  Put a '*' in the var6 field
  Call TOUTOPT.PROGRAM
Else position cursor on the command line

3.3.1.8 Notes

This is a required window.

3.3.1.9 Program screen

Figure 3.3.4 represents the screen the user will see. Figure 3.3.5 contains the field names in parentheses on the program screen.

3.3.2 TABDATA.PROGRAM

3.3.2.1 Program purpose

TABDATA.PROGRAM allows the user to select the column name(s) of data to include on a report through a SAS VARLIST Window. The user must select one or more variables. This program creates new datasets data2 and data2a, keeping the data only in the columns the user has chosen to include from the original datasets data1 and data1a. If the user chooses pickup date as a column to include, this program calls TIME.PROGRAM and keeps the date data in the new datasets in the date format the user chooses. The user may choose to specify a minimum and/or maximum value for one of the numeric variables already chosen by calling RANGE.PROGRAM from this window.

3.3.2.2 Calling program

TABULAR.PROGRAM

3.3.2.3 Programs called

May call  RANGE.PROGRAM
          TIME.PROGRAM
Output Selection

- Select Variables for Analysis
- Select Sort Variable(s)
- Select Calculation Option
- Specify Titles
- Select Output Option

Fig. 3.3.4. The TABULAR.PROGRAM screen.
Output Selection

- (var2) Select Variables for Analysis
- (var3) Select Sort Variable(s)
- (var4) Select Calculation Option
- (var5) Specify Titles
- (var6) Select Output Option

Fig. 3.3.5. The TABULAR.PROGRAM screen with field names
3.3.2.4 Program returns to

TABULAR.PROGRAM

3.3.2.5 SAS window functions called by program

VARLIST Window
Legend Window

3.3.2.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>g_date</th>
<th>contains the date type chosen in TIME-PROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialized</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Set</td>
<td>afstr2</td>
<td>contains a string with the names of all the variables the user chooses</td>
</tr>
<tr>
<td></td>
<td>g_vars</td>
<td>contains a string with the names of all the variables the user chooses</td>
</tr>
</tbody>
</table>

3.3.2.7 Structured English for program code

Define the format for dates
When user presses <return> while in the first field - var1
   If var1 field does not contain a '*' or a blank then do
   Put '*' in var1 field
   Call the leg1 module
   If dataset data1 does not exist then do
      Display message 'Cannot open dataset.'
   Else do
      Call the VARLIST Window and allow user to choose one numeric data column from dataset data1 and put the users choice in the variable varsel
      Display message 'Press F10 to EXIT this screen.'
      Put value in varsel variable into the macrovariables afstr2 and g_vars
      Search varsel for the string 'PUPDATE'
      If 'PUPDATE' exists
         Call TIME.PROGRAM
         Put value in global variable g_date into local variable datesel
         If datesel does not contain a blank then do
         If datesel contains the string 'pumon' then do
Add string 'pumon' to macrovariable afstr2
Create a temporary dataset temp from dataset data1 keeping only the column names included in the macrovariable afstr2, with values for column pumon in character format
Create a temporary dataset temp1 from dataset data1a keeping only the column names included in the macrovariable afstr2, with values for column pumon in numeric format
Create permanent datasets data2 and data2a to include pumon and drop pudate columns
If dataset contains the string 'puyr' then do
Add string 'puyr' to macrovariable afstr2
Create a temporary dataset temp from dataset data1 keeping only the column names included in the macrovariable afstr2, with values for column 'puyr' in character format
Create a temporary dataset temp1 from dataset data1a keeping only the column names included in the macrovariable afstr2, with values for column 'puyr' in numeric format
Create permanent datasets data2 and data2a to include puyr and drop pudate columns
If dataset contains the string 'pumonyr' then do
Add string 'pumonyr' to macrovariable afstr2
Create a temporary dataset temp from dataset data1 keeping only the column names included in the macrovariable afstr2, with values for column pumonyr in character format
Create a temporary dataset temp1 from dataset data1a keeping only the column names included in the macrovariable afstr2, with values for column pumonyr in numeric format
Create permanent datasets data2 and data2a to include pumonyr and drop pudate columns
If dataset contains the string 'pudate' then do
Create permanent datasets data2 and data2a keeping only the columns from datasets data1 and data1a that are included in the macrovariable afstr2
Else do (if 'pudate' was not chosen)
Create permanent datasets data2 and data2a keeping only the columns from datasets data1 and data1a that are included in the macrovariable afstr2
Else position cursor in var2 field
When user presses <return> while in the second field - var2
    If var2 field does not contain a blank then do
        If var2 field is blank then do
            Display message 'You must select variables for your report.'
            Put a blank in var2 field
            Put cursor in var1 field
        Else do
            Put a '*' in var2 field
            Call RANGE.PROGRAM
    Else do
        Put cursor on the command line
        Leg1 module:
            Specifies text and color of SAS Legend Window

3.3.2.8 Notes

This is a required window.
The macrovariable g_vars is redundant and not necessary in this program.

3.3.2.9 Program screen

Figure 3.3.6 represents the screen the user will see. Figure 3.3.7 contains the field names in parentheses on the program screen.

3.3.3 TIME.PROGRAM

3.3.3.1 Program purpose

TIME.PROGRAM allows the user to select a date type for the output. The user may choose to display dates as pickup month; pickup year; pickup month and year; or as pickup day, month, and year.

3.3.3.2 Calling program

TABDATA.PROGRAM

3.3.3.3 Programs called

N/A
Fig. 3.3.6. The TABDATA.PROGRAM screen.
Data Selection

(var1) Select Variables for Report
(var2) Select Minimum/Maximum

Fig. 3.3.7. The TABDATA.PROGRAM screen with field names.
3.3.3.4 Program returns to

TABDATA.PROGRAM

3.3.3.5 SAS window functions called by program

N/A

3.3.3.6 Macrovariables

Used - N/A

Initialized - N/A

Set - g_date contains the string 'pumon', 'puyr', 'pumonyr', or 'pudate'

3.3.3.7 Structured English for program code

When user presses <return> while in the mon field
  If mon field does not contain a '*' or a blank then do
    Put a '*' in the mon field
    Position cursor on the command line
    Put string 'pumon' in macrovariable g_date
  Else position cursor in yr field
When user presses <return> while in the yr field
  If yr field does not contain a '*' or a blank then do
    Put a '*' in the yr field
    Position cursor on the command line
    Put string 'puyr' in the macrovariable g_date
  Else position cursor in monyr field
When user presses <return> while in the monyr field
  If monyr field does not contain a '*' or a blank then do
    Put a '*' in the monyr field
    Position cursor on the command line
    Put string 'pumonyr' in the macrovariable g_date
  Else position cursor in dmyr field
When user press <return> while in the dmyr field
  If dmyr field does not contain a '*' or a blank then do
    Put a '*' in the dmyr field
    Position cursor on the command line
    Put string 'pudate' in the macrovariable g_date
  Else position cursor on command line
3.3.3.8 Notes

This window is called only if the user selects pickup date as a variable to include in the report. This program is also called by PIEDATA.PROGRAM and BARDATA.PROGRAM.

3.3.3.9 Program screens

Figure 3.3.8 represents the screen the user will see. Figure 3.3.9 contains the field names in parentheses for the program screen.

```
<table>
<thead>
<tr>
<th></th>
<th>Month</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Year</td>
</tr>
<tr>
<td></td>
<td>Month/Year</td>
</tr>
<tr>
<td></td>
<td>Day/Month/Year</td>
</tr>
</tbody>
</table>
```

Fig. 3.3.8. The TIME.PROGRAM screen.

```
(m) Month
(E) Year
(monyr) Month/Year
(dmyr) Day/Month/Year
```

Fig. 3.3.9. The TIME.PROGRAM screen with field names.

3.3.4 RANGE.PROGRAM

3.3.4.1 Program purpose

RANGE.PROGRAM allows users to specify a range for any numeric data column. Data sets data2 and data2a are rewritten to contain only the data that falls within the range the user specifies.
3.3.4.2 Calling program

TABDATA.PROGRAM

3.3.4.3 Programs called

N/A

3.3.4.4 Program returns to

TABDATA.PROGRAM

3.3.4.5 SAS window functions called by program

Legend Window

3.3.4.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>-</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialized</td>
<td>-</td>
<td>N/A</td>
</tr>
<tr>
<td>Set</td>
<td>-</td>
<td>N/A</td>
</tr>
</tbody>
</table>

3.3.4.7 Structured English for program code

If dataset data2 does not exist then do
  Display message 'Cannot open dataset.'
Else do
  Call the legend module
  Call varlist function and allow user to choose one numeric data column from dataset data2 and put the user's choice in a local variable varsel
  Initialize local variables lower to '0' and upper to '99999'
When user enters range values in lower field and/or upper field and presses <return>
Read data2
Rewrite data2 as
   Select where value in column named in
   local variable varsel is between the
   values in local variables lower and upper
Read data2a
Rewrite data2a as
   Select where value in column named in
   local variable varsel is between
   the values in local variables lower and upper
Print data2
Print data2a

Leg1 module:
Specific text and color of SAS Legend Window

3.3.4.8 Notes

This screen is optional.

3.3.4.9 Program screen

Figure 3.3.10 represents the screen the user will see. Figure 3.3.11 contains the field names
in parentheses.

3.3.5 SORT.PROGRAM

3.3.5.1 Program purpose

SORT.PROGRAM allows the user to choose a first sort, a second sort, and a third sort
variable from the dataset data2. The sort variables chosen and the order in which they are
chosen control the way data are output in the report.

3.3.5.2 Calling program

TABULAR.PROGRAM

3.3.5.3 Program called

GROUP1.PROGRAM
Lower bound:  

Upper bound:  

Type over to choose upper and lower bounds.
Press F10 to exit.

Fig. 3.3.10. The RANGE.PROGRAM screen
Lower bound: (lower)
Upper bound: (upper)

Type over to choose upper and lower bounds.
Press F10 to exit.

Fig. 3.3.11. The RANGE PROGRAM screen with field names
3.3.5.4 Program returns to

From GROUP1.PROGRAM, control is returned to TABULAR.PROGRAM

3.3.5.5 SAS window functions called by program

Legend Window
VARLIST Window

3.3.5.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th></th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialized</td>
<td>g_sort1</td>
<td>initialized to blank</td>
</tr>
<tr>
<td></td>
<td>g_sort2</td>
<td>initialized to blank</td>
</tr>
<tr>
<td></td>
<td>g_sort3</td>
<td>initialized to blank</td>
</tr>
<tr>
<td></td>
<td>g_vars</td>
<td>initialized to blank</td>
</tr>
<tr>
<td>Set</td>
<td>g_sort1</td>
<td>contains a string holding the variable name for the user's first sort choice if a first sort is chosen, else it remains blank</td>
</tr>
<tr>
<td></td>
<td>g_sort2</td>
<td>contains a string holding the variable name for the user's second sort choice if a second sort is chosen, else it remains blank</td>
</tr>
<tr>
<td></td>
<td>g_sort3</td>
<td>contains a string holding the variable name for the user's third sort choice if a third sort is chosen, else it remains blank</td>
</tr>
<tr>
<td></td>
<td>g_vars</td>
<td>contains a string holding variable names for all the users sort choices if user chooses sort variables, in case the user does not make a choice in the subtotal program, else it remains blank</td>
</tr>
</tbody>
</table>

3.3.5.7 Structured English for program code

Initialize macrovariables g_sort1, g_sort2, g_sort3, and g_vars to blank
When user presses <return> while in the sort1 field
   If sort1 field does not contain a '*' or a blank then do
Put a '*' in the sort1 field
Call Leg1 module
Call varlist function and allow user to choose
   one character data column from dataset data2
   and put the user's choice in the variable varsel
Put the value in the variable varsel into the
   field s1 and into the macrovariable g_sort1
Put a blank in the variable varsel
Else position cursor in sort2 field
When user presses <return> while in sort2 field
   If sort2 field does not contain a '*' or a blank then do
      If field s1 is blank then do
         Sound alarm
         Display message 'You must select a first
         sort before a second sort.'
         Position cursor in sort1 field
         Put a blank in the sort1 field
      Else do
         Put '*' in sort2 field
         Call Leg1 module
         Call varlist function and allow user
            to choose one character data
            column from dataset data2
            and put the user's choice in
            variable varsel
         Put the value in the variable varsel into
            the field s2 and into the
            macrovariable g_sort2
         Put a blank in the variable varsel
         If fields s1 and s2 are equal then do
            Sound alarm
            Display message 'You cannot
            select the same sort variables.'
            Put a blank in the fields
            s2 and sort2
            Position cursor in the
            sort2 field
      Else position cursor in sort3 field
When user presses <return> while in sort3 field
   If sort3 field does not contain a '*' or a blank then do
      If fields s1 or s2 are blank then do
         Sound alarm
         Display message 'You must select a
         first and second sort.'
         Position cursor in the sort1 field
         Put a blank in the sort1 field
Else do
  Put a '*' in the sort3 field
  Call leg1 module
  Call varlist function and allow
    user to choose one
    character data column from
    dataset data2 and put
    user's choice in the variable varsel
  Put the value in the variable varsel into
    the field s3 and into the
    macrovariable g_sort3
  Put a blank in the variable varsel
If field s3 equals field s1 or
  field s2 then do
  Sound alarm
  Display message 'You cannot
    select the same sort variables.'
  Put a blank in the fields
    s3 and sort3
  Position cursor in field sort3
Else position cursor on command line
  Put the value in the macrovariable g_sort1 into the variable temp1
  Put the value in the macrovariable g_sort2 into the variable temp2
  Put the value in the macrovariable g_sort3 into the variable temp3
  Put the values in the variables temp1, temp2, and temp3 in the
    macrovariable g_vars with blanks between each value
Call GROUP1.PROGRAM
  Sort dataset data2a by the values (in order) in macrovariables
    g_sort1, g_sort2, and g_sort3

  leg1 module:
    Specifies the text and color of the SAS Legend Window

3.3.5.8 Notes

  This is an optional program.
  More sort variables could be added if needed.
  The macrovariable g_sort3 is not used in any later program, but it could be if needed.

