NASA TECHNICAL MEMORANDUM

NASA TM X-64862

By Ronald A. Schlagheck, Bert D. Dolierhie, Jr., and Fred J. Ghiglieri
Systems Analysis and Integration Laboratory

June 1974

NASA

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama
The GSE Data Management System is a computerized program which provides for a central storage source for key data associated with the mechanical ground support equipment (MGSE). Eight major sort modes can be requested by the user. Attributes that are printed automatically with each sort include the GSE End Item number, description, class code, functional code, fluid media, use location, design responsibility, weight, cost, quantity, dimensions, and applicable documents. Multiple subsorts are available for the class code, functional code, fluid media, use location, design responsibility, and applicable document categories. This manual includes a description of these sorts and how to use them. The program and GSE data bank may be easily updated and expanded.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. PROGRAM OPERATION INSTRUCTIONS</td>
<td>3</td>
</tr>
<tr>
<td>Control Card Description</td>
<td>3</td>
</tr>
<tr>
<td>GSE Data Card Input</td>
<td>6</td>
</tr>
<tr>
<td>Program Instruction Data Input</td>
<td>6</td>
</tr>
<tr>
<td>III. SAMPLE OUTPUT</td>
<td>11</td>
</tr>
<tr>
<td>Description of Sort Types</td>
<td>11</td>
</tr>
<tr>
<td>User Messages</td>
<td>17</td>
</tr>
<tr>
<td>Error Messages</td>
<td>22</td>
</tr>
<tr>
<td>IV. PROGRAM ASPECTS</td>
<td>27</td>
</tr>
<tr>
<td>Program Execution</td>
<td>27</td>
</tr>
<tr>
<td>Program Modification Procedures</td>
<td>29</td>
</tr>
<tr>
<td>Changes to Program Procedures Tables</td>
<td>30</td>
</tr>
<tr>
<td>Changing and Adding Fortran Routines</td>
<td>33</td>
</tr>
<tr>
<td>Deck Setup, Control Cards, and Options</td>
<td>34</td>
</tr>
<tr>
<td>Program Characteristics</td>
<td>36</td>
</tr>
<tr>
<td>Description of the Program Listing</td>
<td>36</td>
</tr>
<tr>
<td>APPENDIX A. DEFINITION OF INPUT FORMAT FOR GSE</td>
<td>37</td>
</tr>
<tr>
<td>ATTRIBUTE DATA CARDS</td>
<td></td>
</tr>
<tr>
<td>Card Type 1</td>
<td>39</td>
</tr>
<tr>
<td>Card Type 2</td>
<td>42</td>
</tr>
<tr>
<td>Card Type 3</td>
<td>43</td>
</tr>
<tr>
<td>Card Type 4</td>
<td>47</td>
</tr>
<tr>
<td>APPENDIX B. GSE DATA MANAGEMENT SYSTEM PROGRAM</td>
<td>49</td>
</tr>
<tr>
<td>LISTING</td>
<td></td>
</tr>
<tr>
<td>APPENDIX C. SUPPLEMENT RUN PROCEDURES</td>
<td>99</td>
</tr>
</tbody>
</table>

Preceding page blank
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Control card deck setup</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>Pictorial control card deck setup</td>
<td>5</td>
</tr>
<tr>
<td>3.</td>
<td>Control card deck with GSE data input and all program instruction data types</td>
<td>7</td>
</tr>
<tr>
<td>4.</td>
<td>Control card deck with multiple sorts of the same mnemonic code</td>
<td>9</td>
</tr>
<tr>
<td>5.</td>
<td>Example page of Master Summary Table</td>
<td>12</td>
</tr>
<tr>
<td>6.</td>
<td>Example of Stage Sort One Table</td>
<td>13</td>
</tr>
<tr>
<td>7.</td>
<td>Example of Stage Sort Two Table</td>
<td>14</td>
</tr>
<tr>
<td>8.</td>
<td>Example of Media Sort Table</td>
<td>15</td>
</tr>
<tr>
<td>9.</td>
<td>Example of Functional Sort Table</td>
<td>16</td>
</tr>
<tr>
<td>10.</td>
<td>Example of Class Code Sort Table</td>
<td>18</td>
</tr>
<tr>
<td>11.</td>
<td>Example of Use Location Sort Table</td>
<td>19</td>
</tr>
<tr>
<td>12.</td>
<td>Example of Applicable Document Sort Table</td>
<td>20</td>
</tr>
<tr>
<td>13.</td>
<td>Master program execution flow chart</td>
<td>28</td>
</tr>
<tr>
<td>14.</td>
<td>Procedure Table catalogue listing</td>
<td>31</td>
</tr>
<tr>
<td>15.</td>
<td>Control card deck with program modifications</td>
<td>35</td>
</tr>
<tr>
<td>A-1.</td>
<td>GSE input format sheet</td>
<td>40</td>
</tr>
<tr>
<td>A-2.</td>
<td>Example of GSE filled-in data sheet</td>
<td>41</td>
</tr>
<tr>
<td>C-1.</td>
<td>Sample Run Card No. 1</td>
<td>102</td>
</tr>
<tr>
<td>C-2.</td>
<td>Sample Run Card No. 2</td>
<td>103</td>
</tr>
</tbody>
</table>
SECTION I. INTRODUCTION

The Ground Support Equipment (GSE) Data Management System (DMS) is a computer program that was developed to provide a central storage source for key data associated with mechanical GSE. The system is fast, accurate, and easy to use by noncomputer-oriented individuals. This document describes the program and how to use it.

The GSE DMS takes large amounts of GSE data and sorts it alphabetically and numerically (by letter and number) into various categories. For example, a typical user of the program might want a presentable tabular listing of all GSE in the Skylab project associated with a particular vehicle module, such as the Orbital Workshop (OWS). Corresponding GSE attributes [End Item (EI) numbers, descriptions, quantities, fluid media, etc.] are displayed as a major part of this listing. There is a total of eight different types of computer printouts currently available to the user. Seven of these consist of GSE data that have been sorted alphabetically under a particular category. The previous example, Skylab OWS, would be found under what is known as a Stage Sort. Other sort categories include groupings by fluid media, functional code, class code, applicable document, and use location. The other printout available is the master summary table. It is basically an executive inventory list of all the GSE data that are currently in the data bank. Section II, Program Operations Instructions, provides the procedures for data card input format and defines the various program options. Section III illustrates the various example outputs and describes the error and user messages that are included in the GSE DMS. Section IV, Program Aspects, is intended for those individuals who wish to know more about the computer program itself and it has a general explanation of the internal operations of the computer program. Also included in this section is a procedure on how to make temporary or permanent changes to the program.
SECTION II. PROGRAM OPERATION INSTRUCTIONS

The Ground Support Equipment (GSE) Data Management System (DMS) is designed for execution by the Marshall Space Flight Center (MSFC) Univac 1108 Exec VIII Computer System. The GSE Data Management System consists of Fortran programming and a current GSE data bank. For convenience of the user, the complete system has been stored on magnetic tape to permit accessibility from remote-site operation.

The GSE Data Management System is capable of performing eight different major types of sorts using the GSE data bank. The GSE data bank may have data added, updated, and/or deleted as required by the user. If these changes are to be permanent, the user can have a new tape made containing the revised GSE data bank.

Control Card Description

Figures 1 and 2 show the control card deck setup required for execution of the GSE Data Management System. The deck indicates the location for GSE data cards when applicable and the location of the instruction data cards for sort requests. It is imperative that the control cards (cards with the '@' sign punched in column 1) be punched correctly and placed in the sequence as shown in Figures 1 and 2.

Run card no. 1, first control card, is the accounting and identification card for the computer run. Certain user information must be supplied on this card and these parameters will be explained in the remainder of this section. SAMPLE is the run I.D. code and may be any six letter combination assigned by the user and AAAAAA is a six digit number which is the accounting job number assigned to the user. NAMEXX is the programmer's name in six characters, XYZ returns the computer run to the programmer's BIN number, 3 is the maximum central processing unit (CPU) run time expected, and 200 represents the maximum number of pages of output expected. Run card no. 2, a special preprinted green card, is a duplicate of the first card. Appendix C gives the details of completing this control card. A four or five digit number, represented by BBBBB on the third control card, is the current reel number of the magnetic tape which contains the GSE Data Management System. The current tape number can be obtained from the publishing organization. The remaining control cards should be exactly as shown.
@RUN; // P SAMPLE; AAAAAA; NAMExXBIXYZ; 3, 200
@RUN; // P SAMPLE; AAAAAA; NAMExXBIXYZ; 3, 200
@ASGT; GSE-OLD-TAPE*; T, BUBBR
@REWIND; GSE-OLD-TAPE*
@ASGT; 1, F2
@ASGT; 2, F2
@ASGT; 3, F2
@ASGT; 4, F2
@ASGT; CONTROL-FILE*; F/1/TK/K1O
@COPY; GSE-OLD-TAPE*; TPFS
@COPY; GSE-OLD-TAPE*; CONTROL-FILE*
@COPY; GSE-OLD-TAPE*; 1*
@DATAtIL 2*

GSE DATA CARDS HERE

@END
@AW T PROVE

INSTRUCTION DATA CARDS HERE

@FIN

Figure 1. Control card deck setup.
Figure 2. Pictorial control card deck setup.
GSE Data Card Input

The GSE data bank may have data added, updated, and/or deleted. If no GSE data are to be added, updated, and/or deleted from the data bank, one blank computer card must be placed between the @DATA, IL 2 and @END control cards. To add a new item of GSE to the data bank or to update an existing item of GSE in the data bank, the set of cards for each item must follow the card formats described in Appendix A. To delete a piece of GSE from the data bank, one card must be prepared. In column 1 of a computer card the number 1 must be punched and beginning in column 7 the EI number, exactly as it was originally entered into the data bank, must be punched. The remainder of the computer card must be left blank. It is suggested that any deletion cards be placed before any new and/or updated GSE data cards. Figure 3 shows a control card deck with a deletion and an addition to the GSE data bank; however, any single option or combination of the three options is allowable.

Program Instruction Data Input

The program instruction data consists of two types of data cards. Type #1 contains the date of the computer run and an option of whether or not a new GSE Data Management System tape is to be created. Type #2 is the code mnemonic for the sort to be performed.

TYPE #1 INSTRUCTION DATA CARD

In columns 1 through 8 inclusive, the user must punch the date that is desired to be printed on the output tables. The date should follow the format shown in Figure 3. This date will also be recorded with any new GSE data that have been added to the master data file. If a new GSE Data Management System tape is desired because of a GSE data revision or other change, punch a T in column 11 of this same card. If no tape is desired, punch only the date and leave the remainder of the card blank. Only one Type #1 instruction data card is allowed in a control card deck. Figure 3 shows an example of Type #1 instruction card with no tape being created.

TYPE #2 INSTRUCTION DATA CARD(S)

Each Type #2 data card specifies a sort selected for output. One Type #2 data instruction card is required for each sort, but the user may place as many as eight of these cards in each control card deck. Special codes punched
Figure 3. Control card deck with GSE data input and all program instruction data types.
on these cards contain the necessary information to perform the desired sort. The sort code mnemonics that the user must use are listed below:

<table>
<thead>
<tr>
<th>Sort Code Mnemonic</th>
<th>Type of Sort</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>Master Summary Table</td>
</tr>
<tr>
<td>STAGE1</td>
<td>Stage Sort 1</td>
</tr>
<tr>
<td>STAGE2</td>
<td>Stage Sort 2</td>
</tr>
<tr>
<td>MEDIAS</td>
<td>Fluid Media</td>
</tr>
<tr>
<td>FUNCTL</td>
<td>Functional Code</td>
</tr>
<tr>
<td>CLASSC</td>
<td>Class Code</td>
</tr>
<tr>
<td>USELOC</td>
<td>Use Location</td>
</tr>
<tr>
<td>APPDOC</td>
<td>Applicable Document</td>
</tr>
</tbody>
</table>

Explanations of each of the sort types and the type of output are given in Section III.

For the Master Summary Table, Stage Sort 1, or Stage Sort 2 sorts, punch their mnemonic code beginning in column 1 of a computer card and leave the remainder of the card blank. When requesting the Fluid Media, Functional Code, Class Code, or Use Location sorts, punch their mnemonic code beginning in column 1 of a computer card. Since there are multiple classifications within each of these four sorts, a second type of code is also required which specifies detailed information about the sorts. This code is punched beginning in column 11 of the same computer card and is known as the Descriptor Code. The remainder of the card is left blank. The Descriptor Codes for each of these sorts can be found in Appendix A in the discussion of their use in preparing GSE data for the GSE data bank. These codes are used to provide detailed information about the GSE and must be used to relocate this information during these sorts. For the Applicable Document Sort, punch the mnemonic code beginning in column 1 of a computer card, and beginning in column 11 punch the document number to be searched for during the sort. Figure 3 shows an example of each of the eight possible sorts.

For Fluid Media, Functional Code, Class Code, Use Location, and Applicable Document sorts, multiple sorts of each can be requested. For example, Four Fluid Media sorts could be performed using one control card deck by punching a different Descriptor Code on each of four MEDIAS sort request cards. However, a maximum of eight sort requests are allowed. Figure 4 shows an example control card deck with four Fluid Media and two Applicable Document sorts.
Figure 4. Control card deck with multiple sorts of the same mnemonic code.
SECTION III. SAMPLE OUTPUT

Eight different types of sorts are available for selection to generate the outputs from the GSE Data Management System. The user can request sorts by stage groups, fluid media, functional operation, ground operation class, use location, and applicable document. A summary of the available records in the master data bank can also be outputed. The total system is a user oriented retrieval method that can display various types of GSE characteristics on different output formats.

The GSE Data Management System provides error messages and user messages that can be encountered as part of the program's output. These messages are associated with the GSE attribute data, sort requests, and tape option.

Description of Sort Types

The Master Summary Table provides the user with a complete listing of all records contained in the data bank. The table contains the EI number, program title, stage name, and reference date. Figure 5 is an example page of the Master Summary Table output. This table permits one to determine when a particular EI was last updated or changed.

The Stage Sort 1 and Stage Sort 2 outputs are listed by program type and stage, module, or system. The Stage Sort 1 output lists for each EI number the description, class code, functional code, use location, and associated applicable documents. The Stage Sort 2 output lists for each EI number the description, fluid media, weight, length, width, height, quantity, unit cost in purchase year, and design responsibility. Figure 6 is an example page of the Stage Sort One Table output, and Figure 7 is an example page of the Stage Sort Two Table output. A complete listing of all entities contained in the data bank is obtained when Stage Sort One and Two is requested.

The Fluid Media sort provides an output table which contains a complete list of all GSE containing the specified type of fluid media. The Media Sort Table output for the requested fluid media contains for each item of GSE the EI number, description, function code, weight, length, width, height, quantity, unit cost in purchase year, and design responsibility. Figure 8 shows an example page from a Media Sort Table output. The functional code sort provides an output that contains a complete list of GSE that services a particular operational requirement. When a specific operational requirement
### Master Summary Table

<table>
<thead>
<tr>
<th>ENTITY</th>
<th>EI NUMBER</th>
<th>PROGRAM</th>
<th>STAGE</th>
<th>REFERENCE DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>2</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>3</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>4</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>5</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>6</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>7</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>8</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>9</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>10</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>11</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>12</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>13</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>14</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>15</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>16</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>17</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>18</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>19</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>20</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>21</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>22</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>23</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>24</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>25</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>26</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>27</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>28</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>29</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>30</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>31</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>32</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>33</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>34</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>35</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>36</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>37</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>38</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>39</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>40</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>41</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>42</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>43</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>44</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
<tr>
<td>45</td>
<td>CD1089K020A</td>
<td>SHUTTLE</td>
<td>SHUTTLE MAIN ENGN.</td>
<td>02/01/73</td>
</tr>
</tbody>
</table>

Figure 5. Example page of Master Summary Table.

is specified, the output table contains the EI number, description, fluid media, weight, length, width, height, quantity, unit cost in purchase year, and design responsibility for each item of GSE that performs the function. Figure 9 shows an example page from a Functional Sort Table output.
<table>
<thead>
<tr>
<th>CLASS CODE</th>
<th>DESCRIPTION</th>
<th>USE</th>
<th>APPLICABLE DOCUMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>G7-435040-356</td>
<td>STATIONARY PNEUMATIC CONSOLE - ENCLOSED</td>
<td>KSC, SST, MTF</td>
<td>651C9791, 651C9791, 651C9791</td>
</tr>
<tr>
<td>G7-439000-91C</td>
<td>STATIONARY PNEUMATIC CONSOLE - OPEN</td>
<td>KSC, SST, MTF</td>
<td>651C9791, 651C9791, 651C9791</td>
</tr>
<tr>
<td>G7-439000-91D</td>
<td>STATIONARY HEAT EXCHANGER - STATIONARY</td>
<td>KSC, MTF</td>
<td>651C9791, 651C9791, 651C9791</td>
</tr>
</tbody>
</table>

Figure 6. Example of Stage Sort One Table.
<table>
<thead>
<tr>
<th>EI Number</th>
<th>Description</th>
<th>Media</th>
<th>WT</th>
<th>L</th>
<th>W</th>
<th>H</th>
<th>GTY</th>
<th>Cost/Yr</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>010600003A</td>
<td>Floa Tester, Pneumatic - Atmospheric</td>
<td>GN2,HE</td>
<td>TBD</td>
<td>LB</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD CM</td>
<td>Rocketdyne</td>
</tr>
<tr>
<td>010600002A</td>
<td>Floa Tester, Pneumatic - High Pres.</td>
<td>HE</td>
<td>TBD</td>
<td>LB</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD CM</td>
<td>Rocketdyne</td>
</tr>
<tr>
<td>010600003A</td>
<td>Leak Detector, Helium Mass Spec.</td>
<td>HE</td>
<td>TBD</td>
<td>LB</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD CM</td>
<td>Rocketdyne</td>
</tr>
<tr>
<td>010600002A</td>
<td>Tool Set, Special for Space Shuttle Main Engine</td>
<td>N/A</td>
<td>TBD</td>
<td>LB</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD CM</td>
<td>Rocketdyne</td>
</tr>
</tbody>
</table>

NOTE: The last letter in EI Number indicates the effective period of contract. A is for GSE used in engine development, B is for GSE used in flight operations.

N/A or NA - NOT APPLICABLE

Figure 7. Example of Stage Sort Two Table.
<table>
<thead>
<tr>
<th>EL NUMBER</th>
<th>DESCRIPTION</th>
<th>FUNCTIONAL</th>
<th>ENVELOPE DIM</th>
<th>UNIT</th>
<th>DESIGN</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD109R001A</td>
<td>FLOW TESTER, PNEUMATIC - ATMOSPHERIC FOR SPACE SHUTTLE MAIN ENGINE (SSME)</td>
<td>TESTCH</td>
<td>TBD</td>
<td>TBD CH</td>
<td>TBD CH</td>
</tr>
<tr>
<td></td>
<td>NOTE: LAST LETTER IN EL NUMBER INDICATES THE EFFECTIVE PERIOD OF CONTRACT (A IS FOR GSE USED IN ENGINE DEVELOPMENT, B IS FOR GSE USED IN FLIGHT OPERATIONS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD109R002A</td>
<td>FLOW TESTER, PNEUMATIC - HIGH PRESS. FOR SPACE SHUTTLE MAIN ENGINE (SSME)</td>
<td>TESTCH</td>
<td>TBD</td>
<td>TBD CH</td>
<td>TBD CH</td>
</tr>
<tr>
<td></td>
<td>NOTE: LAST LETTER IN EL NUMBER INDICATES THE EFFECTIVE PERIOD OF CONTRACT (A IS FOR GSE USED IN ENGINE DEVELOPMENT, B IS FOR GSE USED IN FLIGHT OPERATIONS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD109R003A</td>
<td>LEAK DETECTOR, HELIUM MASS SPECS. FOR SPACE SHUTTLE MAIN ENGINE (SSME)</td>
<td>TESTCH</td>
<td>TBD</td>
<td>TBD CH</td>
<td>TBD CH</td>
</tr>
<tr>
<td></td>
<td>NOTE: LAST LETTER IN EL NUMBER INDICATES THE EFFECTIVE PERIOD OF CONTRACT (A IS FOR GSE USED IN ENGINE DEVELOPMENT, B IS FOR GSE USED IN FLIGHT OPERATIONS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD209R001A</td>
<td>CONSOLE, ROCKET ENGINE ELECTRICAL PNEUMATIC-CHECKOUT FOR SPACE SHUTTLE MAIN ENGINE (SSME)</td>
<td>TESTCH</td>
<td>TBD</td>
<td>TBD CH</td>
<td>TBD CH</td>
</tr>
<tr>
<td></td>
<td>NOTE: LAST LETTER IN EL NUMBER INDICATES THE EFFECTIVE PERIOD OF CONTRACT (A IS FOR GSE USED IN ENGINE DEVELOPMENT, B IS FOR GSE USED IN FLIGHT OPERATIONS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

N/A OR NA = NOT APPLICABLE

Figure 8. Example of Media Sort Table.
## Functional Sort Table

All Programs

### Test-Checkout Equipment

<table>
<thead>
<tr>
<th>E1 Number</th>
<th>Description</th>
<th>Media</th>
<th>AT</th>
<th>L</th>
<th>H</th>
<th>W</th>
<th>Unit</th>
<th>Design</th>
<th>Cost/Yr Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD395RO002A A</td>
<td>Covers - Protective, Engine Thrust Chamber for Space Shuttle Main Engine (SSME)</td>
<td>N/A</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td>ROCKETDYNE</td>
</tr>
<tr>
<td></td>
<td>Note: Last letter in E1 number indicates the effective period of contract.</td>
<td></td>
<td>AT</td>
<td>L</td>
<td>H</td>
<td>W</td>
<td>Unit</td>
<td>Design</td>
<td>Cost/Yr Responsibility</td>
</tr>
<tr>
<td></td>
<td>A is for GSE used in engine development. H is for GSE used in flight operations.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

N/A or NA - Not Applicable

Figure 9. Example of Functional Sort Table.
The class code sort option provides an output which contains a complete list of the GSE that supports the major hardware category specified. The EI number, description, function code, stage, fluid media, quantity, and design responsibility for each item of GSE associated with the requested operation is contained in the output table. Figure 10 shows an example page from a Class Code Sort Table output. The use location sort option generates a complete list of all GSE that is used at a specific geographical location or facility. The output contains the EI number, description, function code, weight, length, width, height, fluid media, and design responsibility for each item of GSE used in the specified location. Figure 11 shows an example page from a Use Location Sort Table output. The applicable document sort provides an output which contains all GSE that is referenced in the requested document. The output contains the EI number, description, class code, function code, and associated applicable documents. Figure 12 shows an example page from an Applicable Document Sort Table output.

User Messages

The GSE Data Management System can have data added, updated, and/or deleted from the data bank. When no GSE attribute data are to be changed, the message shown in Example 1 is printed showing that no changes occurred. The total number of GSE items in the data bank is also shown.

***GSE ATTRIBUTE DATA FILE HAS BEEN UPDATED***
0 NEW ENTITIES
0 UPDATED ENTITIES
0 DELETED ENTITIES
171 TOTAL ENTITIES USED DURING THIS RUN

Example 1

If any changes to the GSE attribute data have been included in the control card deck, the number of each specific type of change will be printed out along with the new total number of GSE items now in the data bank. Example 2 shows a total of five changes to the GSE data bank and the new total.
### Class Code Sort Table

<table>
<thead>
<tr>
<th>E1 Number</th>
<th>Description</th>
<th>Function</th>
<th>Cool</th>
<th>Stage</th>
<th>Media</th>
<th>UTT</th>
<th>Design Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPDEW105619</td>
<td>S-IC Pneumatic Portable Testers--Six portable suitcases containing switches, indicator lights, and meters used in checking continuity of electrical components of the S-IC pneumatic console and the S-IC pneumatic checkouts.</td>
<td>TESTCH</td>
<td>S-IC</td>
<td>N/A</td>
<td>16</td>
<td>GLMUNTVILLE</td>
<td></td>
</tr>
<tr>
<td>05Y-40-979</td>
<td>Power Cable Heel Cart--This mobile unit consists of storage reel, crank, frame, semi-pneumatic wheels, two water hoses, electrical power cable, and quick disconnects. The unit is used to supply cooling water and electrical power to the OSV-48-78.</td>
<td>SERVIC</td>
<td>S-170</td>
<td>N/A</td>
<td>8</td>
<td>ASTM-SOG</td>
<td></td>
</tr>
<tr>
<td>22A1206</td>
<td>Test Set Assy, Hydraulic Performance--A mobile unit containing switches, indicator lights, cables, and a manifold assembly. The tester simulates the step function flow requirements of the S-IC F-1 engines start sequence to checkout the S-IC hydraulic supply and checkout unit.</td>
<td>TESTCH</td>
<td>S-IC</td>
<td>N/A</td>
<td>01</td>
<td>GLMUNTVILLE</td>
<td></td>
</tr>
<tr>
<td>65036640</td>
<td>Test Set Assy, Forward Umbilical Serviceer Console--A portable suitcase containing switches, indicator lights, meters, and cables to test the S-IC forward umbilical service console.</td>
<td>TESTCH</td>
<td>S-IC</td>
<td>N/A</td>
<td>03</td>
<td>GLMUNTVILLE</td>
<td></td>
</tr>
</tbody>
</table>

Figure 10. Example of Class Code Sort Table.
**Figure 11. Example of Use Location Sort Table.**

<table>
<thead>
<tr>
<th>ET NUMBER</th>
<th>DESCRIPTION</th>
<th>FUNCTIONAL</th>
<th>ENVELOPE DIM</th>
<th>MEDIA</th>
<th>DESIGN RESPONSIBILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSV04045A7</td>
<td>EXTERNAL TANK ELIHZ TEST TANK STORAGE CRADLE-USED TO SUPPORT THE LH2 TEST TANK IN THE HORIZONTAL POSITION DURING PERIODS OF LO GA ACTIVITY ON STORAGE-CONSTRUCTED OF COMMERCIAL GRADE STEEL SECTIONS,PLATES,AND HARDWARE NO ROTATIONAL CAPABILITY,CAN BE DISASSEMBLED FOR SHIPMENT.</td>
<td>TRANSP TBD Lb TBD in TBD in TBD in</td>
<td>TBD KG TBD cm TBD cm TBD cm</td>
<td>N/A</td>
<td>S6E-ASTN-ETT</td>
</tr>
<tr>
<td>DSV04045B8</td>
<td>EXTERNAL TANK ELIHZ TEST TANK STORAGE CRADLE-USED TO SUPPORT THE LOX TEST TANK IN THE HORIZONTAL POSITION DURING PERIODS OF LOA ACTIVITY ON STORAGE-CONSTRUCTED OF COMMERCIAL GRADE STEEL SECTIONS,PLATES,AND HARDWARE NO ROTATIONAL CAPABILITY,CAN BE DISASSEMBLED FOR SHIPMENT.</td>
<td>TRANSP TBD Lb TBD in TBD in TBD in</td>
<td>TBD KG TBD cm TBD cm TBD cm</td>
<td>N/A</td>
<td>S6E-ASTN-ETT</td>
</tr>
<tr>
<td>DSV48-286</td>
<td>PNEUMATIC CONSOLE PORTABLE TEST - THIS MODEL SIMILAR TO DSV-48-286A EXCEPT FOR IDENTIFICATION AND CHANGES IN ELECTRICAL CONNECTORS. THIS MODEL HAS SATURN V EFFECTIVITY.</td>
<td>TESTCH 190 lb 28 in 28 in 12 in</td>
<td>63 KG 71 CM 71 CM 30 CM</td>
<td>N/A</td>
<td>MDAC-WD</td>
</tr>
</tbody>
</table>

N/A OR NA = NOT APPLICABLE
### Figure 12. Example of Applicable Document Sort Table.

#### Applyable Document Sort Table

**Document:** 651CD9200

<table>
<thead>
<tr>
<th>E1 Number</th>
<th>Description</th>
<th>Class Code</th>
<th>Functional Code</th>
<th>Applicable Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>85010401</td>
<td>Hydraulic Feed Unit—Stationary Sub-Unit of S-IC Hydraulic Supply and Checkout Unit containing Pumps, Hoses, Valves, Gages, Filters, and a 500 Gallon Reservoir for the Purpose of Starting, Conditioning, and Delivering E1 Fluid Under Pressure to the Five F-1 Engines of the S-IC Stage During All Ground Phases of Engine and Flight Maintenance Checkout.</td>
<td>1</td>
<td>SERVICE</td>
<td>651CD9790</td>
</tr>
<tr>
<td>89098141</td>
<td>Ground Support Cooling Unit—Enclosed Pumaged Stationary Unit containing a Coolant Reservoir, Coolant Circulation Subsystem, Non-Extensible Subsystem, Pumps, Motors, and Chiller. It is used to Condition Circulating Media for Se and S-IVB UH Aiding Cooled Plates Cooling and Heat Dissipation.</td>
<td>1</td>
<td>SERVICE</td>
<td>651CD9790</td>
</tr>
<tr>
<td>374601</td>
<td>Inert Prefill Unit—Contains Two Separate Stationary Sub-Units, 892996—Collector Pumping Unit Ang 892996—Collector Console, Used for Delivering Inert Fluid to the S-IC Stage F-1 Engine Jactets for Prefill Filling.</td>
<td>1</td>
<td>SERVICE</td>
<td>651CD9790</td>
</tr>
<tr>
<td>892996</td>
<td>Hydraulic Pumping Unit—Stationary Sub-Unit of Inert Prefill Unit—Contains Valves, Pumps, Motors, Gages, and a 770 Gallon Fluid Reservoir, Used for Starting and Pumping Inert Fluid.</td>
<td>1</td>
<td>SERVICE</td>
<td>651CD9790</td>
</tr>
</tbody>
</table>

**N/A or NA = Not Applicable**

**Class Codes:** 1. STAGE SYSTEMS 2. PAYLOAD MODULE 3. EXPERIMENT
Example 2

At the end of each sort requested in the control card deck, a statement is printed stating to the user that the sort was completed. Example 3 is a simple statement from the completion of a Functional Code sort request. If a valid sort is requested for a GSE category that is not in the data bank, a message is printed stating the master file contained no data for the specified sort. If a sort was requested for the fluid media Freon and no Freon was used with any of the GSE items in the data bank, the message shown in Example 4 would be printed out.