3.3.5.9 Program screens

  Figure 3.3.12 represents the screen the user will see. Figure 3.3.13 contains the field names
  in parentheses for the program screen
Fig. 3.3.12. The SORT.PROGRAM screen.
**Sort Variables**

<table>
<thead>
<tr>
<th>(sort1)</th>
<th>Sort 1</th>
<th>(s1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(sort2)</td>
<td>Sort 2</td>
<td>(s2)</td>
</tr>
<tr>
<td>(sort3)</td>
<td>Sort 3</td>
<td>(s3)</td>
</tr>
</tbody>
</table>

Fig. 3.3.13. The SORT.PROGRAM screen with field names.
3.3.6 GROUP1.PROGRAM

3.3.6.1 Program purpose

GROUP1.PROGRAM allows users to choose to subtotal the results in the report based on the sort variables chosen in SORT.PROGRAM. The screen offers the user two options. The user may choose to subtotal on the first sort variable, which will appear on the screen, or the first and second sort variables, which will also appear on the screen.

3.3.6.2 Calling program

SORT.PROGRAM

3.3.6.3 Programs called

N/A

3.3.6.4 Program returns to

TABDATA.PROGRAM

3.3.6.5 SAS window functions called by program

N/A

3.3.6.6 Macrovariables

Used -

g_sort1 contains a string holding the first sort variable the user chose in SORT.PROGRAM

g_sort2 contains a string holding the second sort variable the user chose in SORT.PROGRAM or a blank

g_vars contains a string holding the column names for all the user's sort choices chosen in SORT.PROGRAM

Initialized -

g_group initialized to blank

Set -

g_group contains a string holding the word 'by' and the user's subtotal variable choices, separated by blanks
3.3.6.7 Structured English for program code

Create variable sort1 as the value in the macrovariable g_sort1
Create variable sort2 as the value in the macrovariable g_sort1
Create variable sort3 as the value in the macrovariable g_sort2
Initialize the macrovariable g_group to a blank
When the user presses <return> while in the first field - g2
  If g2 field does not contain a '*' or a blank then do
    If g3 field does not contain a '*' then do
      Put a '*' in the g2 field
      Put the word 'by', a blank space, and the value in the variable sort1 into the macrovariable g_group
      Put the value in the variable sort1 into the macrovariable g_vars
      Display message 'Press F10 to EXIT.'
    Else
      Sound alarm
      Display message 'You may only select one subtotal option.'
      Put a blank in g2 field
      Else position cursor in g3 field
      When user presses <return> while in second field - g3
        If g3 field does not contain a '*' or a blank then do
          If g2 field does not contain a '*' then do
            Put a '*' in g3 field
            Put the word 'by', a blank space, the value in the variable sort2, a blank space, and the value in the variable sort3 into the macrovariable g_group
            Put the values in the variables sort2 and sort3 separated by a blank space into the macrovariable g_vars
            Display message 'Press F10 to EXIT.'
          Else
            Sound alarm
            Display message 'You may only select one subtotal option.'
            Put a blank in the g3 field
            Else position cursor on the command line
      Else
        When the user exits the screen, if fields g2 and g3 are blank then put a blank in the macrovariable g_group
3.3.6.8 Notes

This is a required program if the user chooses the optional SORT.PROGRAM. Additional subtotal options could be added to include additional sort variables.

3.3.6.9 Program screens

Figure 3.3.14 represents the screen the user will see. Figure 3.3.15 contains the field names in parentheses for the program screen.

3.3.7 TOPTION.PROGRAM

3.3.7.1 Program purpose

TOPTION.PROGRAM allows the user to select a calculation type. From this screen the user may choose to calculate the total, the average, or the percent for the report. If the user chooses to average the data, STDDEV.PROGRAM and AVG.PROGRAM are called.

3.3.7.2 Calling program

TABULAR.PROGRAM

3.3.7.3 Programs called

May call STDDEV.PROGRAM and AVG.PROGRAM

3.3.7.4 Program returns to

TABULAR.PROGRAM

3.3.7.5 SAS window functions called by program

VARLIST Window
Legend Window
Fig. 3.3.14. The GROUP1.PROGRAM screen.
Subtotal Options

<table>
<thead>
<tr>
<th></th>
<th>(sort1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(g2)</td>
<td>--------------</td>
</tr>
<tr>
<td>(g3)</td>
<td>(sort2)</td>
</tr>
</tbody>
</table>

Fig. 3.3.15. The GROUP1.PROGRAM screen with field names.
3.3.7.6 Macrovariables

Used - N/A

Initialized - N/A

Set -

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>g_calopt</td>
<td>contains calculation option user chooses (TOTAL, AVERAGE, PERCENT)</td>
</tr>
<tr>
<td>g_calvar</td>
<td>contains the numeric variable chosen from data2 in the SAS Varlist window</td>
</tr>
</tbody>
</table>

3.3.7.7 Structured English for program code

When user presses <return> while in the total field
   If total field does not contain a '*' or a blank then do
      If avg and percent fields contain blanks then do
         Put a '*' in total field
         Put word 'TOTAL' in the macrovariable g_calopt
         Call dataset module
         Position cursor in total field
         Exit to TABULAR.PROGRAM
      Else sound alarm and display message 'Only one option allowed.'
      Else position cursor in avg field
   Else position cursor in total field

When user presses <return> while in the avg field
   If avg field does not contain a '*' or a blank then do
      If total and percent fields contain blanks then do
         Put a '*' in the avg field
         Put word 'AVERAGE' in the macrovariable g_calopt
         Call dataset module
         Call STDDEV.PROGRAM
         Call AVG.PROGRAM
         Position cursor in avg field
         Exit to TABULAR.PROGRAM
      Else sound alarm and display message 'Only one option allowed.'
      Else position cursor in percent field
   Else position cursor in avg field

When user presses <return> while in the percent field
   If percent field does not contain a '*' or a blank then do
      If total and avg fields contain blanks then do
         Put a '*' in the percent field
         Put word 'PERCENT' in the macrovariable g_calopt
Call dataset module
  Position cursor in percent field
  Exit to TABULAR.PROGRAM
Else sound alarm and display message 'Only one option allowed.'
  Put blank in percent field
  Position cursor in percent field
Else put cursor on command line

Datasel module:
  If dataset data2 does not exist then do
    Display message 'Cannot open data set.'
  Else
    Call leg1 module
    Give legend window the title 'Select one variable for computation.'
    Call getvar module
Leg1 module:
  Specifies text and color of SAS Legend Window
Getvar module:
  Specifies a SAS varlist window where user may choose one numeric column from the dataset data2
  Puts variable chosen in SAS varlist window into g_calvar

3.3.7.8 Notes

This is a required window.
Only one calculation option can be chosen in one session.

3.3.7.9 Program screens

Figure 3.3.16 represents the screen the user will see. Figure 3.3.17 contains the field names in parentheses on the program screen.

Calculation Options

____ Total
____ Average
_____ Percent

Fig. 3.3.16. The TOPTION.PROGRAM screen.
3.3.8 STDDEV.PROGRAM

3.3.8.1 Program purpose

STDDEV.PROGRAM allows the user to choose whether or not to include the standard error data on the report. It is called only if the user chooses 'AVERAGE' in the TOPTION.PROGRAM. The default values is 'Y' for yes. If the user does not want to include the standard error, the user may type an 'N' over the 'Y'.

3.3.8.2 Calling program

TOPTION.PROGRAM

3.3.8.3 Programs called

N/A

3.3.8.4 Program returns to

TOPTION.PROGRAM

3.3.8.5 SAS window functions called by program

N/A
3.3.8.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialized</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Set - g_stddev contains a blank if the user does not want to include the standard error or the string 'stderr = StdError' if the user does want to include the standard error.

3.3.8.7 Structured English for program code

If the choice field contains a 'Y' then do
- Put the string 'stderr = StdErr' into the variable string
- Put the value in the variable string into the macrovariable g_stddev

Else if the choice field contains an 'N' then do
- Put a blank into the variable string
- Put the value in the variable string (the blank) into the macrovariable g_stddev

Else do
- Sound alarm
- Display message "You must enter 'Y' or 'N'."
- Put 'Y' in the choice field
- Put the cursor in the choice field

3.3.8.8 Notes

This is a mandatory program if the user chooses 'AVERAGE' in TOPTIONS.PROGRAM.

The 'Y' is on the screen initially because in the attributes window the initial value is defined to be 'Y'.

3.3.8.9 Program screens

Figure 3.3.18 represents the screen the user will see. Figure 3.3.19 contains the field names in parentheses for the program screen.
Include the standard error of the mean.

Type 'N' over 'Y' to omit the standard error.

Fig. 3.3.18. The STDDEV.PROGRAM screen.

(Choice) Include the standard error of the mean.

Type 'N' over 'Y' to omit the standard error.

Fig. 3.3.19. The STDDEV.PROGRAM screen with field name.

3.3.9 AVG.PROGRAM

3.3.9.1 Program purpose

AVG.PROGRAM allows the user to create a dataset file that contains the averages of the data, including or omitting the standard error, according to the user's choice.

3.3.9.2 Calling program

TOPTION.PROGRAM

3.3.9.3 Programs called

N/A

3.3.9.4 Program returns to

TOPTION.PROGRAM
3.3.9.5 SAS window functions called by program

N/A

3.3.9.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>g-calvar contains user chosen calculation variable from TOPTION.PROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>g-group contains user chosen subtotal variables from GROUP1.PROGRAM</td>
</tr>
<tr>
<td></td>
<td>g-stddev contains command to include standard error or is blank</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Initialized</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set</td>
<td>N/A</td>
</tr>
</tbody>
</table>

3.3.9.7 Structured English for program code

Put value in macrovariable g-group into local variable str
If str is null then do
  Calculate average using dataset data2a and the value in g-calvar
  Output dataset data3 including the meaning a new variable vmeans and
  if g-stddev is not null then do include the standard error as a new variable in the dataset data3
Else do
  Sort data2a by the value in g-group
  Calculate the average using dataset data2a and the value in g-calvar
  Output dataset data3 including the meaning as a new variable, vmeans, and
  if g-stddev is not null then do include the standard error as a new variable in the dataset data3
3.3.9.8 Notes

The second part of this program could be tuned to run faster by checking to see if data was sorted in the optional SORT.PROGRAM. Data does not have to be sorted twice.

This program is called if the user chooses 'AVERAGE' in the TOPTIONS screen. There is no user window with this program. It is invisible to the user.

3.3.9.9 The program screen

There is no user window with this program.

3.3.10 TOUTOPT.PROGRAM

3.3.10.1 Program purpose

TOUTOPT.PROGRAM allows the user to choose the destination for the output of a report. The user may choose to send the output to a printer, to a file, or to the monitor screen.

3.3.10.2 Calling program

TABULAR.PROGRAM

3.3.10.3 Program called

TREADY.PROGRAM

3.3.10.4 Program returns to

TABULAR.PROGRAM

3.3.10.5 SAS window functions called by program

N/A

3.3.10.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialized</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Set - g_outopt contains a string holding the output destination option ('screen' or 'print') the user has chosen

3.3.10.7 Structured English for program code

Put a blank in the print field, the file field, and the screen field
When user presses <return> while in the print field
  If print field does not contain a '*' or a blank then do
    Put a '*' in the print field
    Display message 'Press F10 when finished.'
    Put value 'print' into the variable string
    Put value in string variable into the macrovariable g_outopt
    Call TREADY.PROGRAM
    Put a blank in the print field
    Position cursor on the print field
  Else position cursor on the file field
When user presses <return> while in the file field
  If print field does not contain a '*' or a blank then do
    Put a '*' in the file field
    Display message 'This option is not available in this version.'
    Put a blank in the file field
    Position cursor on the file field
  Else position cursor on the screen field
When user presses <return> while in the screen field
  If screen field does not contain a '*' or a blank then do
    Put a '*' in the screen field
    Put value 'screen' into the variable string
    Put value in string variable into the macrovariable g_outopt
    Call TREADY.PROGRAM
    Position cursor on the screen field
    Put a blank in the screen field
  Else position cursor on the command line

3.3.10.8 Notes

The option for saving output to a file is not functional in the prototype. If the user chooses this option, he/she will see a message that explains that this function is not available in this version.