Example 3

Example 4

When all the sorts associated with the control card deck (maximum of eight) have been performed, a message is printed showing the number of sorts actually performed. Example 5 shows this statement when six sorts were requested.

Example 5
If the user has changed the GSE data bank, made programming changes, or has other reasons, a new tape can be made. When a new tape is correctly requested using the Type #1 instruction data card, a message is printed out stating a new tape was created. Example 6 shows this statement. Each tape has a number which must be placed on the third control card of a sort request deck. When a new tape is created, a four or five digit number of the tape is found at the end of the computer run which made the tape. Example 7 shows the location of a five digit number which was generated during this run.

Example 6

```
S<<<<< USER MESSAGE... AN OUTPUT TAPE HAS BEEN CREATED ON THIS RUN >>>>>
```

Example 7

```
SQURT: 1/R=842.5MS, 1C=134, OC=139, DIA=1.2
4- SAVE BLANK 16/14 TAPE =1 50720
SERVICE 17/14 50720
4 10709
TIME: 00:01:12.324 IN: 268 OUT: 0 PAGES: 152
```

Error Messages

Several types of errors can be made when making changes to the GSE data bank. When adding or updating the data bank, a set of four card types must be prepared. If one of the cards in these sets is placed out of order or punched incorrectly, a statement will be printed giving the number of the card in error. These card numbers are tabulated with an input listing of the data set printed out at the beginning of each run. Example 8 shows the result of submitting five changes to the data bank with a card out of order or punched incorrectly in one set. The correct data sets will be incorporated. The message shown in Example 8 will also be printed when a deletion card is out of
order by being placed within the set of four card types for adding or updating of
of the GSE data bank. When punching the project number on card Type #1 or the
class code on card Type #2 of the four card types of the GSE attribute data, a
character other than a number could possibly be used. If this should occur, the
error message shown in Example 8 will be printed. An incorrect dollar sign
($) for a Type #1 continuation card, when adding or updating the GSE data bank,
will also result in the error message shown in Example 8.

Example 8

When adding or updating the GSE data bank, it is allowable to have up
to 20 lines of description which consist of one Type #1 card and 19 Type #1
continuation cards (see Appendix A). If more than 20 lines of description are
used, the error message shown in Example 9 will be printed out. The data
associated with this EI number will not be placed in the data bank, but all
correct data sets will be used.

Example 9

The Type #1 instruction data card contains the date of the computer
run and an option of creating a new tape. If the date is of an incorrect format,
the error in Example 10 will be printed. This error message will also be
printed out if the Type #1 instruction data card is not placed in the computer
card deck. This error will cause the run to be terminated. If a new tape is
to be created, a T is punched in column 11 of the Type #1 instruction data
When an incorrect character is used, the error message shown in Example 11 is printed. The requested sorts will be performed, but a new tape will not be made.

```
*** ERROR IN PROGRAM INSTRUCTIONS
   ...THE DATE IS MISSING OR IS PUNCHED INCORRECTLY***
```

Example 10

```
*** ERROR IN TAPE OPTION INSTRUCTION ...AN INCORRECT CHARACTER WAS USED ***
*** NO TAPE CREATED ON THIS RUN ***
```

Example 11

Each of the Type #2 instruction data cards specifies a specific sort to be output. Of the eight possible sort types, the Master Summary Table, Stage Sort 1, and Stage Sort 2 data cards require a code mnemonic only. The Applicable Document Sort may have any document number as its descriptor code. The remaining four sorts require a code mnemonic and a descriptor code. Section II gives the details of preparing these cards. If a mnemonic code is incorrect, error messages stating that the mnemonic code is misspelled and that the computer is continuing to process the remaining sorts is printed out. Example 12 shows a sample of an incorrect mnemonic code.

```
*** MNEMONIC NAME SUMMAR IS MISSPELLED***
*** CONTINUING TO PROCESS OTHER SORTS ***
```

Example 12

The descriptor code of a sort request can also be in error. When a descriptor code is used that is not valid, see Appendix A for correct codes, the error message in Example 13 is outputed. In this example the fluid media LH₃ was requested. The correct fluid should have been LH₂, therefore the error message was printed.
Example 13

When requesting an Applicable Document Sort, a code mnemonic and a document number must be specified. If the document number used is not associated with any GSE items in the data bank, the message in Example 14 will be printed.

Example 14
SECTION IV. PROGRAM ASPECTS

Program Execution

The GSE Data Management program consists of control cards, a mainline program, subroutines, and procedure tables used for execution by the Univac 1108 system. The following briefly explains each of these items and discusses their function in the program. It will be convenient to refer to Figure 13 showing the general overview of the complete program.

The mainline program takes control over all other routines. The mainline reads in the program instruction data provided by the user on data cards and makes available such information to other routines. The first program instruction data card contains the date of the run and an optional variable that is used to determine whether to generate an output tape with new or updated data. If an invalid date is encountered by the subroutine DATECK an error message will be written and the run terminated. If GSE attribute information is to be added, updated, or deleted from the data base, subroutine UPDATE will be called from the MAINLINE. UPDATE uses four drum files. All information from the current tape is put on FILE 1. If there are any GSE attribute cards in the card deck run-stream, the data from these cards are put on FILE 2. UPDATE then uses FILE 3 as a working file, i.e., all data from FILE 1 and FILE 2, whether it be duplicate or not, are all put on FILE 3. UPDATE then searches FILE 3 for the most current data and puts them on FILE 4. Updated data on FILE 4 are then used for the current run and can be stored on a new user requested output tape.

Up to eight different sets of sort mnemonic codes can appear on individual data cards following the initial data card. The MAINLINE determines if such are valid names and converts them to an integer code. Subroutine MERGE is then called from the MAINLINE and uses this code to determine which sort operation to perform. There are seven different types of sorts and a master summary table available. Each sort thus has a corresponding sort subroutine as contained in Table 1.

Using the Univac Sort/Merge Package feature (see Program Characteristics), each subroutine does the internal sorting logic. Furthermore, each sort subroutine has a peculiar group of routines that handle the printout logic. For example, subroutine PRINT2 handles the pure mechanics of printing the newly sorted GSE data for Stage Sort 2 including use of subroutines DIMMET, WGTMET, and RELALP for calculation of metric units; subroutine
Figure 13. Master program execution flow chart.
<table>
<thead>
<tr>
<th>Sort Subroutine Name</th>
<th>Type of Sort</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASTER</td>
<td>Master Summary Table</td>
</tr>
<tr>
<td>SORT1</td>
<td>Stage Sort 1</td>
</tr>
<tr>
<td>SORT2</td>
<td>Stage Sort 2</td>
</tr>
<tr>
<td>SORT3</td>
<td>Fluid Media Sort</td>
</tr>
<tr>
<td>SORT4</td>
<td>Functional Code Sort</td>
</tr>
<tr>
<td>SORT5</td>
<td>Class Code Sort</td>
</tr>
<tr>
<td>SORT6</td>
<td>Applicable Document Sort</td>
</tr>
<tr>
<td>SORT7</td>
<td>Use Location Sort</td>
</tr>
</tbody>
</table>

HEADS calls the appropriate heading subroutine for use by the respective print routine; subroutine HEADR2 prints the column headings and footnotes on every page of the sort. Subroutine LINCON determines whether to start writing on a new page or continue writing on the current page. This printout would be in presentable tabular listings displaying all GSE items found under Stage Sort 2. In similar fashion, the sequence of calls to PRINT1, HEADS, HEADR1, and LINCON will accomplish the printout logic for Stage Sort 1.

Program Modification Procedures

Each card of the current version of the GSE Data Management Program is catalogued by line entry number. This simplifies the procedure by which any temporary modifications can be made to the program if it is not entirely satisfactory to the program analyst. When making temporary changes, the line entry of each statement can be obtained from the left-hand column number of the current program listing (see Appendix B).

Permanent changes can also be made to the program. Being modular in form, the program is flexible enough to incorporate major additions with
minimum effort. For example, assume that a type of sorting function is desired that is not currently available from the program. Through the use of several control cards, the program analyst can add those subroutines (subprograms) that were designed to provide such extended capability. These additions would then be represented by several new block operations to be fitted on Figure 13. Refer to the discussion in the section on how to change or add Fortran subroutines.

Changes to Program Procedures Tables

Figure 14 is an example sheet from Appendix B that illustrates a typical catalogued listing from one of the program's procedure tables. Notice that each statement has the catalogued computer number positioned to the extreme left of it. These catalogued listing numbers, as shown in Appendix B, will change if any Fortran routines or procedure tables are altered when permanent changes to the program have been made by the user. To temporarily modify any of these statements, the insert card method is used. This requires an insert card that may contain any of the following information:

1. The location number of the statement to be replaced or modified (see Example 15).

2. The location number of the statement to be deleted entirely (see Example 16).

3. The sequential location numbers of the statements to be modified (see Example 17).

4. The location numbers of the statement which modifications are to follow (see Example 18).

All insert cards are punched beginning with a minus sign in card column one (1) and must follow the procedure table control card in the runstream. Example 15 shows the procedure table control card and insert card setup.

---

Example 15
**XSTAGE PROC**

**STAGENAME PROCEDURE TABLE**

**THE ARRAY **GSETLE** CONTAINS THE VARIOUS GSE STAGENAME TITLES**

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSTAGE</td>
<td>THE NUMBER OF STAGENAMES APPEARING IN THIS TABLE + 1</td>
</tr>
</tbody>
</table>

**ROUTES THAT USE XSTAGE PROC**

<table>
<thead>
<tr>
<th>ROUTINES</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MASTER</td>
<td></td>
</tr>
<tr>
<td>HEADR1</td>
<td></td>
</tr>
<tr>
<td>HEADR2</td>
<td></td>
</tr>
<tr>
<td>SORTI</td>
<td></td>
</tr>
<tr>
<td>PRINTI</td>
<td></td>
</tr>
</tbody>
</table>

**INTEGER GSETLE(100)**

**DATA GSETLE**

<table>
<thead>
<tr>
<th>GSETLE</th>
<th>LONGNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-1R</td>
<td>S-1R</td>
</tr>
<tr>
<td>S-1C</td>
<td>S-1C</td>
</tr>
<tr>
<td>S-1V</td>
<td>S-1V</td>
</tr>
<tr>
<td>S-2I</td>
<td>S-2I</td>
</tr>
<tr>
<td>S-2II</td>
<td>S-2II</td>
</tr>
</tbody>
</table>

**INTEGER**

<table>
<thead>
<tr>
<th>INTEGER</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
</tr>
</tbody>
</table>

**DATA**

<table>
<thead>
<tr>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
</tr>
</tbody>
</table>

**Figure 14. Procedure table catalogue listing.**
The statements that were in some way modified are identified as location Nos. 98 and 100. Note that on each insert card the location number is listed twice and there is a one to one replacement on the statement that was 98. It should also be noted that all statements that are placed between the insert card and the succeeding insert card, if any, will replace the statement at location No. 100. Thus it would be possible to insert a finite number of statements at location No. 100 or, as Example 16 illustrates, delete a statement completely if no statement should follow the insert card. However, due to the logic and nature of the particular procedure table, location No. 96 would have to be modified also, so as to reflect the correct number of items that would remain in that procedure table after the deletion.

Example 16

Example 17 shows how the user can modify all statements between and including 99 through 102. If no statements appear after the insert card, the program segment identified on that insert card would be deleted entirely.

Example 17

Example 18 illustrates how any number of statements can be inserted to follow a certain statement in the program. Note that only one location number is required on the insert card following the minus sign.

Example 18
Due to the nature of the procedure tables and the Univac 1108 system, it is necessary to recompile those Fortran subroutines that use a procedure table that has been modified. Note that within the comment statement portion of each procedure table there is a list of subroutine names that use that procedure table. Recompilation requires that these subroutine names must then appear on control cards that follow the end of all insert cards to the procedure tables. Example 19 illustrates the format for such control cards. It is recommended that the "S" option be used and, as the example shows, the subroutine name should appear twice and should be separated by a comma.

Example 19

Changing and Adding Fortran Routines

If it were necessary to modify the actual Fortran statements in the program, the same insert card method is illustrated in the previous example is used. Example 20 illustrates the appearance of the control card needed and an example insert card setup. It is required that the "US" option be used on these control cards, noting also that the subroutine name again appears twice and is separated by a comma. A very important reminder is that these Fortran control cards (@FOR, US) and their associated insert cards must follow the procedure table control card and its associated insert cards, if any, in the card deck runstream.

Example 21 shows that only one control card is needed to add a new subroutine to the program. Simply put the control card before the subroutine deck. Note that the "IS" option is used, followed by a blank space and then the name of the routine. Addition of a new subroutine must follow the procedure table changes, if any, that are being made.
Example 20

Example 21

Deck Setup, Control Cards, and Options

Following the procedure table and Fortran subroutine control cards and their associated insert cards, a final control card is required to allow for these program modifications. Example 22 shows what is called the @ MAP control card. It is used to collect all routines from the current tape and make a new temporary program with the user's specified program modifications.

Example 22
This card must appear immediately after all subroutines and before the @ XQT PROVE card. The program analyst also has the option to make a new tape that would contain any program modifications, as was discussed earlier. Simply punch "T" for tape at card column 11 of the Type 1 card. It is recommended that a trial run be made first without the "T" option so as to validate any program modifications. When such modifications are working correctly, utilize the "T" option. Figure 15 illustrates how the entire deck setup would look with sample program modifications in the runstream.

Figure 15. Control card deck with program modifications.
Program Characteristics

The GSE Data Management program has several special features, one of which is the use of the Univac Sort/Merge Package. It consists of a collection of Fortran-callable subroutines which allow the program to interface with the 1108 System Sort Program. Subroutines UPDATE, SORT1 through SORT7 use this Sort Package. For more details about this package a complete write-up is available from the publishing organization.

Another feature of this program is the use of the Fortran V procedure table. It allows storage of any array information that is used as common reference among the 28 routines. Each routine can therefore use whatever procedure table it needs in order to gain access to desired specifications. Most modifications the program analyst may wish to make to the procedure table do not necessitate changes to those routines that use that table. Remember that the program is also designed in such a way that it can handle programming additions. The program presently uses 23 000 words of core with a maximum of 32 000 words allowable for remote operations. The data bank currently allows for a total of 540 GSE-EI entries but can be enlarged to fit the need for more, merely by changing the specifications on the four @ASG, TN, F2 control cards. Computer execution time per sort ranges from roughly 13 seconds for a relatively smaller output such as the Master Summary Table, to 26 seconds for one of the more typical sorts (Stage Sort1).

Description of the Program Listing

Each routine of the GSE Data Management Program is commented in a standard format so that changes or additions can be easily made by experienced programmers. As can be seen in Appendix B, the current program listing is given for each routine. Appearing first within each routine are comment statements that briefly describe its function to the program. Following this initial description are comment statements which identify and define the important variables in the subroutine. Also note that among the actual Fortran program statements are added comments that give better detail to the significant operations of the subprogram. Special attention should be given to the information provided in the comment portion of each procedure table. Conveniently located there is a description of its function, any definitions of important variables, and a cross reference to those routines that use the table.
APPENDIX A. DEFINITION OF INPUT FORMAT FOR GSE ATTRIBUTE DATA CARDS
APPENDIX A. DEFINITION OF INPUT FORMAT FOR GSE ATTRIBUTE DATA CARDS

Input to the GSE information bank is supplied by data punched in sets of four or more 80-column computer cards. The format for these cards is shown in Figure A-1. Figure A-2 shows an example of all four card types. Care should be taken when using the various Descriptor Codes explained in the following sections. These descriptor Codes are the variables used to determine the specific information which will be placed in four of the eight possible sorts. The Descriptor Codes used must contain no blank spaces within each code and must contain no zeros. (NOTE: All information should be punched left justified unless otherwise stated.)

Card Type 1

1. Card Number – This number identifies that this is the first of a set of four or more data cards (card column 1 punch a 1).

2. Project Number – This digit code number represents the general project in which the GSE item is associated. Presently the five codes used are as follows (card column 4):

<table>
<thead>
<tr>
<th>Code</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Saturn</td>
</tr>
<tr>
<td>2</td>
<td>Skylab</td>
</tr>
<tr>
<td>3</td>
<td>Shuttle</td>
</tr>
<tr>
<td>4</td>
<td>HEAO¹</td>
</tr>
<tr>
<td>5</td>
<td>Spacelab</td>
</tr>
</tbody>
</table>

3. EI Number – This 13-maximum-digit number should be left justified (card columns 7-19).

¹. High Energy Astronomy Observatory.
Figure A-1. GSE input format sheet.
Figure A-2. Example of GSE filled-in data sheet.
4. Title — This information space provides the first available line for the writing of a brief description for GSE associated with the above EI number. Care should be taken in order that this line of description does not utilize more than 36 spaces. Characters should be punched in columns 22 through 57 inclusive, since the computer program will not accept characters outside this limit.

5. Continuation of Card Type 1 — In order to continue the description or remarks, a new card must be punched with a dollar sign in column 1 and, on that same card, the continuation of the title description should be punched in columns 22 through 57 inclusive. Take note not to exceed the designed card space.

6. Each continuation card is identical. The maximum number of continuation cards is 19 (i.e., a total of 20 description cards is allowable including the continuations).

Card Type 2

1. Card Number — Punch a number 2 in column 1.

2. Design Responsibility — This is the organization which is directly responsible for the actual design of an item of GSE. This organization may or may not be the manufacturer. The code is left justified in columns 4 through 15 inclusive (examples: Rocketdyne, ASTN-EPG, MDAC-WD).

3. Class Code — This integer, punched in column 18, represents the major hardware category that is supported by a particular piece of GSE. The code must be one of the following:

<table>
<thead>
<tr>
<th>Descriptor Code</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stage Systems</td>
</tr>
<tr>
<td>2</td>
<td>Payload Module</td>
</tr>
<tr>
<td>3</td>
<td>Experiment</td>
</tr>
</tbody>
</table>

4. Quantity — This is the number of pieces of GSE that fit this particular description. This integer should be placed left justified in columns 21 and 22.
5. Weight In Pounds — The weight is a left justified, six-digit-
maximum integer in columns 25 through 30. If weight is not known, use TBD; right justify entry.

6. Stage — This attribute defines the stage or vehicle assembly that the GSE supports. The present program will key off an alphanumeric or mnemonic code. All mnemonics for these stages are left justified, six-or-less-character words that must conform exactly to one of the following:

<table>
<thead>
<tr>
<th>Code</th>
<th>Module or Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>Airlock Module</td>
</tr>
<tr>
<td>ATM</td>
<td>Apollo Telescope Mount</td>
</tr>
<tr>
<td>ET</td>
<td>Shuttle External Tank</td>
</tr>
<tr>
<td>IU</td>
<td>Instrument Unit</td>
</tr>
<tr>
<td>MDA</td>
<td>Multiple Docking Adapter</td>
</tr>
<tr>
<td>OWS</td>
<td>Orbital Workshop</td>
</tr>
<tr>
<td>SRB</td>
<td>Shuttle Solid Rocket Booster</td>
</tr>
<tr>
<td>SSME</td>
<td>Space Shuttle Main Engine</td>
</tr>
<tr>
<td>S-1C</td>
<td>S-IC Stage</td>
</tr>
<tr>
<td>S-2</td>
<td>S-II Stage</td>
</tr>
<tr>
<td>S-IVB</td>
<td>S-IVB Stage</td>
</tr>
</tbody>
</table>

Other codes can be added to the system based on programming instructions explained in Section IV.

Card Type 3

1. Card Number — Keypunch a 3 in column 1.

2. Use Location — This describes the area of usability for the particular GSE item. Multiple entries of Descriptor Codes are permitted with each
code separated by a comma. The Descriptor Codes should be centered using columns 4 through 21 and must be as follows:

<table>
<thead>
<tr>
<th>Descriptor Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP</td>
<td>Canoga Park</td>
</tr>
<tr>
<td>HB</td>
<td>MDAC-WD, Huntington Beach</td>
</tr>
<tr>
<td>IBM</td>
<td>IBM Space Division</td>
</tr>
<tr>
<td>KSC</td>
<td>Kennedy Center</td>
</tr>
<tr>
<td>MAF</td>
<td>Michoud Assembly Facility</td>
</tr>
<tr>
<td>MFG</td>
<td>Manufacturing Facility</td>
</tr>
<tr>
<td>MSC</td>
<td>Houston Center</td>
</tr>
<tr>
<td>MSFC</td>
<td>Marshall Center</td>
</tr>
<tr>
<td>MTF</td>
<td>Mississippi Test Facility</td>
</tr>
<tr>
<td>SDF</td>
<td>System Development Facility</td>
</tr>
<tr>
<td>STF</td>
<td>Static Test Facility</td>
</tr>
<tr>
<td>WTR</td>
<td>Western Test Range</td>
</tr>
<tr>
<td>FACT</td>
<td>Factory</td>
</tr>
</tbody>
</table>

Other location codes can be added to the program based on instructions given in Section IV.

3. Functional Code — This code describes the basic function that the GSE will perform for the associated piece of hardware. The Descriptor Code must be in columns 24 through 29 and must be one of the following:

<table>
<thead>
<tr>
<th>Descriptor Code</th>
<th>Function Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESS</td>
<td>Access</td>
</tr>
<tr>
<td>GASDET</td>
<td>Gas Detector</td>
</tr>
<tr>
<td>HANDLE</td>
<td>Handling</td>
</tr>
</tbody>
</table>
4. Length in inches is placed in columns 22 through 25. If length is not known, use TBD; right justify entry.

5. Width in inches is placed in columns 28 through 41. If width is not known, use TBD; right justify entry.

6. Height in inches is placed in columns 44 through 47. If height is not known, use TBD; right justify entry.

7. Unit Cost — The cost of design, development, and procurement of one GSE piece of gear and year of purchase is given (example: 320K 71). Center the value in the 13 characters provided in columns 50 through 62. If cost is not known, enter TBD in columns 54 through 56.

8. Media — The type of fluid medium that is used with the GSE is given (examples: GN2, RP-1, He). Multiple entries of fluid media Descriptor Codes are allowed with each code separated by a comma. Use columns 65 through 76 and center the entry(s) in the columns provided. The Descriptor Codes are as follows:

<table>
<thead>
<tr>
<th>Descriptor Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COOL</td>
<td>Coolanol</td>
</tr>
<tr>
<td>ETHGLY</td>
<td>Ethylene Glycol</td>
</tr>
<tr>
<td>FREON</td>
<td>Freon</td>
</tr>
<tr>
<td>GH2</td>
<td>Gaseous Hydrogen</td>
</tr>
</tbody>
</table>

2. Note blank space in code.
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GN2</td>
<td>Gaseous Nitrogen</td>
</tr>
<tr>
<td>GO2</td>
<td>Gaseous Oxygen</td>
</tr>
<tr>
<td>HE</td>
<td>Helium</td>
</tr>
<tr>
<td>HYD</td>
<td>Hydraulic Fluid</td>
</tr>
<tr>
<td>H2O</td>
<td>Water</td>
</tr>
<tr>
<td>H2OGLY</td>
<td>Water Glycol</td>
</tr>
<tr>
<td>ISOALC</td>
<td>Isopropyl Alcohol</td>
</tr>
<tr>
<td>LH2</td>
<td>Liquid Hydrogen</td>
</tr>
<tr>
<td>LN2</td>
<td>Liquid Nitrogen</td>
</tr>
<tr>
<td>ME/H2O</td>
<td>Methanol Water</td>
</tr>
<tr>
<td>MMH</td>
<td>Monomethyl Hydrazine</td>
</tr>
<tr>
<td>N2O4</td>
<td>Nitrogen Tetraoxide</td>
</tr>
<tr>
<td>ORONIT</td>
<td>Oronite</td>
</tr>
<tr>
<td>PRES</td>
<td>Preservative Oil</td>
</tr>
<tr>
<td>RJ-1</td>
<td>RJ-1</td>
</tr>
<tr>
<td>RP-1</td>
<td>RP-1</td>
</tr>
<tr>
<td>TRIC</td>
<td>Trichloroethylene</td>
</tr>
<tr>
<td>VAC</td>
<td>Vacuum</td>
</tr>
</tbody>
</table>

Make sure that the letter O is punched and not the digit 0. Descriptor Codes contain no zeroes (0).

If no length, width, height, cost, and/or fluid media are associated with the GSE item, punch N/A in the assigned columns.
Card Type 4

1. Card Number — Punch a 4 in column 1.

2. Document Number — This card provides room for up to six document numbers, each number being up to 12 characters long. If a document number is less than 12 characters, leave the remaining spaces blank. Starting in column 4 on the same card, punch the first document number, ending in column 15. Skip a space, that is leave column 16 blank. If a second document exists, use columns 17 through 28. Skip a space and then a third document number. Continue this trend for the remainder of the card, if needed, always noting to skip a space between document numbers.

3. Document Continuation Card — If more than 6 documents are needed, a second document card is required. Only one continuation card is allowed. Punch an asterisk in column 1 and pattern the document numbers exactly as stated in item 2 above.
APPENDIX B.  GSE DATA MANAGEMENT SYSTEM
PROGRAM LISTING
SUBROUTINE DATECK

C **********SUBROUTINE DATECK(NERROR)******************************
C **THIS ROUTINE CHECKS TO SEE IF A VALID NUMERIC DATE HAS BEEN
C **SUPPLIED BY THE PROGRAM ANALYST
C ****DEFINITION OF FUNCTION FLD(I,K,E) I=STARTING BIT K=BIT WIDTH
C Definition of word name
C STORE THE 1ST CHARACTER OF DATX(I) IN VARIABLE NPART1
C NPART1=FLD(0,6,WORD)
C STORE THE 3RD CHARACTER OF DATX(I) IN VARIABLE NPART3
C NPART3=FLD(12,6,WORD)
C STORE THE 2ND CHARACTER OF DATX(I) IN VARIABLE NPART2
C NPART2=FLD(6,6,WORD)
C CHECK TO SEE IF THE 1ST CHARACTER IS A BLANK
C IF(NPART1.EQ.5)GO TO 20
C CHECK TO SEE IF THE 1ST CHARACTER IS A VALID INTEGER
C IF(NPART1.GT.0.AND.NPART1.LT.10)GO TO 30
C CHECK TO SEE IF THE 2ND CHARACTER IS A SLASH(/)
C IF(NPART2.EQ.5)GO TO 35
C CHECK TO SEE IF THE 2ND CHARACTER IS A BLANK
C IF(NPART2.EQ.6)GO TO 35
C CHECK TO SEE IF THE 2ND CHARACTER IS A VALID INTEGER
C IF(NPART2.GT.0.AND.NPART2.LT.10)GO TO 45
C CHECK TO SEE IF THE 3RD CHARACTER IS A BLANK
C IF(NPART3.NE.5)GO TO 35
C SET ERROR KEY
C NERROR=1
C RETURN

C END
SUBROUTINE DIMMET

C ********************************************************************
C **OOO**O*****O******OOO**
C ***********e
C oo e*oo*oo*O*MMT
C 
C*************************************************************************

DIMENSION IM(4)

Cwoord is a library function to obtain bits from alphanumeric numbers

C------------------------------------------------------------------------

DIMETIC,INA

DO 10 IM=2,4
   IADD=IADD+2
   GO TO 10
10 CONTINUE

IMETIC=IMETIC*2.54

CALL RELAP(IMETIC,IMETIC)

RETURN

END
SUBROUTINE HEADRI

1 C ********************************************************************** HDR1: 1
2 C * SUBROUTINE HEADRI - THIS ROUTINE PRINTS THE COLUMN HEADINGS AND HDR1: 2
3 C * THE CLASS CODE FOOTNOTES THAT ARE PECULIAR TO STAGE SORT 1 HDR1: 3
4 C ********************************************************************** HDR1: 4
5 C
6 C             variable                definition HDR1: 5
7 C ***          ****                   HDR1: 6
8 C NKLY         DETERMINES WHEN THE CLASS CODE HDR1: 7
9 C FOOTNOTE IS TO BE WRITTEN HDR1: 8
10 C INPUT       HDR1: 9
11 C END         DETERMINES WHEN TO WRITE BOTH FOOTNOTES HDR1: 10
12 C             ON THE FINAL PAGE OF THE SORT HDR1: 11
13 C            HDR1: 12
14 C JKEY        SET EQUAL TO "*FALSE*" FOR HDR1: 13
15 C             STAGE SORT 2 HDR1: 14
16 C
17 C INCLUDE(DA=T=SHORT) HDR1: 15
18 C * SUBROUTINE HEADRI HDR1: 16
19 C INCLUDE XSTAGELIST HDR1: 17
20 C INCLUDE COMUN HDR1: 18
21 C INTEGER P HDR1: 19
22 C LOGICAL JKEY HDR1: 20
23 C COMMON/XKEY/KKEY HDR1: 21
24 C COMMON/TRUE/END,JKEY,PKEY HDR1: 22
25 P=6 HDR1: 23
26 I IF(IEND-1)25b,2 HDR1: 24
27 2 WRITE(P,264) HDR1: 25
28 JKEY=.FALSE. HDR1: 26
29 5 GO TO 27 HDR1: 27
31 C
32 C
33 C 25 NPAGE=NPALE+1 HDR1: 29
34 C IF(NKEY=1) 90,35,36 HDR1: 30
35 C 35 WRITE(P,264) HDR1: 31
36 C 36 WRITE(P,265) HDR1: 32
37 C NKEY=2 HDR1: 33
38 C 38 GO TO 42 HDR1: 34
39 C 40 NKEY=1 HDR1: 35
40 C 42 WRITE(P,272) HDR1: 36
41 C 43 WRITE(P,26D) HDR1: 37
42 C
43 C WRITE OUT THE DATE AND THE PAGE NUMBER HDR1: 38
44 C
45 C 45 WRITE(P,219)DATA,PROGM,NPAGE HDR1: 39
46 C
47 C DETERMINE WHICH MODULE(STAGE) TITLE IS TO BE PRINTED HDR1: 40
48 C SUBSCRIPTED VARIABLE "GSETLE" CONTAINS ALL GSE TITLE HEADINGS HDR1: 41
49 C
50 C 60 J=MOD+4 HDR1: 42
51 C 65 I=J+2 HDR1: 43
52 C 70 WRITE(P,230)GSETLE(K),K=1,J HDR1: 44
53 C