3.3.10.9 Program screens

Figure 3.3.20 represents the screen the user will see. Figure 3.3.21 contains the field names in parentheses on the program screen.
3.3.11 TREADY.PROGRAM

3.3.11.1 Program purpose

TREADY.PROGRAM sends the output to the destination chosen by the user in TOUTOPT.PROGRAM. It also performs data and procedure steps using dataset data2a if the user chose 'PERCENT' in TOPTION.PROGRAM. If the user chose 'AVERAGE' in TOPTION.PROGRAM, dataset data2a is read by the program, and data3 is created.

3.3.11.2 Calling program

TOUTOPT.PROGRAM
3.3.11.3 Programs called

N/A

3.3.11.4 Program returns to

TOUTOPT.PROGRAM

3.3.11.5 SAS window functions called by program

N/A

3.3.11.6 Macrovariables

Used

- g_calopt contains a string holding the calculation option chosen by the user in TOPTION.PROGRAM (i.e., 'TOTAL', 'AVERAGE', or 'PERCENT')

- g_calvar contains a string holding the numeric variable chosen in TOPTION.PROGRAM

- g_group contains a string holding the word 'by' and the user's subtotal variable choices separated by blanks chosen in GROUP1.PROGRAM

- g_outopt contains a string holding the user's output destination choice (i.e., 'screen' or 'print')

- g_sort1 contains a string holding the first sort variable the user chose in SORT.PROGRAM

- g_sort2 contains a string holding the first sort variable the user chose in SORT.PROGRAM

- g_sort3 contains a string holding the third sort variable the user chose in SORT.PROGRAM

- g_str1 see macrovariables set in this program

- g_str2 see macrovariables set in this program
g_tot  
see macrovariables set in this program

g_vars  
contains a string holding the variable names for all the user's sort choices separated by blanks, if the user chose sort variables in SORT.PROGRAM, or it contains a blank

Initialized - N/A

Set - g_str1  
contains a string holding one of two commands depending on whether the user chose 'print' or 'screen' in TOUTOPT.PROGRAM (i.e., 'dm "zoom on; output; clear; top" output', or 'proc printto')

    g_str2  
contains a string holding a command or a blank depending on whether the user chose 'print' or 'screen' in TOUTOPT.PROGRAM (i.e., ' ', or 'proc printto new print = lpr; run')
    (this is only set if the user chose 'PERCENT' in TOPTION.PROGRAM)

    g_tot  
    (this is only set if the user chose 'PERCENT' in TOPTION.PROGRAM) contains the total when all values for the variable g_calvar have been added together.

    g_vars  
    (this is only set if the user chose 'PERCENT' in TOPTION.PROGRAM) contains a string holding the variable names for all the user's sort choice separated by blanks if the user chose sort variables in SORT.PROGRAM, or it contains a blank; a blank, the word 'wsum', another blank, and the word 'percent' are then concatenated with that value

3.3.11.7 Structured English for program code

Create variable optstr as the value in the macrovariable g_calopt
Create variable outstr as the value in the macrovariable g_outopt
If the value in outstr is 'screen' then do
    Set macrovariable g_str1 to contain the string 'dm "zoom on; output; clear; top" output'
    Set macrovariable g_str2 to contain a blank
Display message 'Enter AF to return to this screen then press F10 to exit.'
If the value in outstr is 'print' then do
  Set macrovariable g_str1 to contain the string 'proc printto'
  Set macrovariable g_str2 to contain the string 'proc printto'
  new print = 1pr; run'
  Display message 'Press F10 to exit.'
If the value in opstr is 'TOTAL' then do
  Set options
  If the value in g_str2 is blank then do
    continue
  If the value in g_str2 is a command string then do
    Direct output to the printer defined as 1pr
    Print the dataset data2a without observation numbers
    Group by the variable names in the macrovariable g_group
    Sum the values in the variable named in the
    macrovariable g_calvar
  Read the value in the macrovariable g_str1
  If the value in the macrovariable g_str1 is
    'dm "zoom on; output; clear; top "output" then do
      Go to the display manager windows
      Zoom the window, go to the output section,
      clear it, go to the top
    If the value in the macrovariable g_str1 is
      'proc printto' then do
        Set the output destination back to the
        screen
  If the value in opstr is 'AVERAGE' then do
    Set options
    If the value in g_str2 is blank then do
      continue
    If the value in g_str2 is a command string then do
      Direct output to the printer defined as 1pr
      Print the dataset data3a without observation numbers
      Drop the variable _TYPE_
    Read the value in the macrovariable g_str1
    If the value in the macrovariable g_str1 is
      'dm "zoom on; output; clear; top "output'
      then do
        Go the display manager windows
        Zoom the window, go the output section,
        clear it, go to the top
    If the value in the macrovariable g_str1 is
      'proc printto' then do
        Set the output destination back to the screen
  If the value in opstr is 'PERCENT' then do
    Create variable string as the value in the macrovariable g_vars
    which is either a blank or a list or variable names chosen in
    SORT.PROGRAM
Set macrovariable g_vars to contain the value in the variable string concatenated with a blank, the word 'wsum', another blank, and the word 'percent'.

Create variable temp as the value in the macrovariable g_group which contains the user's subtotal variable choices from GROUP1.PROGRAM.

Create a variable temp1 as the value in the macrovariable g_sort1 which contains the first sort variable the user chose in SORT.PROGRAM.

If the variable temp contains a blank (if the user did not choose a subtotal variable) then do:

If temp1 is not blank (if user did choose a sort variable) then do:

Sort dataset data2a by the variables in the macrovariables g_sort1, g_sort2, and g_sort3.

Create a temporary dataset temp.

Else:

Create a temporary dataset temp using dataset data2a.

Create a temporary dataset new using dataset temp.

Create a new variable wsum as the values in the variable wt.

Drop the variable wt.

Print the dataset.

Else do (if the user did choose a subtotal variable):

Sort the dataset data2a by the values in the macrovariable g_group (the user's subtotal choices).

Create a temporary dataset temp1.

Calculate the means using dataset temp1 by the value in the macrovariable g_group (the user's subtotal choices) on the variable name in the macrovariable g_calvar (the user's calculation variable choice).

Create a new temporary dataset new that includes the sum of the values in the variable wsum.

Set options.

Go through a loop to add all the values in the variable named in the macrovariable g_calvar.

Do not create a new dataset but use dataset data2a.

Create a variable end as the SAS end-of-file indicator.

If this is the first observation in the dataset data2a then do:

Create a variable tot as the first value in the variable named in the macrovariable g_calvar.

Else do:

Set tot as the value in the variable tot plus the next value in the variable named in the macrovariable g_calvar.

Do not reinitialize the variable tot to zero.

If this is the last observation in the dataset data2a then do:

Create macrovariable g_tot as the final value in the variable tot.
Create a new dataset new1 using the dataset new, keeping the variables named in the macrovariable g_vars
Sort by the variables named in the macrovariable g_group
Create a variable percent as the value in wsum divided by the value in the macrovariable g_tot multiplied times 100 (percent = the subtotal of g_calvar by the g_group variable divided by the total value of all the values of the variable contained in g_calvar)
If the value in the macrovariable g_str2 is blank then do
Continue
If the value in the macrovariable g_str2 is a command string then do
Direct output to the printer defined as lpr
Print the dataset new1
Read the value in the macrovariable g_str2
If the value is 'dm "zoom on; output; clear; top "output' then do
Go to the display manager windows
Zoom the window, go to the output section, clear it, go to the top
If the value is 'proc printto' then do
Set the output destination back to the screen

3.3.11.8 Notes

This screen will display a message as the user waits for output to the screen or printer. Right now, the 'percent' calculation code assumes the user wants a percentage of weight. As new numeric variables are added to the SAS dataset files based on the ORACLE queries, this will need to be changed to a variable name. The code now contains the variable name 'wt'. Also, this discussion assumes the printer being used is defined as 'lpr'. This will change depending on the printer in use.

3.3.11.9 Program screen

There are no enterable fields in this program screen. A full-sized screen will appear with a message and instructions depending on whether the user chose 'screen' or 'print' in TOUTOPT.PROGRAM. Figure 3.3.22 represents the screen the user will see.

```
Processing — One moment please.
```

Fig. 3.3.22. The TREADY.PROGRAM screen.
3.4 BAR CHART PROGRAMS

The bar chart programs are called when the user specifies either 'Vertical Bar' or 'Horizontal Bar' on the MENU.PROGRAM screen. The first screen the user will see is BAR.PROGRAM. This screen allows the user to call one of the following screens:

- BARDATA.PROGRAM,
- BOPTION.PROGRAM,
- The SAS titles function window, or
- BOUTOPT.PROGRAM.

Figure 3.4.1 is a hierarchy chart of the programs that are components of the bar program set. BARDATA.PROGRAM allows the user to choose an X-axis variable and a calculation variable from the original dataset for inclusion in the report. It is a mandatory screen. BARDATA.PROGRAM may also call TIME.PROGRAM, BRANGE.PROGRAM, and/or BMRANGE.PROGRAM. If the user chooses the variable 'PICKUP DATE' in BARDATA.PROGRAM, TIME.PROGRAM is called. TIME.PROGRAM allows the user to specify the way he/she wants to look at time (i.e., by year; by month and year; or by day, month, and year). BRANGE.PROGRAM, which allows the user to specify one set of minimum and maximum values is called if the user specifies 'WEIGHT' as the calculation variable. BMRANGE.PROGRAM, which allows the user to specify up to three minimum and maximum boundary groups, is called if the user specifies 'WEIGHT' as the X-axis variable.

BOPTION.PROGRAM allows the user to choose a calculation type for the chart. It is a mandatory screen. The choices are 'AVERAGE,' 'TOTAL,' and 'PERCENT.' The choice the user makes is stored in a macrovariable and used by BREADY.PROGRAM.

The SAS titles function window is optional. The user may specify up to ten lines for a title to be included in the chart.

BOUTOPT.PROGRAM allows the user to direct the output. The output may be directed to a printer or to the screen. BOUTOPT.PROGRAM calls BREADY.PROGRAM.

BREADY.PROGRAM creates the bar chart and sends the output to the destination specified by the user. It also does calculations and creates a new dataset file if the user chose 'PERCENT' in BOPTION.PROGRAM.

Each of these programs may use macrovariables that have been initialized or set from another program or the programs may themselves initialize or set macrovariables that may be used by subsequent programs. Figure 3.4.2 shows the macrovariables that are initialized, set, or used by each of the programs that make up the bar program set. Figure 3.4.3 shows the dataset file(s) that is used as input and the dataset file(s) that is written when the program is executed. In some cases no dataset file is read and/or no new dataset file is written during execution of a program.

3.4.1 BAR.PROGRAM

3.4.1.1 Program purpose

BAR.PROGRAM allows the user to select options for producing a bar chart. From this screen the user chooses to call other screens for the purpose of selecting data columns for the chart, selecting a calculation type, specifying a title, and selecting the output destination.
Fig. 3.4.1. Hierarchy of programs in the bar program set.
Fig. 3.4.2. Hierarchy of bar programs showing macrovariables set, used, or initialized in the programs.
Fig. 3.4.3. Hierarchy of bar programs showing permanent SAS datasets read or created.
3.4.1.2 Calling program

MENU.PROGRAM

3.4.1.3 Programs Called

BARDATA.PROGRAM
BOPTION.PROGRAM
BOUTOPT.PROGRAM

3.4.1.4 Program returns to

MENU.PROGRAM

3.4.1.5 SAS window functions called by program

SAS Titles Window

3.4.1.6 Macrovariables

Used - N/A
Initialized - N/A
Set - N/A

3.4.1.7 Structured English for program code

When user presses <return> while in first field - var1
   If there is not a '*' or a blank in the var1 field
      then do
         Put a '*' in the var1 field
         Call BARDATA.PROGRAM
   Else position cursor in var2 field
When the user presses <return> while in second field - var2
   If there is not a '*' or a blank in the var2 field
      then do
         Put a '*' in the var2 field
         Call BOPTION.PROGRAM
   Else position cursor in var3 field
When the user presses <return> while in third field - var3
   If there is not a '*' or a blank in the var3 field
then do
  Put a '*' in the var3 field
  Display message 'Press RETURN then fill in titles. Press F10 when finished.'
  Call the SAS Titles Window
Else position cursor in the var4 field
When the user presses <return> while in the fourth field - var4
If there is not a '*' or a blank in the var4 field
then do
  Put a '*' in the var4 field
  Call BOUTOPT.PROGRAM
Else position cursor on the command line

3.4.1.8 Notes

This window is required when producing bar charts.

3.4.1.9 Program screens

Figure 3.4.4 represents the screen the user will see. Figure 3.4.5 contains the field names in parentheses on the program screen.

| Output Selection |
| --- | --- |
| _ | Select Variables for Analysis |
| _ | Select Calculation Option |
| _ | Specify Titles |
| _ | Select Output Option |

Enter 'X' beside your choice.
Press RETURN.
Press F10 to exit this screen.

Fig. 3.4.4. The BAR.PROGRAM screen.
Output Selection

(var1) Select Variables for Analysis
(var2) Select Calculation Option
(var3) Specify Titles
(var4) Select Output Option

Enter 'X' beside your choice.
Press RETURN.
Press F10 to exit this screen.

Fig. 3.4.5. The BAR.PROGRAM screen with field names.