53
SUBROUTINE HEADR1 (Concluded)

54 C*****PRINT THE GENERAL COLUMN HEADINGS
55 C
56 75 WRITE(P,226)
57 80 WRITE(P,230)
58 83 WRITE(P,235)
59 84 WRITE(P,240)
60 215 FORMAT (8x, A6, A2, 3X, 2A6, 1X, 'PROJECT', 4X, 'PAGE', 1X, 12)
61 220 FORMAT (8x, 57x, 3A6)
62 225 FORMAT (1X, 15x(1H*))
63 230 FORMAT (AUX, 5X CLASS, 3X *FUNCTIONAL, 19X, 3X 'USE', 17X, 10X 'APPLICABLE')
64 235 FORMAT (3X, 9HEI NUMBER, 2UX, 11X DESCRIPTION, 18X, 9H CODE, 6X, 9H CODE, 20X)
65 * LOCATION, 15X, 9X DOCUMENTS)
66 240 FORMAT (1X, 15x(1H*))
67 260 FORMAT (1X, 'N/A' OR 'NA' = NOT APPLICABLE)
68 265 FORMAT (1X, 'CLASS CODES: 1=STAGE SYSTEMS 2=PAYLOAD MODULE 3=EXPERIMENT')
69 270 FORMAT (1H*, 5X STAGE SORT ONE TABLE*)
70 275 RETURN
END
**SUBROUTINE HEADR2**

This routine prints the stage sort 2 column headers at the top of each page.

```fortran
C ********************************************************************
C * SUBROUTINE HEADR2 - THIS ROUTINE PRINTS THE STAGE SORT 2 COLUMN *
C * HEADINGS AT THE TOP OF EACH PAGE                               *
C ********************************************************************
C
C VARIABLE DEFINITION                                               *
C *KEY*                                                             *
C DETERMINES WHEN THE "NOT APPLICABLE" FOOTNOTE IS TO BE WRITTEN.
C *END*                                                             *
C DETERMINES WHEN TO WRITE THE LAST FOOTNOTE ON THE FINAL PAGE OF
C THE SORT.                                                        *
C JKEY SET EQUAL TO "TRUE" FOR PERFORMING STAGE SORT.               *
C IEND DETERMINES WHEN TO WRITE THE LAST FOOTNOTE ON THE FINAL PAGE
C OF THE SORT.                                                    *
C JKEY SET EQUAL TO "FALSE" FOR PERFORMING STAGE SORT.              *
C P*66*                                                             *
C INCLUDE (DATA=SHORT)                                             *
C SUBROUTINE HEADR2                                                *
C INCLUDE STAGE, LIST                                              *
C INCLUDE COMMON, PREKEY                                           *
C COMMON/IKEY, JKEY, PKEY                                           *
C P = 6                                                             *
C IF (IEND = 1)                                                    *
C WRITE(P, 24)                                                     *
C KEY = "FALSE"                                                    *
C WRITE(P, 25)                                                     *
C PAGE = PAGE + 1                                                  *
C IF (PKEY = 1)                                                    *
C WRITE(P, 26)                                                     *
C WRITE(P, 27)                                                     *
C GO TO 42                                                         *
C PKEY = 1                                                         *
C WRITE(P, 28)                                                     *
C WRITE(P, 29)                                                     *
C WRITE(P, 30)                                                     *
C WRITE(P, 31)                                                     *
C WRITE(P, 32)                                                     *
C WRITE(P, 33)                                                     *
C WRITE(P, 34)                                                     *
C WRITE(P, 35)                                                     *
C WRITE(P, 36)                                                     *
C WRITE(P, 37)                                                     *
C WRITE(P, 38)                                                     *
C WRITE(P, 39)                                                     *
C WRITE(P, 40)                                                     *
C WRITE(P, 41)                                                     *
C WRITE(P, 42)                                                     *
C WRITE(P, 43)                                                     *
C WRITE(P, 44)                                                     *
C WRITE(P, 45)                                                     *
C WRITE(P, 46)                                                     *
C WRITE(P, 47)                                                     *
C WRITE(P, 48)                                                     *
C WRITE(P, 49)                                                     *
C WRITE(P, 50)                                                     *
C WRITE(P, 51)                                                     *
C WRITE(P, 52)                                                     *
C WRITE(P, 53)                                                     *
C WRITE(P, 54)                                                     *
C WRITE(P, 55)                                                     *
C WRITE(P, 56)                                                     *
C WRITE(P, 57)                                                     *
C WRITE(P, 58)                                                     *
C WRITE(P, 59)                                                     *
C WRITE(P, 60)                                                     *
C WRITE(P, 61)                                                     *
C WRITE(P, 62)                                                     *
C RETURN                                                            *
C END                                                               *
```
C subroutine HEADR3

**SUBROUTINE HEADR3** - The function of this routine is to print the column headings at the top of each page of the fluid media sort.

VARIABLE

NAME: HDR3

DEFINITION

HDR3: Subscript of entry in fluid media mnemonic code table

**COMPILER (DATA SHORT)**

**SUBROUTINE HEADR3**

**INCLUDE XMETA8D LIST**

**COMMON THREE, MESAVE**

**NPAGE = NPAGEx + 1**

**WRITE (6,100)**

100 FORMAT (1H1,59), "MEDIA SORT TABLE";

**WRITE (6,150)** (LATX, NPAGE)

150 FORMAT (4X, A6, A2, 46X, "ALL PROGRAMS", 47X, "PAGE", 1X, 12)

**JJ = MESAVE + 9**

**KK = MESAVE + 6**

**WRITE (6, 200)**, TABMEDIM (MM), MM = JJ, KK)

200 FORMAT (59X, 346)

**MULT (6, 250)**

250 FORMAT (1X, I3, (1HM))

**WRITE (6, 280)**

280 FORMAT (168X, "FUNCTIONAL", 16X, "ENVELOPE DIM", 14X, "UNIT", 6X, "DESIGN")

**WRITE (6, 270)**

270 FORMAT (168X, "NUMBER", 16X, "DESCRIPTION", 29X, "CODE", 7X, "WT")

**WRITE (6, 260)**

260 FORMAT (168X, "QTY", 16X, "COST/YR", "RESPONSIBILITY")

**WRITE (6, 250)**

300 FORMAT (168X, "UNIT", 6X, "QTY", 16X, "COST/YR", 2X, "RESPONSIBILITY")

**RETURN**
SUBROUTINE HEADR4

C *** **** *** ** ** ***
C-----
C THE FUNCTION OF THIS ROUTINE IS TO PRINT THE
C COLUMN HEADINGS AT THE TOP OF EACH PAGE OF THE FUNCTIONAL CODE SORT.
C---------
C VARIABLE DEFINITION
C---------
C I5AVL SUBSCRIPT OF ENTRY IN FUNCTIONAL CODE
C MNEMONIC TABLE
C
C COMPILER(UATAwSHURT)
C INCLUDE COMMON
C INCLUDE FUNTABLIST
C COMMON/FUUR/ISAVE
C NPAG = NPAG + 1
C WRITE(6,100)
C FORMAT(INH15.4X,'FUNCTIONAL SORT TABLE')
C WRITE(6,150);ATX,NPAGE
C FORMAT(8X,A5,A2,A43X,'ALL PROGRAMS',97X,'PAGE',1X,I2)
C JJ = ISAVL + 1
C KK = ISAVL + 4
C WRITE(6,ZU0);FUCOTA(MM),MM = JJ,KK)
C FORMAT(5X250)
C WRITE(6,250)
C FORMAT(1X,13);(1M1)
C WRITE(6,260)
C FORMAT(8BJX,'ENVELOPE DIM',14X,'UNIT','6X,'DESIGN')
C WRITE(6,270)
C FORMAT(3X,E1 NUMBER',18X,'DESCRIPTION',20X,'MEDIA',7X,'RT'
C 7X,'6X,'N',6X,'H',3X,'QTY',6X,'COST/YR',2X,'RESPONSIBILITY')
C WRITE(6,250)
C RETURN
C END
SUBROUTINE HEADR5

<table>
<thead>
<tr>
<th>line</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C ******<strong><strong><strong><strong>e</strong>e</strong></strong></strong> HDRS 1</td>
</tr>
<tr>
<td>2</td>
<td>C ** SUBROUTINE HEADR5 - THE FUNCTION OF THIS ROUTINE IS TO PRINT THE HDRS 2</td>
</tr>
<tr>
<td>3</td>
<td>C ** COLUMN HEADINGS AT THE TOP OF EACH PAGE OF THE CLASS CODE SORT HDRS 3</td>
</tr>
<tr>
<td>4</td>
<td>C ******<strong><strong><strong><strong>e</strong>e</strong></strong></strong> HDRS 4</td>
</tr>
<tr>
<td>5</td>
<td>C ***<em>Oo eaosseo e</em>Oo *****<em>OOeoOe</em> ******HORS HDRS 5</td>
</tr>
<tr>
<td>6</td>
<td>C VARIABLE DEFINITION HDRS 6</td>
</tr>
<tr>
<td>7</td>
<td>C ************ HDRS 7</td>
</tr>
<tr>
<td>8</td>
<td>C IHAVIT SUBSCRIPT OF ENTRY IN CLASS CODE MNEMONIC HDRS 8</td>
</tr>
<tr>
<td>9</td>
<td>C TABLE HDRS 9</td>
</tr>
<tr>
<td>10</td>
<td>C HDRS 10</td>
</tr>
<tr>
<td>11</td>
<td>C HDRS 11</td>
</tr>
<tr>
<td>12</td>
<td>INCLUDE COMMON HDRS 12</td>
</tr>
<tr>
<td>13</td>
<td>INCLUDE CLASSJ LIST HDRS 13</td>
</tr>
<tr>
<td>14</td>
<td>COMMON/FIVE/IHAVIT HDRS 14</td>
</tr>
<tr>
<td>15</td>
<td>NPAGE = NPAGE + 1 HDRS 15</td>
</tr>
<tr>
<td>16</td>
<td>WRITE(6,100) HDRS 16</td>
</tr>
<tr>
<td>17</td>
<td>100 FORMAT(15I6) HDRS 17</td>
</tr>
<tr>
<td>18</td>
<td>WRITE(6,150) DATA,NPAGE HDRS 18</td>
</tr>
<tr>
<td>19</td>
<td>150 FORMAT(16X,A6,A2,2X,9X,<em>ALL PROGRAMS</em>,47X,<em>PAGE</em>,I2) HDRS 19</td>
</tr>
<tr>
<td>20</td>
<td>JJ = IHAVIT + 1 HDRS 20</td>
</tr>
<tr>
<td>21</td>
<td>KK = IHAVIT + 3 HDRS 21</td>
</tr>
<tr>
<td>22</td>
<td>WRITE(6,200) (CLASSCODE(MM), MM = JJ,KK) HDRS 22</td>
</tr>
<tr>
<td>23</td>
<td>200 FORMAT(6,250) HDRS 23</td>
</tr>
<tr>
<td>24</td>
<td>WRITE(6,250) HDRS 24</td>
</tr>
<tr>
<td>25</td>
<td>250 FORMAT(1X,13I1) HDRS 25</td>
</tr>
<tr>
<td>26</td>
<td>WRITE(6,260) HDRS 26</td>
</tr>
<tr>
<td>27</td>
<td>260 FORMAT(6X,<em>FUNCTION</em>,43X,<em>DESIGN</em>) HDRS 27</td>
</tr>
<tr>
<td>28</td>
<td>WRITE(6,270) HDRS 28</td>
</tr>
<tr>
<td>29</td>
<td>270 FORMAT(3X,<em>NUMBER</em>,20X,<em>DESCRIPTION</em>,23X,<em>CODE</em>,9X,<em>STAGE</em>,I1X, HDRS 29</td>
</tr>
<tr>
<td>30</td>
<td>3I,NAME*,9X,<em>TYPE</em>,5X,<em>RESPONSIBILITY</em>) HDRS 30</td>
</tr>
<tr>
<td>31</td>
<td>WRITE(6,280) HDRS 31</td>
</tr>
<tr>
<td>32</td>
<td>WRITE(6,280) HDRS 32</td>
</tr>
<tr>
<td>33</td>
<td>RETURN HDRS 33</td>
</tr>
<tr>
<td>34</td>
<td>END HDRS 34</td>
</tr>
<tr>
<td>35</td>
<td>HDRS 35</td>
</tr>
</tbody>
</table>
SUBROUTINE HEADR6

C *************** SUBROUTINE TO PRINT HEADING FOR APPLICABLE DOCUMENT SORT TABLE ***************
C

C VARIABLE DEFINITION

C

C **COMPILE(UNIT=SHORT)
C SUBROUTINE HEADR6
C INCLUDE COMUIN,
C INCLUDE CODEST
C COMM/U/SIX/ IFHR6
C IF(IFHR6=1) THEN 11
C 10 CONTINUE
C IFHR6 = IFHR6 + 1
C 12 CONTINUE
C NPAGE = NPAGE + 1
C
C****PRINT MAIN HEADING AT TOP OF EACH PAGE
C WRITE(6,800)
C 800 FORMAT(1X,'APPLICABLE DOCUMENT SORT TABLE')
C WRITE(6,811) DATA,DESC01(NUMBER),DESC02(NUMBER),NPAGE
C 801 FORMAT(1X,8A,2X,3B,4X,DOCUMENT ',2A,42X,PAGE ',12)
C WRITE(6,805)
C 805 FORMAT(59X,'ALL PROGRAMS')
C WRITE(6,802)
C 802 FORMAT(13A2,3X,'DOCUMENTS')
C WRITE(6,803)
C 803 FORMAT(478A,B,LASS,9X,'FUNCTIONAL',19X,'APPLICABLE')
C WRITE(6,804)
C 804 FORMAT(3JX,'NUMBER',28X,'DESCRIPTION',28X,'CODE',12X)
C WRITE(6,801)
C 801 FORMAT(1X,8A,2X,3B,4X,DOCUMENT ',2A,42X,PAGE ',12)
C WRITE(6,805)
C 805 FORMAT(59X,'ALL PROGRAMS')
C WRITE(6,802)
C 802 FORMAT(13A2,3X,'DOCUMENTS')
C WRITE(6,803)
C 803 FORMAT(478A,B,LASS,9X,'FUNCTIONAL',19X,'APPLICABLE')
C WRITE(6,804)
C 804 FORMAT(3JX,'NUMBER',28X,'DESCRIPTION',28X,'CODE',12X)
C RETURN TO SUBROUTINE HEADS
C RETURN
C CONTINUE
C****PRINT FOOTNOTE AT BOTTOM OF EACH PAGE
C WRITE(6,801)
C 801 FORMAT(1X,8A,2X,3B,4X,DOCUMENT ',2A,42X,PAGE ',12)
C WRITE(6,805)
C 805 FORMAT(59X,'ALL PROGRAMS')
C WRITE(6,802)
C 802 FORMAT(13A2,3X,'DOCUMENTS')
C WRITE(6,803)
C 803 FORMAT(478A,B,LASS,9X,'FUNCTIONAL',19X,'APPLICABLE')
C WRITE(6,804)
C 804 FORMAT(3JX,'NUMBER',28X,'DESCRIPTION',28X,'CODE',12X)
C ***EXPERIMENT***
C GO TO 12
C END
SUBROUTINE HEADR7

<table>
<thead>
<tr>
<th>Line</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

COMMENTS:
- The function of this routine is to print the column headings at the top of each page of the use location sort.

VARIABLE DEFINITION:
- KESAVE: Subscript of entry in use location mnemonic.

CODE TABLE:
- INCLUDE COMMON/SVEN/KESAVE
- NPAGE = NPAGE + 1
- WRITE(6,100)
- WRITE(6,150) DATA,NPAGE
- WRITE(6,200) FORMAT(1X,5X,USE LOCATION SORT TABLE)
- WRITE(6,250) FORMAT(1X,13X('**FUNCTIONAL'),15X,'ENVELOPE DIM',24X,'*DESIGN*)
- WRITE(6,270) FORMAT(1X,'NUMBER',18X,'DESCRIPTION',20X,'CODE',7X,'WT',6X,'MEDIA',8X,'RESPONSIBILITY')
- WRITE(6,290) WRITE(6,290) WRITE(6,290) WRITE(6,290) WRITE(6,290)
- RETURN
- END
SUBROUTINE HEADS

C SUBROUTINE HEADS CALLS THE APPROPRIATE HEADING SUBROUTINE FOR
C USE BY THE RESPECTIVE PRINT ROUTINE
C
C******************************************************************************
C
0 INCLUDE COUNTER LIST
8 CONTINUE
10 CALL HEAD1
16 GO TO 80
18 CONTINUE
24 CALL HEAD2
26 GO TO 80
28 CONTINUE
32 CALL HEAD3
34 GO TO 80
36 CONTINUE
40 CALL HEAD4
42 GO TO 80
44 CONTINUE
48 CALL HEAD5
50 GO TO 80
52 CONTINUE
56 CALL HEAD6
58 GO TO 80
60 CONTINUE
62 CALL HEAD7
64 GO TO 80
66 CONTINUE
68 RETURN
70 END
SUBROUTINE LINCON

C ********************************************************************
C * SUBROUTINE LINCON - THIS ROUTINE DETERMINES WHETHER TO START A NEW PAGE OR CONTINUE WRITING ON THE CURRENT PAGE
C ********************************************************************
C
C VARIABLE DEFINITION

C

LAOD IS THE LINE COUNTER USED BY ALL ROUTINES

C

COMPILER DATA (SHORT) INCLUDE COMMON

C

C****DETERMINE IF THIS IS THE TOP OF A NEW PAGE

C

80 IF (LAOD=1) 90,90,100

90 CONTINUE

LAOD = 1

RETURN

100 CONTINUE

ITEST = NFLINES

LAOD = LAOD + ITEST

C

C****DETERMINE IF THERE IS ENOUGH SPACE ON THE CURRENT PAGE TO COMPLETE PRINTOUT

C

1 IF (LAOD-49) 10,20,20

10 CONTINUE

RETURN

20 CONTINUE

LAOD = 0

C

C*****WRITE THE FOOTNOTE AT THE BOTTOM OF THE PAGE AND PRINT THE NEW PAGE HEADING

C

200 WRITE(*,201)

201 FORMAT (/IX=N/A OR NA = NOT APPLICABLE)

250 CALL HEADS

LAOD = 10

260 RETURN

END
MAIN PROGRAM

C MAIN - THIS IS THE MAINLINE TO THE GSE DATA MANAGEMENT PROGRAM
C
C FUNCTIONS OF THE MAINLINE:
C
1. READ IN INFORMATION PROVIDED BY THE USER ON THE PROGRAM
2. DETERMINE WHETHER TO UPDATE THE MASTER DATA FILE
3. PROCESS, SORT, AND MERGE ALL GSE INFORMATION BY MEANS OF ITS SUPPORT ROUTINES
4. DETERMINE WHETHER OR NOT TO GENERATE A NEW TAPE

INCLUDE MAIN,LIST
INCLUDE COMMON,LIST
INCLUDE CUSTLIST
INTEGER READWRITE
DIMENSION CIMAGE(4)
DATA CIMAGE/*WADD CONTROL-FILE+
READ*
WRITE=6
MARG=0

C READ IN THE PROGRAM INSTRUCTION VARIABLES
READ READ,41 DATA,ANSWER
FORMAT(A6,A2,2X,A1)
C DETERMINE IF THE DATE VARIABLE IS VALID
CALL DATECK(NERROR)
IF(NERROR.EQ.1)GO TO 73
CONTINUE

C READ IN THE SORT MNEMONICS CODES
DO 7 I=1,MAXSRT
READ(READ,6)ERR=73,END=10)CDESRT(I),DESCOOD(I),DESCO2(I)
FORMAT(A6,4X,2A6)
NUMSRT=I
7 CONTINUE
10 CONTINUE

C UPDATE THE MASTER DATA FILE
CALL UPDATE

C CONVERT THE REQUESTED SORT MNEMONICS TO INTEGER CODES
C
DO 40 6=1,14,1

63
MAIN PROGRAM (Concluded)

54   25 DO 40 J=1,9
55   30 IF ICIDESRT(I),EQ,XMOD(J),ICIDESRT(I)=J
56   35 IF ICIDESRT(I),EQ,J GO TO 60
57   40 CONTINUE
58   45 WRITE(10,50) ICIDESRT(I)
59   50 FORMAT(*5X,16AH8MNEMONIC NAME, A6,17H IS MISSPELLED*) MAIN 59
60   55 WRITE(10,56)
61   56 FORMAT(*5X,1X,CONTINUING TO PROCESS OTHER SORTS ***) MAIN 61
62   57 ICIDESRT(I)=10 MAIN 62
63   60 CONTINUE

C   C****PERFORM SORTS

64   C MAIN 64
65   65 CALL MENGL MAIN 65
66   66 C MAIN 66
67   67 IF(NTEST.IEQ.5) GO TO 75 MAIN 67
68   68 WRITE(6,60)
69   69 FORMAT(*5X,CONTINUING TO PROCESS OTHER SORTS ***) MAIN 69
70   70 CALL CSFHEI(CIMAGE) MAIN 70
71   71 WRITE(6,71)
72   72 FORMAT(*5X,CHARACTER WAS NOT ALPHABETICALLY CORRECT***) MAIN 72
73   73 WRITE(6,73)
74   74 FORMAT(*5X,NO TAPE CREATED ON THIS RUN ***) MAIN 74
75   75 STOP MAIN 75
76   76 C MAIN 76
77   77 IF(NTEST.IEQ.26) GO TO 70 MAIN 77
78   78 WRITE(6,78)
79   79 FORMAT(*5X,ERROR IN TAPE OPTION INSTRUCTION ***) MAIN 79
80   80 C MAIN 80
81   81 IF(NTEST.IEQ.26) GO TO 70 MAIN 81
82   82 WRITE(6,72)
83   83 FORMAT(*5X,ERROR IN PROGRAM INSTRUCTIONS ***) MAIN 83
84   84 WRITE(6,74)
85   85 FORMAT(*5X,THE DATE IS MISSPELLED***) MAIN 85
86   86 FORMAT(*5X,ERROR IN USER MESSAGE***) MAIN 86
87   87 WRITE(6,76)
88   88 IF(NTEST.IEQ.26) GO TO 70 MAIN 88
89   89 IF(NTEST.IEQ.26) GO TO 70 MAIN 89
90   90 WRITE(6,78)
91   91 FORMAT(*5X,ERROR IN PROGRAM INSTRUCTIONS ***) MAIN 91
92   92 75 STOP MAIN 92
93   93 80 END MAIN 93
**SUBROUTINE MASTER**

```fortran
C******************************************************************************MSTR 1  
C ** MASTER - THIS ROUTINE PRINTS OUT A SUMMARY OF ALL GSE ENTITIES **MSTR 2  
C ** CONTAINED IN THE MASTER DATA FILE **MSTR 3  
C******************************************************************************MSTR 4  
C
C INCLUDE COMPUN MSTR 5  
C INCLUDE ENTITY MSTR 6  
C INCLUDE PROGRAM MSTR 7  
C INCLUDE STAGE MSTR 8  
C REMIND 4 MSTR 9  
C READ(4,END=99)WORDS MSTR 10  
C WRITE(6,100) MSTR 11  
C FORMAT(1H1) MSTR 12  
C NCHECK=0 MSTR 13  
C WRITE(6,105) MSTR 14  
C FORMAT(1H1) MSTR 15  
C WRITE(6,101) MSTR 16  
C WRITE(6,102) MSTR 17  
C FORMAT(5X,ENTITY # NUMBER PROGRAM STAGE REFERENCE DATE) MSTR 18  
C WRITE(6,102) MSTR 19  
C READ(4,END=99)WORDS MSTR 20  
C GOTO 9 MSTR 21  
C CONTINUE MSTR 22  
C********************************************************READ THE MASTER DATA FILE**********************************************************MSTR 23  
C READ(9,END=99)WORDS MSTR 24  
C NCHECK=NCHECK+1 MSTR 25  
C********************************************************DETERMINE IF A VALID PROGRAM CODE EXISTS**********************************************************MSTR 26  
C IF(INPRO.GT.5.OR.INPRO.LE.0)INPRO=6 MSTR 27  
C WRITE(6,106) MSTR 28  
C IF(INPRO.EQ.6)WRITE(6,106) MSTR 29  
C LENPRO=*2 MSTR 30  
C M=1-1 MSTR 31  
C DO 10 I=1,NSTAGE MSTR 32  
C J=J+1 MSTR 33  
C IF(MOD(J,I)=1)GO TO 11 MSTR 34  
C IF(MOD(J,II)=1)GO TO 11 MSTR 35  
C CONTINUE MSTR 36  
C********************************************************DETERMINE WHAT STAGE NAME IS TO BE PRINTED OUT**********************************************************MSTR 37  
C IF(MOD(J,E)MOD=1 MSTR 38  
C WRITE(6,109) MSTR 39  
C IF(MOD(J,E)MOD=1)GO TO 11 MSTR 40  
C CONTINUE MSTR 41  
C********************************************************ERROR***STAGE NAME COULD NOT BE DETERMINED**********************************************************MSTR 42  
C IF(NCHECK.NEQ.1)WRITE(6,104) MSTR 43  
C WRITE(6,104) MSTR 44  
C********************************************************ERROR***PROGRAM CODE VALUE IS INVALID**********************************************************MSTR 45  
C IF(NCHECK.NEQ.1)WRITE(6,106) MSTR 46  
C WRITE(6,106) MSTR 47  
C********************************************************END**********************************************************MSTR 48  
```

65
SUBROUTINE MERGE

1 SUBROUTINE MERGE - THIS ROUTINE DETERMINES WHICH SORT LEG TO
2 PERFORM. THE ARRAY CALLED **CODESRT** REPRESENTS THE FOLLOWING
3
4 INTEGER CODES:

5 1 stage sort 1
6 2 stage sort 2
7 3 fluid media sort
8 4 functional code sort
9 5 class code sort
10 6 applicable document sort
11 7 use location sort
12 8 master summary table

13 SUBROUTINE MERGE
14 INCLUDE CODESLIST
15 DO 60 I=1,NUMSRT
16 NUMSRT=NUMSRT+2
17 IF (CODESRT(I).LE.0 .OR. CODESRT(I).GT.NUMSRT) GO TO 68
18 NARGU=CODESRT(I)
19 GO TO (16,18,24,32,40,48,56,64,72,80,NARGU)
20 CALL SORT1
21 WRITE(*,10)
22 GO TO 60
23 CALL SORT2
24 WRITE(*,10)
25 GO TO 60
26 CALL SORT3
27 WRITE(*,10)
28 GO TO 60
29 CALL SORT4
30 WRITE(*,10)
31 GO TO 60
32 CALL SORT5
33 WRITE(*,10)
34 GO TO 60
35 CALL SORT6
36 WRITE(*,10)
37 GO TO 60
38 CALL SORT7
39 WRITE(*,10)
40 GO TO 60
41 CALL SORT8
42 WRITE(*,10)
43 GO TO 60
44 CALL SORT9
45 WRITE(*,10)
46 GO TO 60
47 CALL MASTER
48 WRITE(6,10)
49 GO TO 60
50 WRITE(6,10)
51 CONTINUE
52 WRITE(6,10)
53 WRITE(6,10)
54 FORMAT(*ERROR** CURRENT SORT COULD NOT BE PERFORMED DUE TO AN INCORRECT INTEGER SORT CODE IN SUBROUTINE MERGE*)
55 TO AN INCORRECT INTEGER SORT CODE IN SUBROUTINE MERGE*)
56 WRITE(*,10)
57 GO TO 74
58 WRITE(6,10)
59 RETURN
60 END
SUBROUTINE PRINT1

C * SUBROUTINE PRINT1 - THIS ROUTINE PRINTS CERTAIN GSE ATTRIBUTES FOR *
C * THOSE EI NUMBERS TO BE FOUND IN STAGE SORT 1 *
C ******************************************************PRT1
C
C VARIABLE DEFINITION
C ********
C
NKEY TELLS SUBROUTINE HEADRI WHEN TO WRITE THE
C CLASS CODE FOOTNOTE
C
C
C COMMON/ONE/NKEY
C COMMON/ONE/NKEY
C
C COMPILE(WDATA=SHORT)
C SUBROUTINE PRINT1
C INCLUDE ASSTAGLILIST
C INCLUDE COMMON
C INTEGER P,OLD,RULE
C DATA BLANK/1
C NAME
C LINES
C
C C DETERMINE IF THE STAGE NAME HAS CHANGED
C C
C 7 IF(MODE=OLD) GO TO 86
C 27 IF(TITLE=4) GO TO 22
C 10 NKEY=1
C 18 GO TO 24
C 24 CALL LINCON
C
C C DETERMINE WHETHER THE NUMBER OF APPLICABLE DOCUMENTS IS GREATER
C C THAN THE NUMBER OF LINES OF DESCRIPTION
C C
C 86 DU 89,1,1,29,2
C 87 IF(APDOC(1)-BLANK)89,90,88
C 89 JTAB=1
C 90 CONTINUE
C 91 IF(APDOC-N_LINES)<6,92,92
C 92 LDIFF=APDOC-N_LINES
C 94 CALL LINCON
C
C C WRITE THE FIRST LINE OF EQUIPMENT INFORMATION
C C
C 100 WRITE(P,245) (EINUM(I),J=1,3),(TITLE(K),K=1,NX),NCODE,FCTCDE,USLOC
C (APDOC(N),N=1,2)