3.4.2 BARDATA.PROGRAM

3.4.2.1 Program purpose

BARDATA.PROGRAM allows the user to select the column to be used for the X-axis variable and the column to be used for the calculation variable to produce a bar chart. This program will call TIME.PROGRAM if the user has chosen the column 'PDATE' as the X-axis variable. It will call the BMRANGE.PROGRAM if the user chooses a numeric column for either the X-axis or the calculation variables. If the user chooses 'WEIGHT' as the X-axis variable, this program will call BRANGE.PROGRAM. This program uses dataset data1 for all functions except for the time functions, where it uses dataset data1a.

3.4.2.2 Calling program

BAR.PROGRAM

3.4.2.3 Programs called

BMRANGE.PROGRAM
BRANGE.PROGRAM
TIME.PROGRAM
3.4.2.4 Program returns to

BAR.PROGRAM

3.4.2.5 SAS window functions called by program

Legend Window
VARLIST Window

3.4.2.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>g_date</th>
<th>contains the date type chosen in the TIME.PROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>g_group1</td>
<td>contains a string that is used to label the first minimum and maximum values; set in the BMRANGE.PROGRAM</td>
</tr>
<tr>
<td></td>
<td>g_group2</td>
<td>contains a string that is used to label the second minimum and maximum values; set in the BMRANGE.PROGRAM</td>
</tr>
<tr>
<td></td>
<td>g_group3</td>
<td>contains a string that is used to label the third minimum and maximum values; set in the BMRANGE.PROGRAM</td>
</tr>
<tr>
<td></td>
<td>g_high</td>
<td>contains the maximum value chosen in BRANGE.PROGRAM</td>
</tr>
<tr>
<td></td>
<td>g_low</td>
<td>contains the minimum value chosen in BRANGE.PROGRAM</td>
</tr>
<tr>
<td></td>
<td>g_max1</td>
<td>contains the first maximum value chosen in the BMRANGE.PROGRAM</td>
</tr>
<tr>
<td></td>
<td>g_max2</td>
<td>contains the second maximum value chosen in the BMRANGE.PROGRAM</td>
</tr>
<tr>
<td></td>
<td>g_max3</td>
<td>contains the third maximum value chosen in the BMRANGE.PROGRAM</td>
</tr>
<tr>
<td></td>
<td>g_min1</td>
<td>contains the first minimum value chosen in the BMRANGE.PROGRAM</td>
</tr>
</tbody>
</table>
g_min2 contains the second minimum value chosen in the BM RANGE.PROGRAM

g_min3 contains the third minimum value chosen in the BM RANGE.PROGRAM

Initialized -

g_high initialized to 99999

g_low initialized to 0

Set -

g_calvar contains a string with the name of the calculation variable the user chooses

g_label contains a string with the text 'label group = "str1"'

g_str1 contains a string with the name of the X-axis variable the user chooses

g_xvar contains a string with the name of the X-axis variable the user chooses

3.4.27 Structured English for program code

Set the length for variables str1 and str2
Initialize the macrovariables g_low to '0' and g_high to '99999'

When user presses <return> while in first field - var1

If var1 field does not contain a '*' or a blank

then do

Put a '*' in the var1 field
Call leg1 module
If dataset datal does not exist then do

Display message 'Data set not found. Press F10'

Else do

Call the varlist function and allow user to choose one data column from dataset datal and put the users choice in the variable str1

If the value in str1 is the same as the value in str2 then do

Display message 'Error: You chose the same variable for calculation variable'

Else do

Put the value in str1 variable into the macrovariable g_str1
Put the position of the column the user has chosen into the num variable
Put the type of column the user has chosen into the vtype variable
Close the dataset data1
Close the legend window
If the value in the vtype variable is an 'N' and the value in the str1 variable is 'WT' then do
Call the mult-range module
Else if the value in the str1 variable is 'PUPDATE' then do
  Call the handle-date module
Else do
  Create a temporary dataset temp
  Put the value in str1 variable into the macrovariable g_xvar
  Position cursor in the var2 field
When the user presses <return> while in second field - var2
  If var2 field does not contain an 'X' or a blank then do
    Put a '*' in the var2 field
    Call the leg1 module
    If dataset data1 does not exist then do
      Display message 'Data set not found. Press F10'
    Else do
      Call the varlist function and allow user to choose one numeric data column from the dataset data1 and put the users choices in the variable str2
      If the value in str1 (the X-axis variable) and str2 (the calculation variable) are the same then do
        Display error message 'Error: You chose the same variable for X-axis variable'
        Position the cursor in the var2 field
      Else do
        Put the value in str2 variable into the macrovariable g_calvar
        If the value in the str2 variable is 'WT' then call the BRANGE.PROGRAM
        Close the dataset data1
        Close the legend window
        Else position cursor on the command line
When the user exits this screen then do
  Use the temporary dataset temp to create a new dataset data2 that contains data sorted by the column contained in the macrovariable g_xvar and contains only the values for the g_calvar macrovariable that are between the values contained in the macrovariables g_low and g_high
leg1 module:
  Specifies text and color for the legend window

mult_range module:
  Call the BMRANGE.PROGRAM
  Put the text 'label group = "str1"' into the variable string
  Put the value in string into the macrovariable g_label
  Create a temporary dataset temp from the dataset data1
    with an additional column named group
  Format the group variable in the dataset to be 11 characters
  If the values in the column named in the g_str1 macrovariable are between the
    values in the macrovariables g_min1 and g_max1 then
    Put the value in the g_group1 macrovariable into
      the variable string
      Put the value in string into the variable group
  If the values in the column named in the g_str1 macrovariable are between the
    values in the macrovariables g_min2 and g_max2 then
    Put the value in the g_group2 macrovariable into
      the variable string
      Put the value in string into the variable group
  If the values in the column named in the g_str1 macrovariable are between the
    values in the macrovariable g_min3 and g_max3 then
    Put the value in the g_group3 macrovariable into the variable string
    Put the value in string into the variable group
  Create the dataset temp with the text contained in the
    g_label macrovariable as an argument
  Put the value in the group variable into the macrovariable
    g_xvar

handle_date:
  Call the TIME.PROGRAM
  Put the value in the g_date macrovariable into the str1 variable
  If the value in str1 variable is 'pumon' then do
    Create a temporary dataset temp from dataset data1a
    Add the column pumon to the temp dataset and
    convert the values for this column into month format
  If the value in str1 variable is 'puyr' then do
    Create a temporary dataset temp from dataset data1a
    Add the column puyr to the temp dataset and
    convert the values for this column into year format
  If the value in str1 is 'pumonyr' then do
    Create a temporary dataset temp from dataset data1a
Add the column pumonyr to the temp dataset and convert the values for this column into month/year format.

If the value in str1 is 'pudate' then do

Create a temporary dataset temp from dataset data1a

Put the value in the str1 variable into the g_xvar macrovariable

3.4.2.8 Notes

This is a required window.

3.4.2.9 Program screens

Figure 3.4.6 represents the screen the user will see. Figure 3.4.7 contains the field names in parentheses on the program screen.

![Data Selection](Fig. 3.4.6. The BARDATA.PROGRAM screen.)
3.4.3 TIME.PROGRAM

3.4.3.1 Program purpose

TIME.PROGRAM allows the user to select a date type for the output. The user may choose to display dates as pickup month; pickup year; pickup month and year; or as pickup day, month, and year.

3.4.3.2 Calling program

BARDATA.PROGRAM

3.4.3.3 Programs Called

N/A

3.4.3.4 Program returns to

BARDATA.PROGRAM

3.4.3.5 SAS window functions called by program

N/A
3.4.3.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>-</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialized</td>
<td>-</td>
<td>N/A</td>
</tr>
<tr>
<td>Set</td>
<td>-</td>
<td>g_date contains the string 'pumon', 'puyr', 'pumonyr', or 'pudate'</td>
</tr>
</tbody>
</table>

3.4.3.7 Structured English for program code

When user presses <return> while in the mon field

If mon field does not contain a '*' or a blank then do

- Put a '*' in the mon field
- Position cursor on the command line
- Put string 'pumon' in macrovariable g_date

Else position cursor in yr field

When user presses <return> while in the yr field

If yr field does not contain a '*' or a blank then do

- Put a '*' in the yr field
- Position cursor on the command line
- Put string 'puyr' in the macrovariable g_date

Else position cursor in monyr field

When user presses <return> while in the monyr field

If monyr field does not contain a '*' or a blank then do

- Put a '*' in the monyr field
- Position cursor on the command line
- Put string 'pumonyr' in the macrovariable g_date

Else position cursor in dmyr field

When user presses <return> while in the dmyr field

If dmyr field does not contain a '*' or a blank then do

- Put a '*' in the dmyr field
- Position cursor on the command line
- Put string 'pudate' in the macrovariable g_date

Else position cursor on command line

3.4.3.8 Notes

This program is also used by TABDATA.PROGRAM and PIEDATA.PROGRAM.
3.4.3.9 Program screens

Figure 3.4.8 represents the screen the user will see. Figure 3.4.9 contains the field names in parentheses for the program screen.

```
<table>
<thead>
<tr>
<th></th>
<th>Month</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Year</td>
</tr>
<tr>
<td></td>
<td>Month/Year</td>
</tr>
<tr>
<td></td>
<td>Day/Month/Year</td>
</tr>
</tbody>
</table>
```

Fig. 3.4.8. The TIME.PROGRAM screen.

```
(mon)   Month
(yr)    Year
(monyr) Month/Year
(dmyr)  Day/Month/Year
```

Fig. 3.4.9. The TIME.PROGRAM screen with field names.

3.4.4 BRANGE.PROGRAM

3.4.4.1 Program purpose

BRANGE.PROGRAM allows the user to specify one minimum and maximum boundary for data values for the calculation variable of a bar chart. BRANGE.PROGRAM is called if the user chooses 'weight' as the calculation variable.

3.4.4.2 Calling program

BARDATA.PROGRAM
3.4.4.3 Programs called

N/A

3.4.4.4 Program returns to

BARDATA.PROGRAM

3.4.4.5 SAS window functions called by program

N/A

3.4.4.6 Macrovariables

<table>
<thead>
<tr>
<th></th>
<th>Used</th>
<th>Initialized</th>
<th>Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N/A</td>
<td>N/A</td>
<td>g_high</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>g_low</td>
</tr>
</tbody>
</table>

- g_high contains the maximum value
- g_low contains the minimum value

3.4.4.7 Structured English for program code

When this program is called
Put a '0' in the lower field
Put '9999' in the upper field

When the user exists this program
Put the value in the lower field into the g_low macrovariable
Put the value in the upper field into the g_high macrovariable

3.4.4.8 Notes

This is an optional window.
This program is also called by the LINEDATA.PROGRAM and the PIEDATA.PROGRAM.
3.4.4.9 Program screens

Figure 3.4.10 represents the screen the user will see. Figure 3.4.11 contains the field names in parentheses on the program screen.

Fig. 3.4.10. The BRANGE.PROGRAM screen.

Fig. 3.4.11. The BRANGE.PROGRAM screen with field names.

3.4.5 BMRANGE.PROGRAM

3.4.5.1 Program purpose

BMRANGE.PROGRAM allows the user to specify up to three minimum and maximum boundaries for data values for the X-axis of a bar chart. BMRANGE is called if the user chooses 'weight' as the X-axis variable.
3.4.5.2 Calling program

BARDATA.PROGRAM

3.4.5.3 Programs called

N/A

3.4.5.4 Program returns to

BARDATA.PROGRAM

3.4.5.5 SAS window functions called by program

N/A

3.4.5.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>-</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialized</td>
<td>-</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Set -

- **g_group1** contains a string that is used to label the first minimum and maximum values
- **g_group2** contains a string that is used to label the second minimum and maximum values
- **g_group3** contains a string that is used to label the third minimum and maximum values
- **g_max1** contains the first maximum value
- **g_max2** contains the second maximum value
- **g_max3** contains the third maximum value
- **g_min1** contains the first minimum value
- **g_min2** contains the second minimum value
- **g_min3** contains the third minimum value
3.4.5.7 Structured English for program code

When the user exits this program
   If the min1 field does not contain a blank then
      Put the value in the min1 field into the macrovariable g_min1
   Else put a 0 into the g_min1 macrovariable
   If the max1 field does not contain a blank then
      Put the value in the max1 field into the macrovariable g_max1
   Else put a 0 into the g_max1 macrovariable
   If the min2 field does not contain a blank then
      Put the value in the min2 field into the macrovariable g_min2
   Else put a 0 into the g_min2 macrovariable
   If the max2 field does not contain a blank then
      Put the value in the max2 field into the macrovariable g_max2
   Else put a 0 into the g_max2 macrovariable
   If the min3 field does not contain a blank then
      Put the value in the min3 field into the macrovariable g_min3
   Else put a 0 into the g_min3 field
   If the max3 field does not contain a blank then
      Put the value in the max3 field into the macrovariable g_max3
   Else put a 0 into the g_max3 field
   Put the text string "min1 - max1" into the variable string
   Put the value in string into the macrovariable g_group1
   Put the text string "min2 - max2" into the variable string
   Put the value in string into the macrovariable g_group2
   Put the text string "min3 - max3" into the variable string
   Put the value in string into the macrovariable g_group3

3.4.5.8 Notes

This is an optional window.