67
SUBROUTINE PRINT1 (Concluded)

50 C
51 C*****WRITE THE REMAINING LINES OF DESCRIPTION
52 C
53 115 IF(LINES<=4) GO TO 165
54 120 DO 155 LINES = LINES + 1
55 130 BASE = BASE + 1
56 135 RULE = LTYPE + 1
57 136 IF(LTYPE = 45) GO TO 146
58 140 WRITE(P,255)(TITLE(WORDS),WORDS=BASE,RULE)(APDOC(2*LTYPE-1)),(APRTI 58
59 *POC(ZLTYFE)
60 146 WRITE(P,251)(TITLE(WRITE),WRITE=BASE,RULE)
61 150 RULE = RULE
62 155 CONTINUE
63 156 RULE = RULE
64 157 LINES = LINES
65 160 IF(LINES<6) GO TO 186
66 C
67 C*****WRITE THE REMAINING APPLICABLE DOCUMENTS
68 C
69 165 IF(APDOC(2*LINES+1)-BLANK) GO TO 170
70 170 WRITE(P,255)(APDOC(2*LINES+1)))(APDOC(2*LINES+2))
71 175 LINES = LINES + 1
72 177 IF(LINES>12) GO TO 185
73 180 LINES = LINES
74 185 WRITE(P,250)
75 190 LADD = LADD + 1
76 C
77 C*****SAVE THE CURRENT STAGE NAME
78 C
79 200 OLD = MOD
80 245 FORMAT(X,2A4,A16X,A6,A6,5X,A3,13X,A6,14X,18A),18X,2A6)
81 250 FORMAT(Z0X,A6,A6,6X,5A6)
82 251 FORMAT(Z0X,A6)
83 265 FORMAT(15X,6A6)
84 260 FORMAT(1)
85 C
86 C*****RETURN TO SORT 1 ROUTINE
87 C
88 275 RETURN
89 280 END
SUBROUTINE PRINT3

C * SUBROUTINE PRINT3 - THIS ROUTINE PRINTS CERTAIN GSE ATTRIBUTES FOR PRT3
C * THOSE EI NUMBERS TO BE FOUND IN THE FLUID MEDIA SORT
C *

C ******************************************PRINT3

C INCLUDE COMON
CALL WGTMLT(IOGHTMAGHT)
CALL DIMET(ILENGTH,ILNGTH)
CALL DIMET(IWIDTH,IWIDTH)
CALL DIMET(IHEIGHT,IHEIGHT)
CALL LINCON
WRITE(4,LU1)
101 FORMAT(I)
LA0 = LADD + 1
WRITE(100,100) (EINUM(I),J=1,3),(TITLE(K),K=1,NX),FCTCD,EWGT,LENGTH,
100 FORMAT(I,E6,A1,4X,A6,5X,A6,6X,A6,5X,A6,PLB,1X,A4,1INF,1X,A4)
NX = 12
WRITE(LLY,11) GO TO 30
WRITE(LLX,11) GO TO 40

20 IF(NLINES.EQ.4) GO TO 30
WRITE(4,I&6,WGTMAGHT,WIDTM,HEIGHT)
S01 FORMAT(I8,I3,A6,16X,A6,5X,A6,5X,A6,A6,5X,A6,A6,5X,A6,
20 IF(NLINES.EQ.4) GO TO 30
WRITE(4,I&6,WGTMAGHT,WIDTM,HEIGHT)

30 WRITE(6,600,7,2) GO TO 40
WRITE(16,16) TITLE(K),K=1,12,HEIGHT,LENGTH,WIDTM
40 FORMAT(I8,2A6,A1,2X,4A6,4A6,4A6,19
41 RETURN

CONTINUE
50 CONTINUE
50 RETURN

END
SUBROUTINE PRINT4

C **********************************************************************
C * SUBROUTINE PRINT4 - THIS ROUTINE PRINTS CERTAIN GSE ATTRIBUTES FOR *PRT4 *
C * THOSE NUMBERS TO BE FOUND IN THE FUNCTIONAL CODE SORT *PRT4 *
C **********************************************************************

C COMPLETEI(UNITA=SHORT)
C SUBROUTINE PRINT4
C INCLUDE COMMON
C NX = 6
C CALL WGMET(IKGHTM,KGHT)
C CALL DGMET(ILNGTHM,LNGTH)
C CALL DGMET(IMGHTM,IMGHT)
C CALL LGMET(IWIDTM,IWIDTH)
C CALL LGMET(IHGTM,HGT)
C CALL UNGLN
C WHITE(I,101);
C
20 FORMAT(1X)
21 LADD = LADD + 1
22 WHITE(I,101), J = 1,3 TITLE(K),K = 1,NX1, MEDIA,WGMET,LNGTHM,
23 IWIDTHM,HGT, JUAN, UNCST, DESC 
24 100 FORMAT(1X,2A6,A1,4X,6A6,3X,12A1,1X,49,1X,49,49,49,49,49,1X,49,49,49,49,1X,49,49,49,49,1X,49,49,49,49,1X)
25 20 FORMAT(1X,1X,49,1X,49,49,49,49,1X)
26 CALL 1DH(6.0,6.0)
27 CALL DIMMET(IADTM,WIDTHM)
28 CALL DIMMET(IMGHTM,HGT)
29 CALL DIMMET(IWIDTM,IWIDTH)
30 CALL DIMMET(IHGTM,HGT)
31 CALL LINCON
32 WHITE(I,101)
33 30 FORMAT(1X,2A6,A1,4X,6A6,3X,12A1,1X,49,1X,49,49,49,49,1X,49,49,49,49,1X,49,49,49,49,1X,49,49,49,49,1X)
34 40 FORMAT(1X,1X,49,1X,49,49,49,49,1X)
35 CALL CONTINUE
36 RETURN
37 END

C **************************************************************************
SUBROUTINE PRINTS

C ***************************************************************
C * SUBROUTINE PRINTS - THIS ROUTINE PRINTS CERTAIN GSE ATTRIBUTES FOR *PRTS
C * THOSE RT NUMBERS TO BE FOUND IN THE CLASS CODE SORT *PRTS
C ***************************************************************

CONTACT DATA
INCLUDE COPON
NX = 6

CALL LINCON
WHITE(6, 101)

101 FORMAT(1X)

100 FORMAT(1X, 2A, 16X, 6A, 10X, 6A, 6X, 12A, 6X, 2A, 6X, 2A)

10 CONTINUE

150 FORMAT(1X, 6A)

END
SUBROUTINE PRINT6

C ************************************************************prt6
C * subroutine to print out all information associated with the users prt6
C * specified applicable document(s) *prt6
C ************************************************************prt6

C        C*compileridata=short)
C        SUBROUTINE PRINT6
C        INCLUDE C014JN
C        DATA BLANK/#

C        C****CALL subroutine used for line control
C        CALL LINCON
C        C****Determine number of applicable documents associated with ei numberprt6
DO 99 Im=3,1,2
     IF(APDOC(I)=BLANK) 88,90,88
88 JTAB=1
90 CONTINUE
99     IF(NAPDOC=NLINES) 96,92,92
C        C****MAKE CONNECTION IF NUMBER OF APPLICABLE DOCUMENTS IS GREATER-
C        THAN NUMBER OF LINES OF DESCRIPTION
   92 LDIFF = NAPDOC-NLINES
   93 LADD = LADD+LDIFF
C        C****CALL subroutine used for line control
C        CALL LINCON
   96     NX = 6
   97     NXL = I
   98     LADD = LADD + I
C        C****print out information associated with specified
C        APPLICABLE DOCUMENT
   10 WRITE(6,900) (TITLE(K),K=NXL,NX)
   11 FORMAT(25A6)

   90 FORMAT(I10,2A6,A1,14X,6A6,14X,13X,A6,14X,2A6)
   91 10 N=N+1,LINESL=I
   92 WRITE(6,901) (TITLE(K),K=NXL,NX)
   93 FORMAT(25A6)
   94 901 CONTINUE
   95 10 CONTINUE
   96 WRITE(6,904)
   97 16
SUBROUTINE PRINT6 (Concluded)

54  904 FORMAT (1)
55     LADD = LADD + 1
56
57     C RETURN TO SUBROUTINE SORT6
58     C
59
60     RETURN
61  60  61  IF(LINESL-116.I,7)  PRINT6  59
62     6 = L+1
63     WRITE(4,903)(APDOC(I),J=1,L)
64  903 FORMAT (116X,2A6)
65     LADD = LADD + 1
66     1 = 1+2
67
68  70 TO 15  PRINT6  66
69  LINESL = LINESL-1
70  NX = NX+6
71
72  WRITE (6,902) (TITLE(I),K=NXL,NI,APDOC(J),J=1,L)  PRINT6  71
73  902 FORMAT (28X,6A6,52X,2AA)
74
75  END
76
SUBROUTINE PRINT7

C ************ SUBROUTINE PRINT7 - THIS ROUTINE PRINTS CERTAIN GSE ATTRIBUTES FOR *PRT7
C THOSE ON NUMERALS TO BE FOUND IN THE USE LOCATION SORT
C
C ************

COMPILED(in11&m SHORT)
INCLUDE COMMON
NX = 6
CALL NGMTL(NMTH,NGH)
CALL DIMLT(LENGTH,LENGTH)
CALL DIMLT(IWTH,WIDTH)
CALL DIMLT(IHGH,HEIGHT)
CALL LINCON
WRITE(8,101)
101 FORMAT(1X)
LAUD = LAUO + 1
WRITE(0,101) (EINUM(J),J=1,3), (TITLE(K),K=1,NX), (FCTCOE,NGTH,LENGTH),
IAMTH,NGTH,MA,DESCN
100 FORMAT(1X,2A6,A4,X,6A6,X,A5X,A6,9LB4,I14,A4,*IN1,,I14,A4,
13N,IX,A4,*IN1,,I2A1,IX,,2A6)
12N
N=12
LAUD = LAUD + 1
12N
IF(JUMN+0.1) GO TO 30
IF(JUMN+0.2) GO TO 40
WHITE(0,SU),TITLE(K),K=1,12), INGTH, INGMT, INGMHT
26 SUM FORMATT(1X,2A6,,14X,A4,A6,KG*,IX,A4,*CH*,IX,A4,*CH*)
27 LAUD = LAUO + 1
28 GO TO 20
29 WHITE(0,SU), INGTH, INGMT, INGMHT
30 SUM FORMATT7IX,6A6,KG*,IX,A4,*CH*,IX,A4,*CH*)
31 LAUD = LAUO + 1
32 RETURN
33 WHITE(0,SU), TITLE(J), J=1,12), INGTH, INGMT, INGMHT
34 GO TO 31
35 CONTINUE
36 WHITE(0,SU), TITLE(JM,JM = ILMW, INGMHT
40 150 FORMAT(1X,2A6)
LAUD = LAUO + 1
NX = INGMHT
CONTINUE
RETURN
END

75
SUBROUTINE RELALP

C ********************************************************************** R LLP 1
2 C SUBROUTINE RELALP - THIS SUBROUTINE CONVERTS REAL NUMBERS TO R LLP 2
3 C ALPHANUMERIC NUMBERS R LLP 3
4 C ********************************************************************** R LLP 4
5 C ********************************************************************** R LLP 5
6 SUBROUTINE RELALP(IREK,WFAL,K) R LLP 6
7 DIMENSION IX(6) R LLP 7
8 I=REAL R LLP 8
9 DO 1 J=1,K R LLP 9
10 C R LLP 10
11 C*****MOD IS A LIBRARY FUNCTION TO GIVE RESIDUAL NUMBER FROM DIVISION R LLP 11
12 C R LLP 12
13 IX(J)=MOD(I*10) R LLP 13
14 I=I/10 R LLP 14
15 CONTINUE R LLP 15
16 IRES=I R LLP 16
17 ICHK=0 R LLP 17
18 GO TO 2 J=K,1,-1 R LLP 18
19 IF(ICHK*NE.0.OR.J.EQ.1)GO TO 3 R LLP 19
20 IF(I(X(J)*NE.0) GO TO 4 R LLP 20
21 IX(J)=5 R LLP 21
22 GO TO 5 R LLP 22
23 ICHK=1 R LLP 23
24 I=I*X(J)+48 R LLP 24
25 IRES=IRES*64+IX(J) R LLP 25
26 CONTINUE R LLP 26
27 IF(K.EQ.6) GO TO 10 R LLP 27
28 IRES=(IRES*64+51*064+R LLP 28
29 RETURN R LLP 29
30 END R LLP 30
SUBROUTINE SORT1

C SUBROUTINE SORT1 - THIS ROUTINE DOES ALL THE SORTING OF THE GSE TO
C BE FOUND IN STAGE SORT 1
C
C NOTE: THIS ROUTINE, WHEN CALLED FROM SUBROUTINE SORT2, PERFORMS
C THE SAME SORT ON GSE TO BE FOUND IN STAGE SORT 2
C
C ********************************************************************

C VARIABLE DEFINITION

C ********

C JKLY IF THIS LOGICAL VARIABLE IS SET EQUAL TO
C TRUE, **+ SUBROUTINE PRINT1 IS CALLED
C
C IF ** FALSE ** SUBROUTINE PRINT1 IS CALLED

C ********

C JEND DETERMINES WHEN THE PRINT ROUTINES ARE TO
C WRITE BOTH FOOTNOTES ON THE LAST PAGE OF
C THE SORT

C******************************************************

C COMPILER = SHORT
C SUBROUTINE SORT1
C INCLUDE STAGE, LIST
C INCLUDE PROGRAM LIST
C INCLUDE COMMON
C INCLUDE LCIV
C LOGICAL JKLY COMMON/ TAG/EJN, KEY, PREY

C******/DEFINE THE RECORD FILE SIZE, SPECIFY KEY FIELDS, AND INITIALIZE
C******/THE SORT/MERGE PACKAGE
C
C CALL KEY(1,5136, 'B', 'A', 1)
C CALL KEY(2,35136, 'A', 'A', 2)
C CALL KEY(2,35136, 'A', 'A', 3)
C CALL SRTOPN
C HEAND 4
C LIEND 0
C NPAG0 0
C LAU0 0
C MAU1 4, LRH=2, END=81, WORDS
C
C C******RELEASE THE MOKUS TO BE SORTED ON THE RECORD FILE
C
C CALL SRHET(WORDS)
C GO TO 75
C
C C******START SORT OF DATA FILE
C
C CALL SRTHHT
C CONTINUE
C
C******************************************************
SUBROUTINE SORT1 (Concluded)