3.4.5.9 Program screens

Figure 3.4.12 represents the screen the user will see. Figure 3.4.13 contains the field names in parentheses on the program screen.
Fig. 3.4.12. The BMANGE.PROGRAM screen.

Tab to the desired field.

Fig. 3.4.13. The BMANGE.PROGRAM screen with field names.

\[\begin{align*}
\text{Min 1: } & \quad \text{Max 1: } \\
\text{Min 2: } & \quad \text{Max 2: } \\
\text{Min 3: } & \quad \text{Max 3: }
\end{align*}\]

Tab to the desired field.

3.4.6 BOPTION.PROGRAM

3.4.6.1 Program purpose

BOPTION.PROGRAM allows the user to select a calculation type. From this screen the user may choose to calculate the total, the average, or the percent for this report.
3.4.6.2 Calling program

BAR.PROGRAM

3.4.6.3 Programs called

N/A

3.4.6.4 Program returns to

BAR.PROGRAM

3.4.6.5 SAS window functions called by program

N/A

3.4.6.6 Macrovariables

Used - N/A

Initialized - N/A

Set - g_calopt contains the string that indicates the calculation option the user has chosen (i.e., 'TOTAL', 'AVERAGE', or 'PERCENT')

3.4.6.7 Structured English for program code

When user presses <return> while in the total field
   If total field does not contain a '*' or a blank then do
      If avg and percent fields contain blanks then do
         Put a '*' in the total field
         Put word 'TOTAL' in the variable string
         Put the value in string into the macrovariable g_calopt
         Position cursor in total field
         Exit to BAR.PROGRAM
      Else sound alarm and display message 'Only one option allowed.'
      Else position cursor in avg field
   Else position cursor in total field
When user presses <return> while in the avg field
If avg field does not contain a '*' or a blank then do
    If total and percent fields contain blanks then do
        Put a '*' in the avg field
        Put word 'AVERAGE' in the variable string
        Put the value in string in the macrovariable g_calopt
        Position cursor in avg field
        Exit to BAR.PROGRAM
    Else sound alarm and display message 'Only one option allowed.'
        Put blank in avg field
        Position cursor in avg field
    Else position cursor in percent field
When user presses <return> while in the percent field
    If percent field does not contain a '*' or a blank then do
        If total and avg fields contain blanks then do
            Put a '*' in the percent field
            Put word 'PERCENT' in the variable string
            Put the value in string in the macrovariable g_calopt
            Position cursor in the percent field
            Exit to BAR.PROGRAM
        Else sound alarm and display message 'Only one option allowed.'
            Put blank in percent field
            Position cursor in percent field
    Else position cursor on command line

3.4.6.8 Notes

    This is a required window.
    Only one calculation option can be chosen in one session.

3.4.6.9 Program screens

    Figure 3.4.14 represents the screen the user will see. Figure 3.4.15 contains the field names in parentheses for the program screen.
Calculation Options

_____ Total
_____ Average
_____ Percent

Enter 'X' beside your choice.
Press RETURN.

Press F10 to exit this screen.

Fig. 3.4.14. The BOPTION.PROGRAM screen.

Calculation Options

(total) Total
(avg) Average
(percent) Percent

Enter 'X' beside your choice.
Press RETURN.

Press F10 to exit this screen.

Fig. 3.4.15. The BOPTION.PROGRAM screen with field names.
3.4.7 BOUTOPT.PROGRAM

3.4.7.1 Program purpose

BOUTOPT.PROGRAM allows the user to choose the destination for the output of a bar chart. The user may choose to send the output to a printer, to an operating system file, to a Freelance file, or to the screen.

3.4.7.2 Calling Program

BAR.PROGRAM

3.4.7.3 Program called

BREADY.PROGRAM

3.4.7.4 Program returns to

BAR.PROGRAM

3.4.7.5 SAS window functions called by program

N/A

3.4.7.6 Macrovariables

Used - N/A

Initialized - N/A

Set - g_outopt contains a string holding the destination option ('screen' or 'print') the user has chosen

3.4.7.7 Structured English for program code

Put a blank in the print field, the osfile field, the ffie field, and the screen field

When user presses <return> while in the print field

If print field does not contain a '*' or a blank then do

Put a '*' in the print field

Put the word 'print' into the variable string
Put the value in string variable into the macrovariable \texttt{g\_outopt}

Call \texttt{BREADY\_PROGRAM}

Put a blank in the print field

Position cursor on the print field

Else position cursor on the next field

When user presses \texttt{<return>} while in the screen field

If screen field does not contain a '⋆' or a blank then do

- Put a '⋆' in the screen field
- Put word 'screen' into the variable string
- Put the value in string variable into the macrovariable \texttt{g\_outopt}
- Call \texttt{BREADY\_PROGRAM}
- Put a blank in the print field
- Position cursor on the print field

Else position cursor on the next field (see notes in this program)

When user presses \texttt{<return>} while in the osfile field

If the osfile field does not contain a '⋆' or a blank then do

- Put a '⋆' in the osfile field
- Display message 'This option is not available in this version.'
- Put a blank in the osfile field
- Position cursor on the osfile field

Else position cursor on the next field

When user presses \texttt{<return>} while in the flfile field

If the flfile field does not contain an '⋆' or a blank then do

- Put a '⋆' in the flfile field
- Display message 'This option is not available in this version.'
- Put a blank in the flfile field
- Position cursor on the flfile field

Else position cursor on the next field

3.4.7.8 Notes

The options for sending output to an operating system file, or a Freelance file are not functional in the prototype. If the user chooses either of these options, he/she will see a message that explains that the function is not available in this version.

The 'next field' for the last field, the screen field, is the command line.

This is a required window.

3.4.7.9 Program screens

Figure 3.4.16 represents the screen the user will see. Figure 3.4.17 contains the field names in parentheses for the program screen.
Output Options

___  Print Output

___  Save Output to an Operating System File

___  Save Output to a FreeLance File

___  Send Output to the Screen

Enter 'X' beside your choice.
Press RETURN.

Press F10 to exit this screen.

Fig. 3.4.16. The BOUTOPT.PROGRAM screen.

Output Options

(print)  Print Output

(osfile)  Save Output to an Operating System File

(llfile)  Save Output to a FreeLance File

(screen)  Send Output to the Screen

Enter 'X' beside your choice.
Press RETURN.

Press F10 to exit this screen.

Fig. 3.4.17. The BOUTOPT.PROGRAM screen with field names.
3.4.8 BREADY.PROGRAM

3.4.8.1 Program purpose

BREADY.PROGRAM creates the bar chart and sends the output to the destination chosen by the user in BOUTOPT.PROGRAM.

3.4.8.2 Calling program

BOUTOPT.PROGRAM

3.4.8.3 Programs called

N/A

3.4.8.4 Program returns to

BOUTOPT.PROGRAM

3.4.8.5 SAS window functions called by program

N/A

3.4.8.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>g_bartyp</th>
<th>contains a string holding the bar type option (i.e., 'HBAR' or 'VBAR') chosen by the user in MENU.PROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>g_calopt</td>
<td>contains a string holding the calculation option chosen by the user in BOPTION.PROGRAM (i.e., 'TOTAL', 'AVERAGE', or 'PERCENT')</td>
</tr>
<tr>
<td></td>
<td>g_calvar</td>
<td>contains a string holding the numeric variable chosen in BOPTION.PROGRAM</td>
</tr>
<tr>
<td></td>
<td>g_file</td>
<td>see macrovariables set in this program</td>
</tr>
<tr>
<td></td>
<td>g_nc</td>
<td>see macrovariables set in this program</td>
</tr>
<tr>
<td></td>
<td>g_outopt</td>
<td>contains a string holding the user's output destination choice (i.e., 'screen' or 'print')</td>
</tr>
</tbody>
</table>
### 3.4.8.7 Structured English for program code

Create variable optstr as the value in the macrovariable g_calopt
Create variable outstr as the value in the macrovariable g_outopt
Create variable xvar as the value in the macrovariable g_xvar
If the value in xvar is 'pumon' or 'pyr' or 'pumonyr' then
    Set macrovariable g_nc to contain the string 'discrete'
Else
    Set macrovariable g_nc to contain a blank
If the value in outstr is 'screen' then
    Display message 'Press F10 to return to this screen then press F10 to exit.'
Else if the value in outstr is 'print' then
    Display message 'Press F10 to exit this screen.'
If the value in optstr is 'TOTAL' then do
   Put the string 'type = sum sumvar = &g_calvar'
      into the macrovariable g_str1
   Put the string 'whist.data2' into the macrovariable g_file
If the value in optstr is 'AVERAGE' then do
   Put the string 'type = mean sumvar = &g_calvar' into the macrovariable g_str1
   Put the string 'whist.data2' into the macrovariable g_file
If the value in optstr is 'PERCENT' then do
   Create a temporary dataset temp1 using data2
      sorted by the value in g_xvar
   Calculate the means using the temporary dataset temp1 and the values in g_xvar and g_calvar
   Create a temporary dataset new that contains the column ssum from temp1 that was called sum
   Go through a loop to add all the values in the variable named in the macrovariable g_calvar
      Do not create a new dataset but use dataset data2
      Create a variable end as the SAS end-of-file indicator
      If this is the first observation in the dataset data2a then do
         Create a variable tot as the first value in
            the column named in the macrovariable g_calvar
      Else do
         Set tot as the value in the variable tot plus the next value in the column named in the macrovariable g_calvar
         Do not initialize the variable tot to zero
      If this is the last observation in the dataset data2 then do
         Create macrovariable g_tot as the final value in the variable tot
   Create a new dataset data21 using the dataset new
   Sort by the column names in the macrovariable g_xvar
   Create a variable percent as the value in ssum divided by the value in g_tot, times 100
   Put the string 'whist.data21' into the macrovariable g_file
   Put the string 'sumvar = percent' into the macrovariable g_str1
If the value in outstr is 'screen' then
   Put the string 'device = sun' into the macrovariable g_str2
If the value in outstr is 'print' then
   Put the string 'gaccess = "sasgacmd>lpr" device = hpljs2'
      into the macrovariable g_str2
Set options
Generate a bar chart using the dataset named in the macrovariable \texttt{g\_file}, the type of bar chart named in the macrovariable \texttt{g\_bartyp}, the bar variable in the macrovariable \texttt{g\_xvar}, and the options named in the macrovariables \texttt{g\_str1} and \texttt{g\_nc}.

3.4.8.8 Notes

\textsc{BREADY.PROGRAM} will display a message as the user waits for output to the screen or printer. The device names for printers or screens that are contained in the macrovariable \texttt{g\_str2} can easily be changed to accommodate different hardware. It is also easy to change the colors set in the \texttt{goptions} statement and the colors and the patterns set in the \texttt{gchart} statement.

3.4.8.9 Program screens

There are no enterable fields in this program screen. A full-sized screen will appear with a message or instructions depending on whether the user chose 'screen' or 'print' in \textsc{BOUTOPT.PROGRAM}. Figure 3.4.18 represents the screen the user will see.

![Processing...One moment please.](image)

Fig. 3.4.18. The \textsc{BREADY.PROGRAM} screen.

3.5 \textsc{LINE CHART PROGRAMS}

The line chart programs are called when the user specifies 'Line Chart' on the \textsc{MENU.PROGRAM} screen. The first screen the user will see is \textsc{LINE.PROGRAM}. This screen allows the user to call one of the following screens:

\begin{itemize}
  \item \textsc{LINEDATA.PROGRAM},
  \item \textsc{LOPTION.PROGRAM},
  \item The SAS titles function window,
  \item \textsc{LOUTOPT.PROGRAM}.
\end{itemize}
Figure 3.5.1 is a hierarchy chart of the programs that are components of the line program set. LINEDATA.PROGRAM allows the user to select a Y-axis variable and a line variable from the original dataset for inclusion in the report. It is a mandatory screen. LINEDATA.PROGRAM may call BRANGE.PROGRAM if the user chooses to specify a range of values for the Y-axis variable. BRANGE.PROGRAM allows the user to specify one set of minimum and maximum values for a numeric variable.

LOPTION.PROGRAM allows the user to choose a calculation type for the chart. It is a mandatory screen. The choices available are 'AVERAGE,' 'TOTAL,' and 'PERCENT.' The choice the user makes is stored in a macrovariable and used by LREADY.PROGRAM.

The SAS title function window is optional. The user may specify up to ten lines for a title to be included in the chart.

LOUTOPT.PROGRAM allows the user to direct the output. The output may be directed to a printer or to the screen. LOUTOPT.PROGRAM calls LREADY.PROGRAM.

LREADY.PROGRAM creates the line chart and sends the output to the destination specified by the user. It also does calculations and creates a new dataset file.

Each of these programs may use macrovariables that have been initialized or set from another program or the programs may themselves initialize or set macrovariables that may be used by subsequent programs. Figure 3.5.2 shows the macrovariables that are initialized, set, or used by each of the programs that make up the line program set. Figure 3.5.3 shows the dataset file(s) that is used as input and the dataset file(s) that is written when the program is executed. In some cases no dataset file is read and/or no new dataset file is created during execution of a program.

3.5.1 LINE.PROGRAM

3.5.1.1 Program purpose

LINE.PROGRAM allows the user to select options to generate a line chart. From this screen the user chooses to call other screens for the purpose of selecting data columns for a report, selecting a calculation type, specifying a title, and selecting the output destination.

3.5.1.2 Calling program

MENU.PROGRAM

3.5.1.3 Programs called

LINEDATA.PROGRAM
LOPTION.PROGRAM
LOUTOPT.PROGRAM

3.5.1.4 Program returns to

MENU.PROGRAM
Fig. 3.5.1. Hierarchy of programs in the line program set.
Fig. 3.5.2. Hierarchy of line programs showing macrovariables set, used, or initialized in the programs.
Fig. 3.5.3. Hierarchy of line programs showing permanent SAS datasets read or created.
3.5.1.5 SAS window functions called by program

SAS Titles Window

3.5.1.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialized</td>
<td>N/A</td>
</tr>
<tr>
<td>Set</td>
<td>N/A</td>
</tr>
</tbody>
</table>

3.5.1.7 Structured English for program code

When user presses <return> while in first field-var1
  If there is not a '*' or a blank in the var1 field
  then do
  Put a '*' in the var1 field
  Call LINEDATA.PROGRAM
  Else position cursor in var2 field
When the user presses <return> while in second field-var2
  If there is not a '*' or a blank in the var2 field
  then do
  Put a '*' in the var2 field
  Call LOPTION.PROGRAM
  Else position cursor in var3 field
When the user presses <return> while in third field-var3
  If there is not a '*' or a blank in the var3 field
  then do
  Put a '*' in the var3 field
  Display message 'Press RETURN then fill in titles. Press F10 when finished.'
  Call the SAS Titles Window
  Else position cursor in the var4 field
When the user presses <return> while in the fourth field-var4
  If there is not an '*' or a blank in the var4 field
  then do
  Put a '*' in the var4 field
  Call LOUTOPT.PROGRAM
  Else position cursor on the command line
3.5.1.8 Notes

The X-axis variable is always pickup month.
This window is required when producing a line chart.

3.5.1.9 Program screens

Figure 3.5.4 represents the screen the user will see. Figure 3.5.5 contains the field names in parentheses on the program screen.

```
Output Selection

_____ Select Variables for Analysis
_____ Select Calculation Option
_____ Specify Titles
_____ Select Output Option

Enter 'X' beside your choice.
Press RETURN.
Press F10 to exit this screen.
```

Fig. 3.5.4. The LINE.PROGRAM screen.

```
Output Selection

(var1) Select Variables for Analysis
(var2) Select Calculation Option
(var3) Specify Titles
(var4) Select Output Option

Enter 'X' beside your choice.
Press RETURN.
Press F10 to exit this screen.
```

Fig. 3.5.5. The LINE.PROGRAM screen with field names.
3.5.2 LINEDATA.PROGRAM

3.5.2.1 Program purpose

LINEDATA.PROGRAM allows the user to select a Y-axis variable and a line variable through a SAS VARLIST Window. This program creates a new dataset data2 using datasets data1 and data1a. If the user chooses to specify a range of values for the Y-axis variables this program calls BRANGE.PROGRAM.

3.5.2.2 Calling program

LINE.PROGRAM

3.5.2.3 Program called

May call BRANGE.PROGRAM

3.5.2.4 Program returns to

LINE.PROGRAM

3.5.2.5 SAS window functions called by program

VARLIST Window
Legend Window

3.5.2.6 Macrovariables

Used -
  g_calvar  see macrovariables set in this program
  g_high   see macrovariables set in this program
  g_low    see macrovariables set in this program
  g_yvar   see macrovariables set in this program

Initialized -
  g_high   initialized to '99999'
  g_low    initialized to '0'
  g_sign   initialized to blank
Set \( \text{g\_calvar} \) contains the string 'puyr' if pickup date was chosen as the line variable
\( \text{g\_high} \) contains the maximum value the user chose in \text{BRANGE.PROGRAM} or '99999'
\( \text{g\_low} \) contains the minimum value the user chose in \text{BRANGE.PROGRAM} or '0'
\( \text{g\_sign} \) contains '=' if the user chose pickup date as the line variable or is blank
\( \text{g\_yvar} \) contains the numeric column name the user chose for the Y-axis variable

3.5.2.7 Structured English for program code

Declare a ten character length for the variables str1 and str2
Initialize the macrovariable \( \text{g\_low} \) to '0'
Initialize the macrovariable \( \text{g\_high} \) to '99999'
Initialize the macrovariable \( \text{g\_sign} \) to blank
Define the format for dates
Create a temporary dataset temp as dataset data1a
with a column pumon which is the month position of the data1a column called pudate
When the user presses <return> while in the first field-var1
If the var1 field does not contain a '*' or a blank then do
  Put a '*' in the var1 field
  Call the leg1 module
  If the dataset data1 does not exist then do
    Display message 'Cannot open dataset.'
  Else do
    Call the VARLIST window and allow user to choose one numeric data column from dataset data1 and put the user's choice in the variable str1
    Put the value in the variable str1 into the macrovariable \( \text{g\_yvar} \)
    Position the cursor in the var2 field
Else position the cursor in the var2 field
When the user presses <return> while in the second field-var2
If the var2 field does not contain a '*' or a blank then do
  Put a '*' in the var2 field
  Call the leg1 module
  If the dataset data1 does not exist
    Display message 'Cannot open dataset.'
Else
Call the VARLIST window and allow user to
choose one character variable column
from dataset data1 and put the user's
choice in the variable str2
If str2 is 'PUPDATE' then do
Put 'puyr' in the variable str2
Rewrite the dataset temp to contain
a column 'puyr' which is the
year portion of the temp
dataset column called pudate
If the variable str2 is not blank
Put '=' in the macrovariable g_sign
Put the value in the variable str2 in the macrovariable
g_calvar
Position the cursor in the var3 field
Else position the cursor in the var3 field
When the user presses <return> while in the third field-var3
If the var3 field does not contain a '*' or a blank then do
Put a '*' in the var3 field
Call the leg2 module
Call BRANGE.PROGRAM
Else position the cursor on the command line
Create a new dataset data2 using the dataset temp
Sort by the variable named in the macrovariable g_calvar and pumon
Keep only the values in the column named in the macrovariable g_yvar
that have values greater than or equal to the value
in the macrovariable g_low and less than or equal to
the value in the macrovariable g_high
Leg1 module:
   Specifies text and colors of a SAS Legend Window
Leg2 module:
   Specifies text and colors of a SAS Legend Window

3.5.2.8 Notes

This is a required window.

3.5.2.9 Program screens

Figure 3.5.6 represents the screen the user will see. Figure 3.5.7 contains the field names
in parentheses on the program screen.
Data Selection

- Select Y-axis variable
- Select Line variable
- Select Maximum/Minimum

Fig. 3.5.6. The LINEDATA.PROGRAM screen.

Data Selection

(var1) Select Y-axis variable
(var2) Select Line variable
(var3) Select Maximum/Minimum

Fig. 3.5.7. The LINEDATA.PROGRAM screen with field names.

3.5.3 BRANGE.PROGRAM

3.5.3.1 Program purpose

BRANGE.PROGRAM allows the user to specify one minimum and maximum boundary for data values for the calculation variable of a bar chart. BRANGE.PROGRAM is called if the user chooses 'weight' as the calculation variable.

3.5.3.2 Calling program

LINEDATA.PROGRAM

3.5.3.3 Programs called

N/A
3.5.3.4 Program returns to

LINETYPE.PROGRAM

3.5.3.5 SAS window functions called by program

N/A

3.5.3.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>-</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialized</td>
<td>-</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Set - g_high
contains the maximum value

g_low contains the minimum value

3.5.3.7 Structured English for program code

When this program is called
  Put a '0' in the lower field
  Put '9999' in the upper field
When the user exists this program
  Put the value in the lower field into the
    g_low macrovariable
  Put the value in the upper field into the
    g_high macrovariable

3.5.3.8 Notes

This program is also used by the BARDATA.PROGRAM and the PIEDATA.PROGRAM.

3.5.3.9 Program screens

Figure 3.5.8 represents the screen the user will see. Figure 3.5.9 contains the field names in parentheses on the program screen.
3.5.4 LOPTION.PROGRAM

3.5.4.1 Program purpose

LOPTION.PROGRAM allows the user to select a calculation type. From this screen the user may choose to calculate the total, the average, or the percent for a line chart.

3.5.4.2 Calling program

LINE.PROGRAM
3.5.4.3 Programs called

N/A

3.5.4.4 Program returns to

LINE.PROGRAM

3.5.4.5 SAS window functions called by program

N/A

3.5.4.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>-</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialized</td>
<td>-</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Set - g_calopt contains calculation option user chooses (i.e., 'TOTAL,' 'AVERAGE,' or 'PERCENT')

3.5.4.7 Structured English for program code

When user presses <return> while in the total field
If total field does not contains a '*'' or a blank then do
    If avg and percent fields contain blanks then do
        Put a '*' in total field
        Put the string 'TOTAL' in the macrovariable g_calopt
        Exit to LINE.PROGRAM
    Else sound alarm and display message 'Only one option allowed.'
        Put blank in total field
        Position cursor in total field
    Else position cursor in avg field

When user presses <return> while in the avg field
If avg field does not contain a '*' or a blank then do
    If total and percent fields contain blanks then do
        Put a '*' in the avg field
        Put the string 'AVERAGE' in the macrovariable g_calopt
        Exit to LINE.PROGRAM
    Else sound alarm and display message 'Only one option allowed.'
        Put blank in avg field
        Position cursor in avg field
    Else position cursor in percent field
When user presses <return> while in the percent field
   If percent field does not contain a '*' or a blank then do
      If total and avg fields contain blanks then do
         Put a '*' in the percent field
         Put the string 'PERCENT' in the macrovariable g_calopt
         Exit to LINE.PROGRAM
      Else sound alarm and display message 'Only one option allowed.'
         Put blank in percent field
         Position cursor in percent field
   Else put cursor on command line

3.5.4.8 Notes

   This is a required window.

3.5.4.9 Program screens

   Figure 3.5.10 represents the screen the user will see. Figure 3.5.11 contains the field names in parentheses on the program screen.

   ![
   ![Calculation Options]
   
   Calculation Options

   — Total
   — Average
   — Percent

   Fig. 3.5.10. The LOPTION.PROGRAM screen.
LOUTOPT.PROGRAM

3.5.5.1 Program purpose

LOUTOPT.PROGRAM allows the user to choose the destination for the output of the line report. The user may choose to send the output to a printer, to a file, or to the monitor screen.

3.5.5.2 Calling program

LINE.PROGRAM

3.5.5.3 Program called

LREADY.PROGRAM

3.5.5.4 Program returns to

LINE.PROGRAM

3.5.5.5 SAS window functions called by program

N/A
3.5.5.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialized</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Set g_outopt contains a string holding the output destination option the user has chosen (i.e., 'screen' or 'print')

3.5.5.7 Structured English for program code

Put a blank in the print field, the osfile field, the flfile field, and the screen field.

When user presses <return> while in the print field

If print field does not contain a '*' or a blank then do

- Put a '*' in the print field
- Put the string 'print' into the macrovariable g_outopt
- Call LREADY.PROGRAM
- Put a blank in the print field
- Position cursor on the print field

Else

- Position cursor on the next field

When the user presses <return> while in the screen field then do

If screen field does not contain a '*' or a blank then do

- Put a '*' in the screen field
- Put the word 'screen' into the macrovariable g_outopt
- Call LREADY.PROGRAM
- Put a blank in the screen field
- Position the cursor on the screen field

Else

- Position cursor on the next field (see notes on this program)

When user presses <return> while in the osfile field

If osfile field does not contain a '*' or a blank then do

- Put a '*' in the osfile field
- Display message 'This option is not available in this version.'
- Put a blank in the osfile field
- Position cursor on the osfile field

Else

- Position cursor on the next field

When the user presses <return> while in the flfile field

If the flfile field does not contain a '*' or a blank then do

- Put a '*' in the flfile field
- Display message 'This option is not available in this version.'
Put a blank in the file field
Position cursor on the file field
Else
Position cursor on the next field

3.5.5.8 Notes

The options for sending output to an operating system file or a Freelance file are not functional in the prototype. If the user chooses either of these options, he/she will see a message that explains that the function is not available in this version.

The 'next field' for the last field in the program, the screen field, is the command line. This a required window.

3.5.5.9 Program screens

Figure 3.5.12 represents the screen the user will see. Figure 3.5.13 contains the field names in parentheses on the program screen.

Output Options

- Print Output
- Save Output to an operating system file
- Save Output to a Freelance file
- Send Output to the Screen

Enter 'X' beside your choice.
Press RETURN.

Press F10 to exit this screen.

Fig. 3.5.12. The LOUTOPT.PROGRAM screen.
Output Options

(print) Print Output
(osfile) Save Output to an operating system file
(ffile) Save Output to a FreeLance file
(screen) Send Output to the screen

Enter 'X' beside your choice.
Press RETURN.

Press F10 to exit this screen.

Fig. 3.5.13. The LOUTOPT.PROGRAM screen with field names.

3.5.6 LREADY.PROGRAM

3.5.6.1 Program purpose

LREADY.PROGRAM sends the output to the destination chosen by the user in LOUTOPT.PROGRAM. This program creates a new dataset data3 using dataset data2.

3.5.6.2 Calling program

LOUTOPT.PROGRAM

3.5.6.3 Program called

N/A

3.5.6.4 Program returns to

LOUTOPT.PROGRAM

3.5.6.5 SAS window functions called by program

N/A
3.5.6.6 Macrovariables

**Used**

- **g_calopt** contains a string holding the calculation option chosen by the user in LOPTION.PROGRAM (i.e., 'TOTAL', 'AVERAGE', or 'PERCENT')
- **g_calvar** contains the string 'puyr' if pickup date was chosen as the line variable in LINEDATA.PROGRAM
- **g_outopt** contains a string holding the user's output destination choice (i.e., 'screen' or 'print')
- **g_sign** contains a blank or if the user chose pickup date as the line variable, it contains '='
- **g_str2** see macrovariables set in this program
- **g_tot** see macrovariables set in this program
- **g_yvar** see macrovariables set in this program
- **str3** see macrovariables set in this program

**Initialized**

- **N/A**

**Set**

- **g_str2** contains a string holding one of two commands for output devices depending on whether the user chose 'print' or 'screen' in LOUTOPT.PROGRAM (i.e., 'device = sun cback=black ctext=yellow' or 'gaccess=''sasgacmd>lpr'' device=hpljs2 cback=white ctext=black'

- **g_tot** (this is only set if the user chose 'PERCENT' in LOPTION.PROGRAM) contains the total when all values for the variable g_calvar have been added together

- **g_yvar** contains the name of the numeric column chosen as the Y-axis variable in LINEDATA.PROGRAM; this macrovariable is overwritten in this program to contain a string holding one of three values for the calculation depending on the value in the macrovariable g_calopt, chosen by the user in LOPTION.PROGRAM (i.e., if g_calopt = 'TOTAL' then g_yvar = 'ssum', if g_calopt = 'PERCENT' then g_yvar = 'percent')
str3 contains a string holding one of two values depending on the value in the macrovariable g_calopt (i.e., if 'screen' then str3 contains the string 'white', if 'print' then str3 contains the string 'black')

3.5.6.7 Structured English for program code

Create variable optstr as the value in the macrovariable g_calopt
Create variable outstr as the value in the macrovariable g_outopt
Create a dataset data3 using the data in dataset data2 sorted by
the column in the macrovariable g_calvar and by the column
pumon using the variable named in the macrovariable
g_yvar to perform the means calculation; rename
the output columns mean to smean and sum to ssun
If the value in the variable outstr is 'screen' then
Display message 'Press F10 to return to this screen
then press F10 to exit.'
Else if the value in the variable outstr is 'print' then
Display message 'Press F10 to exit this screen.'
If the value in the variable outstr is 'AVERAGE' then do
Put the string 'smean' into the macrovariable g_yvar
If the value in the variable optstr is 'PERCENT' then do
Do not create a new dataset but use dataset data2
Create a variable end as the SAS end-of-file indicator
If this is the first observation in the dataset data2 then do
Create a variable tot as the first value in the column
named in the macrovariable g_yvar
Else do
Set tot as the value in the variable tot plus the
next value in the column named in the macrovariable
g_yvar
Do not initialize the variable tot to zero
If this is the last observation in the dataset data2
then do
Create macrovariable g_tot as the final value in
the variable tot
Create a new dataset data3 using the dataset data2
Sort by the column name in the macrovariable g_calvar
and the column pumon
Create a variable percent as the value in ssun divided
by the value in g_tot, times 100
Put the string 'percent' in the macrovariable g_yvar
If the value in the variable outstr is 'screen' then do
Put the string 'device=sun cback=black ctext=yellow'
into the macrovariable g_str2
Put the string 'white' into the macrovariable str3
If the value in the variable outstr is 'print' then do
   Put the string 'access=sasacmd>lpr device=hpljs2'
       into the macrovariable g_str2
   Put the string 'black' into the macrovariable str3
Set options
   Generate a line chart using dataset data3, use the word 'Month'
       as the label for the data from the pumon
       column, format pumon as a three letter abbreviation,
       use the column named in the macrovariable
       g_yvar plotted against the pumon column, include
       the equals sign if the user has chosen a line variable,
       followed by that line variable choice
       column name, set the axis colors, include a
       frame, and set 12 tick marks on the X-axis

3.5.6.8 Notes

LREADY.PROGRAM will display a message as the user waits for output to the screen or
printer. The device names for the printers or screens are contained in the macrovariable g_str2
and can easily be changed to accommodate different hardware. It is also easy to change the
colors and other options set in the gplot and goptions statements.

3.5.6.9 Program screen

There are no enterable fields in this program screen. A full-sized screen will appear with a
message and instructions depending on whether the user chose 'screen' or 'print' in
LOUTOPT.PROGRAM. Figure 3.5.14 represents the screen the user will see.

![Processing — One moment please.](image)

Fig. 3.5.14. The LREADY.PROGRAM screen.
3.6 PIE CHART PROGRAMS

The pie chart programs are called when the user specifies 'Pie Chart' on the MENU.PROGRAM screen. The first screen the user will see is PIE.PROGRAM. This screen allows the user to call one of the following screens:

PIEDATA.PROGRAM,
The SAS titles function window, or
POUTOPT.PROGRAM.

Figure 3.6.1 is a hierarchy chart of the programs that are components of the pie program set. PIEDATA.PROGRAM allows the user to select a pie slice variable and a calculation variable from the original dataset for inclusion in the report. It is a mandatory screen. PIEDATA.PROGRAM will call BRANGE.PROGRAM if the user chooses 'WT' as the calculation variable. BRANGE.PROGRAM allows the user to specify one set of minimum and maximum values for the 'WT' variable. PIEDATA.PROGRAM will call TIME.PROGRAM if the user chooses 'PUPDATE' as the pie slice variable. TIME.PROGRAM allows the user to specify the way he/she wants to look at time (i.e., by year; by month and year; or by day, month, and year).

The SAS titles function window is optional. The user may specify up to ten lines for a title to be included in the chart.

POUTOPT.PROGRAM allows the user to direct the output. The output may be directed to a printer or the screen. POUTOPT.PROGRAM calls PREADY.PROGRAM. PREADY.PROGRAM creates the pie chart and sends the output to the destination specified by the user.

Each of these programs may use macrovariables that have been initialized or set from another program or the programs themselves may initialize or set macrovariables that may be used by subsequent programs. Figure 3.6.2 shows the macrovariables that are initialized, set, or used by each of the programs in the pie program set. Figure 3.6.3 shows the dataset file(s) that is used as input and the dataset file(s) that is written when the program is executed. In some cases, no dataset file is read and/or no new dataset is created during execution of a program.

3.6.1 PIE.PROGRAM

3.6.1.1 Program purpose

PIE.PROGRAM allows the user to select options for producing a pie chart. From this screen the user chooses to call other screens for the purpose of selecting data columns for the chart, specifying a title, and selecting the output destination.

3.6.1.2 Calling program

MENU.PROGRAM
Fig. 3.6.1. Hierarchy of programs in the pie program set.
Fig. 3.6.2. Hierarchy of pie programs showing macrovariables set, used, or initialized in the programs.
Fig. 3.6.3. Hierarchy of pie programs showing permanent SAS datasets read or created.
3.6.1.3 Programs called

PIEDATA.PROGRAM
POUTOPT.PROGRAM

3.6.1.4 Program returns to

MENU.PROGRAM

3.6.1.5 SAS window functions called by program

SAS Titles Window

3.6.1.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>-</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialized</td>
<td>-</td>
<td>N/A</td>
</tr>
<tr>
<td>Set</td>
<td>-</td>
<td>N/A</td>
</tr>
</tbody>
</table>

3.6.1.7 Structured English for program code

When user presses <return> while in the first field - var1
  If there is not a '*' or a blank in the var1 field
    then do
      Put a '*' in the var1 field
      Call PIEDATA.PROGRAM
    Else position cursor in the var2 field
 When the user presses <return> while in the second field - var2
  If there is not a '*' or a blank in the var2 field
    then do
      Put a '*' in the var2 field
      Display message 'Press RETURN then fill in titles.
      Press F10 when finished.'
      Call the SAS titles window
    Else position cursor in the var3 field
 When the user presses <return> while in the third field - var3
  If there is not a '*' or a blank in the var3 field
    then do
      Put a '*' in the var3 field
      Call POUTOPT.PROGRAM
    Else position cursor on the command line
3.6.1.8 Notes

This is a required window.
Percent is hardcoded as the only calculation option.

3.6.1.9 Program screens

Figure 3.6.4 represents the screen the user will see. Figure 3.6.5 contains the field names in parentheses on the program screen.

```
Output Selection

--- Select Variables for Analysis
--- Specify Titles
--- Select Output Option

Enter 'X' beside your choice.
Press RETURN.
Press F10 to exit this screen.
```

Fig. 3.6.4. The PIE.PROGRAM screen.
3.6.2 PIEDATA.PROGRAM

3.6.2.1 Program purpose

PIEDATA.PROGRAM allows the user to select the column to be used for the pie slice variable and the calculation variable to produce a pie chart. This program will call TIME.PROGRAM if the user has chosen the column 'PUDATE' as the pie variable. It will call BRANGE.PROGRAM if the user has chosen the column 'WT' as the calculation variable. This program uses dataset data1 for all functions unless the user chose 'PUDATE' as the pie variable, then the program uses data1a. This program writes the dataset data2 in all cases.

3.6.2.2 Calling program

PIE.PROGRAM

3.6.2.3 Programs called

may call BRANGE.PROGRAM
may call TIME.PROGRAM

3.6.2.4 Program returns to

PIE.PROGRAM
3.6.2.5 SAS window functions called by program

Legend Window
VARLIST Window

3.6.2.6 Macrovariables

Used -

- \texttt{g\_calvar}\hspace{1cm} see macrovariables set in this program

- \texttt{g\_date}\hspace{1cm} contains the date type the user chose in \texttt{TIME.PROGRAM}

- \texttt{g\_high}\hspace{1cm} contains the maximum value chosen in \texttt{BRANGE.PROGRAM}

- \texttt{g\_low}\hspace{1cm} contains the minimum value chosen in \texttt{BRANGE.PROGRAM}

- \texttt{g\_xvar}\hspace{1cm} see macrovariables set in this program

Initialized -

- \texttt{g\_high}\hspace{1cm} initialized to '99999'

- \texttt{g\_low}\hspace{1cm} initialized to '0'

Set -

- \texttt{g\_calvar}\hspace{1cm} contains a string holding the name of the calculation variable the user chooses

- \texttt{g\_str1}\hspace{1cm} contains a string holding the name for the pie variable the user chooses

- \texttt{g\_xvar}\hspace{1cm} contains a string holding the name of the pie variable the user chooses

3.6.2.7 Structured English for program code

Set the length for variables \texttt{str1} and \texttt{str2}
Initialize the macrovariables \texttt{g\_low} to '0' and \texttt{g\_high} to '99999'
Define the format for dates
When the user presses <\texttt{return}> while in the first field - \texttt{var1}
   If \texttt{var1} field does not contain a '*' or a blank then do
      Call leg1 module
      If dataset \texttt{data1} does not exist then do
         Display message 'Dataset not found. Press F10.'
Else do
   Call the VARLIST window and allow the user to choose
   one character column from dataset datal and put
   the user's choice in the variable str1
   Put the value in the str1 variable into the
   macrovariable g_str1
If the value in str1 is 'PUDATE' then do
   Call the handle_date module
Else do
   Create a temporary dataset temp using dataset datal
   Put the value in the str1 variable into the macrovariable g_xvar
If the value in str1 is the same as
   the value in str2 then do
   Sound alarm
   Display message 'Pie variable = calculation variable'
   Position the cursor in the var1 field
Else position cursor in the var2 field
When the user presses <return> while in the second field - var2
   If the var2 field does not contain a '*' or a blank then do
   Call the leg1 module
   If dataset datal does not exist then do
      Display message 'Data not found. Press F10.'
   Else do
      Call the VARLIST window and allow the user to choose
      one numeric column from dataset datal and put the user's choice in the
      variable str2
   If the value in str2 is 'WT' then do
      Call BRANGE.PROGRAM
   If the value in str2 is the same as the value in str1 then do
      Sound alarm
      Display message 'Pie variable = calculation variable'
      Position the cursor in the var2 field
Else do
   Put the value in the variable str2 into the
   macrovariable g_calvar
Else do
   Position cursor on the command line
Create a dataset data2 using the dataset temp
Sort by the column name in the macrovariable g_xvar
Use only the values in the column named in the macrovariable g_calvar
   that are greater than or equal to the value in the
   macrovariable g_low and less than or equal to the
   value in the macrovariable g_high
leg1 module:
   Specifies the color and text for the legend window
handle-date module:

Call TIME.PROGRAM
Put the value in the macrovariable g_date (from TIME.PROGRAM)
into the variable str1
If the value in str1 is 'pumon' then do
Create a temporary dataset temp using dataset data1a
Use the format for dates defined at the beginning of
PIEDATA.PROGRAM
Create a variable 'pumon' as the month value in the column
'PUDATE' from data1a
If the value in str1 is 'puyr' then do
Create a temporary dataset temp using dataset data1a
Create a variable 'puyr' as the year value in the column
'PUDATE' from data1a
If the value in str1 is 'pumonyr' then do
Create a temporary dataset temp using dataset data1a
Use the format for dates monyy5., which is the three-letter
month abbreviation and the two-position year
Create a variable 'pumonyr' as the reformatted date
value from the column 'PUDATE' in data1a
If the value in str1 is 'pudate' then do
Create a temporary dataset temp using the
dataset data1a
Put the value from the variable str1 into the macrovariable g_xvar

3.6.2.8 Notes

This is a required window.

3.6.2.9 Program screens

Figure 3.6.6 represents the screen the user will see. Figure 3.6.7 contains the field names
in parentheses on the program screen.
Data Selection

- Select pie variable
- Select calculation variable

Enter 'X' beside your choice.
Press RETURN.

Press F10 to exit this screen.

Fig. 3.6.6. The PIEDATA.PROGRAM screen.

Data Selection

(var1) Select pie variable
(var2) Select calculation variable

Enter 'X' beside your choice.
Press RETURN.

Press F10 to exit this screen.

Fig. 3.6.7. The PIEDATA.PROGRAM screen with field names.

3.63 BRANGE.PROGRAM

3.63.1 Program purpose

BRANGE.PROGRAM allows the user to specify one minimum and maximum boundary for data values for the calculation variable of a bar chart. BRANGE.PROGRAM is called if the user chooses 'weight' as the calculation variable.

3.63.2 Calling program

PIEDATA.PROGRAM
3.63.3 Programs called

N/A

3.63.4 Program returns to

PIEDATA.PROGRAM

3.63.5 SAS window functions called by program

N/A

3.63.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialized</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Set

g_high contains the maximum value

  g_low contains the minimum value

3.63.7 Structured English for program code

When this program is called
Put a '0' in the lower field
Put '9999' in the upper field
When the user exists this program
Put the value in the lower field into the
g_low macrovariable
Put the value in the upper field into the
g_high macrovariable

3.63.8 Notes

This program is also called by the LINEDATA.PROGRAM and the BARDATA.PROGRAM.

3.63.9 Program screens

Figure 3.6.8 represents the screen the user will see. Figure 3.6.9 contains the field names in parentheses on the program screen.
3.6.4 TIME.PROGRAM

3.6.4.1 Program purpose

TIME.PROGRAM allows the user to select a date type for the output. The user may choose to display dates as pickup month; pickup year; pickup month and year; or as pickup day, month, and year.

3.6.4.2 Calling program

PIEDATA.PROGRAM
3.6.4.3 Programs called

N/A

3.6.4.4 Program returns to

PIEDATA.PROGRAM

3.6.4.5 SAS window functions called by program

N/A

3.6.4.6 Macrovariables

Used - N/A

Initialized - N/A

Set - g_date contains the string 'pumon', 'puyr', 'pumonyr', or 'pudate'

3.6.4.7 Structured English for program code

When user presses <return> while in the mon field
   If mon field does not contain a '*' or a blank then do
      Put a '*' in the mon field
      Position cursor on the command line
      Put string 'pumon' in macrovariable g_date
   Else position cursor in yr field

When user presses <return> while in the yr field
   If yr field does not contain a '*' or a blank then do
      Put a '*' in the yr field
      Position cursor on the command line
      Put string 'puyr' in the macrovariable g_date
   Else position cursor in monyr field

When user presses <return> while in the monyr field
   If monyr field does not contain a '*' or a blank then do
      Put a '*' in the monyr field
      Position cursor on the command line
      Put string 'pumonyr' in the macrovariable g_date
   Else position cursor in dmyr field
When user press <return> while in the dmyr field
If dmyr field does not contain a '*' or a blank then do
Put a '*' in the dmyr field
Position cursor on the command line
Put string 'pudate' in the macrovariable g_date
Else position cursor on command line

3.6.4.8 Notes

This program is also used by TABDATA.PROGRAM and BARDATA.PROGRAM.

3.6.4.9 Program screens

Figure 3.6.10 represents the screen the user will see. Figure 3.6.11 contains the field names in parentheses for the program screen.

<table>
<thead>
<tr>
<th></th>
<th>Month</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Year</td>
</tr>
<tr>
<td></td>
<td>Month/Year</td>
</tr>
<tr>
<td></td>
<td>Day/Month/Year</td>
</tr>
</tbody>
</table>

Fig. 3.6.10. The TIME.PROGRAM screen.

<table>
<thead>
<tr>
<th>(mon)</th>
<th>Month</th>
</tr>
</thead>
<tbody>
<tr>
<td>(yr)</td>
<td>Year</td>
</tr>
<tr>
<td>(monyr)</td>
<td>Month/Year</td>
</tr>
<tr>
<td>(dmyr)</td>
<td>Day/Month/Year</td>
</tr>
</tbody>
</table>

Fig. 3.6.11. The TIME.PROGRAM screen with field names.
3.6.5 POUTOPT.PROGRAM

3.6.5.1 Program purpose

POUTOPT.PROGRAM allows the user to choose the destination for the output of the pie chart. The user may choose to send the output to a printer, to a file, or to the monitor screen.

3.6.5.2 Calling program

PIE.PROGRAM

3.6.5.3 Program called

PREADY.PROGRAM

3.6.5.4 Program returns to

PIE.PROGRAM

3.6.5.5 SAS window functions called by program

N/A

3.6.5.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>Initialized</th>
<th>Set</th>
</tr>
</thead>
</table>
| N/A  | N/A         | g_outopt contains a string holding the destination option ('screen' or 'print') the user has chosen

3.6.5.7 Structured English for program code

Put a blank in the print field, the os file field, the ffile field, and the screen field
When user presses <return> while in the print field then do
   If print field does not contain a '*' or a blank then do
     Put a '*' in the print field
     Put the string 'print' into the macrovariable g_outopt
   Call PREADY.PROGRAM
3.6.5.8 Notes

The options for sending output to an operating system file or a Freelance file are not functional in the prototype. If the user chooses either of these options, he/she will see a message that explains that the function is not available in this version.

The 'next field' for the last field, the screen field, is the command line.

3.6.5.9 Program screens

Figure 3.6.12 represents the screen the user will see. Figure 3.6.13 contains the field names in parentheses for the program screen.
Output Options

___ Print Output

___ Save Output to an operating system file

___ Save Output to a FreeLance file

___ Send Output to the screen

Enter 'X' beside your choice.
Press RETURN.

Press F10 to exit this screen.

Fig. 3.6.12. The POUTOPT.PROGRAM screen.

Output Options

(print) Print Output
(osfile) Save Output to an operating system file
(ffile) Save Output to a FreeLance file
(screen) Send Output to the screen

Enter 'X' beside your choice.
Press RETURN.

Press F10 to exit this screen.

Fig. 3.6.13. The POUTOPT.PROGRAM screen with field names.
3.6.6 PREADY.PROGRAM

3.6.6.1 Program purpose

PREADY.PROGRAM sends the output to the destination chosen by the user in POUTOPT.PROGRAM. This program uses dataset data2.

3.6.6.2 Calling program

POUTOPT.PROGRAM

3.6.6.3 Programs called

N/A

3.6.6.4 Program returns to

POUTOPT.PROGRAM

3.6.6.5 SAS window functions called by program

N/A

3.6.6.6 Macrovariables

<table>
<thead>
<tr>
<th>Used</th>
<th>g_calvar</th>
<th>contains a string holding the name of the calculation variable the user chose in PIEDATA.PROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>g_nc</td>
<td>see macrovariables set in this program</td>
<td></td>
</tr>
<tr>
<td>g_outopt</td>
<td>contains a string holding the output destination the user chose in POUTOPT.PROGRAM (i.e., 'screen' or 'print')</td>
<td></td>
</tr>
<tr>
<td>g_str2</td>
<td>see macrovariables set in this program</td>
<td></td>
</tr>
<tr>
<td>g_str3</td>
<td>see macrovariables set in this program</td>
<td></td>
</tr>
<tr>
<td>g_xvar</td>
<td>contains a string holding the name of the pie variable the user chose in PIEDATA.PROGRAM</td>
<td></td>
</tr>
</tbody>
</table>
3.6.6.7 Structured English for program code

Create variable outstr as the value in the macrovariable g_calopt
Create variable xvar as the value in the macrovariable g_xvar
Create the variable num as the SAS number of the column named in the variable xvar
Create the variable vtype as the variable type of the SAS number contained in the variable num
If the value in vtype is 'N' then
   Put the string 'discrete' into the macrovariable g_nc
Else
   Put a blank into the macrovariable g_nc
If the value in the variable outstr is 'screen' then
   Display message 'Press F10 to return to this screen then press F10 to exit.'
Else if the value in the variable outstr is 'print' then
   Display message 'Press F10 to exit this screen.'
If the value in the variable outstr is 'screen' then do
   Put the string 'device=sun colors = (cream gold yellow tan olive rose blue violet green brown)' into the macrovariable g_str2
   Put the string 'fill=solid' into the macrovariable g_str3
If the value in the variable outstr is 'print' then do
   Put the string 'gaccess= "sasgacmd> lpr" device=hpljs2' into the macrovariable g_str2
   Put a blank into the macrovariable g_str3
Define options as background color = cyan, text color = black, and the destination is the value in the macrovariable g_str2
Generate a pie chart using dataset data2, using the value in the macrovariable g_xvar as the pie variable, using the value in the macrovariable g_calvar as the calculation variable, printing the percentages inside each slice, outlining the pie in black, include the value of the macrovariable g_nc, tilt the pie chart 150 degrees, include the value in the macrovariable g_str3

3.6.6.8 Notes

This screen will display a message as the user waits for output to the screen or printer. The device names for the printers or screens are contained in the macrovariable g_str2 and can easily be changed to accommodate different hardware. It is also easy to change the colors and other options set in the gchart and goptions statements.

3.6.6.9 Program screen

There are no enterable fields in this program screen. A full-sized screen will appear with a message and instructions depending on whether the user chose 'screen' or 'print' in POUTOPT.PROGRAM. Figure 3.6.14 represents the screen the user will see.

![Processing — One moment please.](image)

Fig. 3.6.14. The PREADY.PROGRAM screen.
4. SUMMARY AND CONCLUSIONS

The MTPP staff members are responsible for monitoring almost one million shipments annually. They must evaluate the Personal Property Program and investigate the effects of policy changes. Currently, MTPP staff spend the majority of their time manually gathering data, organizing data, checking data, and preparing reports. They have little time to analyze data that identify trends, predict impacts of trends and other changes on the Personal Property Program, or to recommend policy changes based on their analysis of the program.

The user applications module designed and prototyped for WHIST-MOD will automate many of the tasks now performed manually by MTPP staff. This automation will allow the staff to spend more time analyzing the Personal Property Program. In addition, the front- and back-end interfaces will allow the user quick access to data and the ability to manipulate and display these data easily.

The back-end interface meets the goals identified during the requirements analysis phase of the project. Its design is modular and easily supports modifications and expansions. This interface is generic enough to support the selection and manipulation of data for four types of output for each of 36 report types. It is also easy-to-use and does not require the user to know anything about SAS procedures or the SAS/AF Screen Control Language. The staff at MTPP are pleased with the prototype and anxious to see it fully implemented so that they can begin to use this software to aid them in their decision-making processes.

The SAS back-end prototype extensively utilized the SAS/AF macrovariable facility and many of the default settings for the graphical output produced using SAS/GRAPH. By using the defaults when possible, the software is capable of producing a variety of basic reports for each report type. These default settings, used in conjunction with values passed through macrovariables, made it possible for ORNL to produce a prototype interface that was even more flexible and generic than the conceptual design for the interface.

This interface does not require a separate program for each specific report the user needs to produce. The procedures ORNL used for prototyping the back-end interface could be used by other projects to develop a flexible and generic user interface.
INTERNAL DISTRIBUTION

1. R. B. Braid
2. T. L. Chiang
3. R. Goeltz
4-8. T. James
9. M. A. Kuliasha
10. J. Loftis
11-12. R. L. Noe
13. P. Singley
14. T. Truett
15. D. Vogt
16. ORNL Patent Office
17. Central Research Office
18. Document Reference Section
19-21. Laboratory Records
22. Laboratory Records (RC)

EXTERNAL DISTRIBUTION

23. B. G. Buchanan, Department of Computer Science, University of Pittsburgh, 206 Mineral Industries Building, Pittsburgh, PA 15260
24. J. J. Cuttca, Vice President of Research and Development, Gas Research Institute, 8600 W. Bryn Mawr Avenue, Chicago, IL 60631
25. S. W. Jebo, Military Traffic Management Command Headquarters, PPM, 5611 Columbia Pike, Falls Church, VA 22041-5050
26. D. E. Morrison, 333 Oxford Road, East Lansing, MI 48823
27. A. Hirsch, President, Dynamac Corporation, The Dynamac Building, 11140 Rockville Pike, Rockville, MD 20852
28. M. Williams, Professor, Department of Economics, Northern Illinois University, DeKalb, IL 60115
29. Office of Assistant Manager for Energy Research and Development, DOE/ORO, P.O. Box 2001, Oak Ridge, TN 37831-8600