54 **THE SORTED WORDS FROM THE RECORD FILE**
55 C
56 CALL SRTRNL(WORDS,89)
57 GO TO 1
58 KEEP=(1*4)=3
59 IF(MOD.LW+55)تصرف(KEEP)MOD=1
60 IF(MOD.LW=11) GO TO 11
61 IF(continues)
62 WRITE(6,100)MOD,LISTNUM
63 IFSTAGEHM=10X,****ERROR****THE PROCEDURE TABLE HAS BEEN SEARCHED ASRTI 63
64 IFSTAGE=NAME;01X,A4,A2X,ET,NO;01X,A4,A1,2A,TCOULD NOT BE FOUND ASRTI 64
65 2)
66 GO TO 87
67 CONTINUE
68 IF(NPRO.LE.0,OR.NPRO.GE.e)
69 WRITE(69,120)
70 IF(NPRO.LE.0,OR.NPRO.GE.e) NPRO=SRTI 69
71 IF(NPRO.LE.0,OR.NPRO.GE.e) NPRO=SRTI 70
72 JUMP=NPR0=SRTI 71
73 INDEX=JUMP=SRTI 72
74 PROG(1)=AGENDU(INDEX)
75 PROG(2)=AGENDU(JUMP)
76 IF(JKEY) GO TO 90
77 CALL PRINT1
78 GO TO 87
79 CALL PRINT2
80 GO TO 87
81 IF(JKEY) GO TO 95
82 IEND=1
83 CALL HEAD3
84 WRITE(6,110)
85 **COMPLETION OF STAGE SORT 1 ****
86 GO TO 99
87 IEND=1
88 CALL HEAD3
89 WRITE(6,111)
90 IEND=1
91 CALL HEAD2
92 WRITE(6,109) ****ERROR**** ERROR WHILE READING DATA BASE FILE RTI 93
93 RETURN
94 END
SUBROUTINE SORT2

C**********************************************************************SRT2
2 C * SUBROUTINE SORT2 - THIS ROUTINE SETS THE VARIABLE KEY FOR PERFORMING STAGE 2 SRT2
3 C * ING STAGE SORT 2 SRT2
4 C**********************************************************************SRT2
5 C SRT2 5
6 C***************************************************************SRT2 6
7 C***************************************************************SRT2 7
8 C***************************************************************SRT2 8
9 C***************************************************************SRT2 9
10 C***************************************************************SRT2 10
11 C***************************************************************SRT2 11
12 C***************************************************************SRT2 12
13 C***************************************************************SRT2 13
14 C***************************************************************SRT2 14
15 C***************************************************************SRT2 15
16 C***************************************************************SRT2 16
17 C***************************************************************SRT2 17

C SUBROUTINE SORTZ

CALL SORT1
RETURN
END
SUBROUTINE SORT3

C *************************************************************************************************SRT3
C * SUBROUTINE SORT3 - THIS ROUTINE DOES ALL THE SORTING OF THE GSE TO SRT3
C * BE FOUND IN THE FLUID MEDIA SORT
C *************************************************************************************************SRT3

COMPILED DATA=SHORT)
SUBROUTINE SORT3
INCLUDE COULST
INCLUDE COMON
INCLUDE EVQIV
INCLUDE XMETA8LIST
COMMON/THREATLIST/...-
LADD
NPAGE = 0
NSCTO = 0
CMED = DESCODINGNUMBER)
DO I = 1, MECODS
MESAVE = (I*7)-6
IF (CMED*E(TABMEDIMESAVE)) GO TO 30
CONTINUE
WHITE(I,IOUI) CMED
100 FORMAT(I/IOA,**DESCRIPTOR NAME(2X,A6,3X*IN ERROR***)
RETURN
30 CONTINUE
LET1 = TABMEDIMESAVE + I)
LET2 = TABMEDIMESAVE + 2)
LET3 = TABMEDIMESAVE + 3)
CALL R52M1(WORDS)
CALL KEYM1A5108A1AP01)
CALL SRTREP
REWIND 4
READ(X,Y=50) WORDS
CALL SRTREI(WORDS)
GO TO 40
50 CALL SRTSRT
52 CALL SRTRET(WORDS,ST0)
C
C****INITIALIZE
C
I = 0
1 CONTINUE
I = I + 1
C
C****CHECK REMAINING FIELD SIZE
C
2 CONTINUE
IF (I.EQ.12) GO TO 52
IF (LETLET1=MEDIA(I)) 1,4,11
4 CONTINUE
I = I + 1
IF (I.EQ.12) GO TO 52
C
C****CHECK NEXT LETTER
C
CONTINUE
SUBROUTINE SORT3 (Concluded)

54 IF (LET2-MEDIA(1)) .GE. 2.11.2
55 11 CONTINUE
56 I=I+1
57 IF (I.EQ.12) GO TO 52
58 Continue
59 C
60****CHECK THIRD LETTER
61 C
62 IF (LET3-MEDIA(1)) .GE. 2.17.2
63 12 Continue
64 NSET=1
65 CALL PRINT3
66 GO TO 52
67 C
68 C
69 C****HAVE SEARCHED DATA THE SORT IS NOW COMPLETE
70 C
71 70 CONTINUE
72 IF (NSET.EQ.1) WRITE(6,38)CHMED
73 38 FORMAT(1X,'***MASTER FILE CONTAINED NO DATA WITH *AA* FLUID
74 C
75 RETURN
76 37 FORMAT(1X,'*** COMPLETION OF MEDIA SORT ***)
77 RETURN
78 END
SUBROUTINE SORT4

C ***************************************************************************
C * SUBROUTINE SORT4 - THIS ROUTINE DOES ALL THE SORTING OF THE GSE TO *SRT4
C * BE FOUND IN THE FUNCTIONAL CODE SORT
C ***************************************************************************
C
C **COMPILE(UIN_FILE,SHORT)**
C SUBROUTINE SORT4
C INCLUDE CMNU
C INCLUDE EQU
C INCLUDE CODEST
C INCLUDE FNTABLIST
C COMMON/FOUR/ISAVE
C MPAGE = 0
C LDDU = 0
C
NSET = 0
FUNK = DESCOD (NUMBER)
DO 1 1 = 1, NCODS
ISAVE = (1 + 5) - 1
IF (FUNK < FC + FC (ISAVE)) GO TO 30
1 CONTINUE
WRITE (6,16) FUNK
100 FORMAT (10X, "***** DESCRIPTION NAME1,2X,A6,2X,F10.10 IN ERROR*****")
RETURN

30 CONTINUE
CALL RS2 (1, WORDS)
CALL KEY1 (23, 35, 36, A1, A1)
CALL KEY2 (2, 35, 16, A1, A1)
CALL SRT1PN
K = 1
40 HEAD (4, END) = 50) ROMDS
CALL SRT2RL (WORDS)
GO TO 40
50 CALL SRT4RT
CALL SRT4T (WORDS, 570)
IF (FUNK < EV + FC (DE)) GO TO 60
GO TO 52
60 CONTINUE
NSET = 1
CALL PRINT4
GO TO 52
70 CONTINUE
IF (NSET = = 0) WRITE (6,79) FUNK
79 FORMAT (10X, "***** MASTER FILE CONTAINED NO DATA WITH A6,F10.10 FUNCTION SORT*****")
CALL CODE SORT
WRITE (6, 80)
80 FORMAT (10X, "***** COMPLETION OF FUNCTIONAL CODE SORT *****")
RETURN
END
SUBROUTINE SORT5

C ************************************************************SRT5 1
C * SUBROUTINE SORTS - THIS ROUTINE DOES ALL THE SORTING OF THE GSE TO SRTS 2
C * BE FOUND IN THE CLASS CODE SORT SRTS 3
C ************************************************************SRT5 4

C (COMMON (CLASS SHORT))
C SUBROUTINE SORTS 5
C INCLUDE COMMON 6
C INCLUDE EQUIV 7
C INCLUDE COSEX 8
C INCLUDE CLASSOD LIST 9
C COMMON/FLFVE/IHAVIT 10
C LADO = 0 11
NPAU = 0 12
NSET = U 13
I0 IN 1 = 1, JUCUDS 14
IHAVIT = (1#4) 15
IF(DESCUD(INL.EMBER).EQ.CLASCO(IHAVIT)) GO TO 30 16
CONTINUE 17
IOUT 18
IUU FORMAT(//ILA,**DESCRIPTOR NAME',2X,A6,2X,**IN ERROR***) 19
RETURN 20
CONTINUE 21
CALL SRTF1, SRTS 22
CALL SRTSRT SRTS 23
GO TO 40 24
CALL SRTSRTS 25
CALL SRTRLI(WORDS) 26
GO TO 52 27
CALL SRTLIL(WORDS) 28
CALL SRTF1(WORDS) 29
GO TO 30 30
CALL SRTF1(WORDS) 31
GO TO 40 32
CALL SRTF1(WORDS) 33
IF (ICODE .LT. CODE) GO TO 36 34
GO TO 52 35
CONTINUE 36
NSET = U 37
CALL PRIN5 38
GO TO 52 39
CONTINUE 40
IF (NSET.EQ.0) WRITE(6,38) CODE 41
FORMATA//ILA,**MASTER FILE CONTAINED NO DATA WITH CLASS CODE',125) 42
I****7 43
WRITE(6,37) 44
FORMATA//ILA,**COMPLETION OF CLASS CODE SORT ****) 45
RETURN 46
END 47
C SUBROUTINE TO FIND THE GSE INFORMATION ASSOCIATED WITH THE APPLICABLE DOCUMENT

C VARIABLE DEFINITION

IERR - DETERMINES IF THE SPECIFIED APPLICABLE DOCUMENT WAS FOUND

C COMPILER DATA=SHORT
C SUBROUTINE SORT6
C INCLUDE COMMON
C INCLUDE EQUIV
C INCLUDE CODEST
C COMMON/ISR/ IFHR6
C DATA BLANK/ *
C LAUD = 0
C IERR = 0
C IFHR6 = 0

C IERR IS VARIABLE WHICH DETERMINES IF SPECIFIED APPLICABLE DOCUMENT WAS FOUND
C REMIND 4
C READ TAPE DATA FROM FILE 4
C READ14,END=981WORDS

C DETERMINE WHICH GSE EQUIPMENT IS ASSOCIATED WITH SPECIFIED APPLICABLE DOCUMENT

I = 1
I = I + 2
GO TO 9
J = J + 1
IF (APDOC(I) = DESCOD(NUMBER)) 13, 14, 15
CONTINUE
I = J + 1
GO TO 9
C CALL SUBROUTINE TO PRINT INFORMATION ASSOCIATED WITH SPECIFIED APPLICABLE DOCUMENT
C CALL PRINT6
IERR = IERR + 1
GO TO 1
SUBROUTINE SORT6 (Concluded)

98 CONTINUE
C
C****IF IERR VALUE IS ZERO, APPLICABLE DOCUMENT WAS NOT FOUND
C
C****IF IERR VALUE IS GREATER THAN ZERO, APPLICABLE DOCUMENT WAS FOUND
C
IF (IERR-1) .GE. 21 .OR. 21 .GE. 21
C
C****PRINT ERROR - DOCUMENT SPECIFIED COULD NOT BE FOUND
C
WRITE(6,6U1) DESCODINUMBER, DESCOZNUMBER
6U1 FORMAT('I10, I10, I10, I10', IERRO**DOCUMENT NOT FOUND**)

C
C*****PRINT FOOTNOTES FOR LAST PAGE
C
WRITE(6,6U2)
6U2 FORMAT('I10, I10, I10, I10, I10, I10', I1ST. I1ST. I1ST. I1ST. I1ST. I1ST.)

C
C****PRINT MESSAGE THAT DOCUMENT SORT IS COMPLETED
C
WRITE(6,6U1) DESCODINUMBER, DESCOZNUMBER
6U1 FORMAT('I10, I10, I10, I10, I10', I1ST. I1ST. I1ST. I1ST. I1ST. I1ST.)

C
C*****RETURN TO SUBROUTINE MERGE
C
RETURN
END
SUBROUTINE SORT7

C **************************************************
C * SUBROUTINE SORT7 - THIS ROUTINE DOES ALL THE SORTING OF THE GSE TO **
C * BE FOUND IN THE USE LOCATION SORT
C **************************************************

COMPLK(DATA=SHORT)

C ******************************************************
SUBROUTINE SORT7
C ******************************************************

INCLUDE COMMON

INCLUDE EQUIV

INCLUDE CUREST

INCLUDE USLOC+LIST

COMMON/SEVEN/KESAVE

LOAD = 0

NPAGE = 0

ILOC = 0

CHLOC = DESCOD(INNUMBER)

UD TO 1 = I, LOCOD

IF (CHLOC =+ PLTAB(KESAVE)) GO TO 30

10 CONTINUE

WRITE (1,100) CHLOC

100 FORMAT (10X, '***DESCRIPTOR NAME I'TX, A6, ZX, A4 IN ERROR***)

RETURN

30 CONTINUE

CALL H52M(WORDS)

CALL KEYM(135+108, 'A&P', 1)

CALL SRTPN

REMKN 4

40 HEA(N;END+50)WORDS

CALL SRTRL(WORDS)

UD TO 40

50 CALL SRTSN

52 CALL SRTRL(WORDS, 57U)

USCK1 = PLTAB(KESAVE)

USCK2 = PLTAB(KESAVE+2)

U0 = 0

1 CONTINUE

1 = 1 + 1

2 CONTINUE

IF (I.EQ.18) GO TO 20

IF (USCK1 = USLOC(I)) 1, 1

4 CONTINUE

1 = 1 + 1

4 CONTINUE

NSLW = 1

CALL PRINT7

GO TO 52

20 CONTINUE

GO TO 52

70 CONTINUE

WRITE (16,36)CHLOC

36 FORMAT (10X, '***MASTER DATA FILE CONTAINED NO DATA WITH 'A6, 'USR777 52

1E LOCATION SORT***)

WRITE (16,37)

37 FORMAT (10X, '*** COMPLETION OF USE LOCATION SORT ***)

RETURN

END
### PROCEDURE TABLE

**PEG001**

**MAIN PROC**

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0002</td>
<td>C **********************************************************TBLE 1</td>
</tr>
<tr>
<td>0003</td>
<td>C * SORT CODE MNEMONIC PROCEDURE TABLE *TBLE 3</td>
</tr>
<tr>
<td>0004</td>
<td>C * THE ARRAY &quot;XMOD&quot; CONTAINS THE STANDARD SORT CODE MNEMONICS *TBLE 4</td>
</tr>
<tr>
<td>0005</td>
<td>C * TYPE OF SORT *TBLE 5</td>
</tr>
<tr>
<td>0006</td>
<td>C * MNEMONIC *TBLE 6</td>
</tr>
<tr>
<td>0007</td>
<td>C * **********************************************************TBLE 7</td>
</tr>
<tr>
<td>0008</td>
<td>C * **********************************************************TBLE 8</td>
</tr>
<tr>
<td>0009</td>
<td>C * **********************************************************TBLE 9</td>
</tr>
<tr>
<td>0010</td>
<td>C * STAGE1 *TBLE 10</td>
</tr>
<tr>
<td>0011</td>
<td>C * STAGE2 *TBLE 11</td>
</tr>
<tr>
<td>0012</td>
<td>C * MEDIA1 *TBLE 12</td>
</tr>
<tr>
<td>0013</td>
<td>C * FUNCTIONAL CODE SORT *TBLE 13</td>
</tr>
<tr>
<td>0014</td>
<td>C * CLASS CODE SORT *TBLE 14</td>
</tr>
<tr>
<td>0015</td>
<td>C * APPLICABLE DOCUMENT SORT *TBLE 15</td>
</tr>
<tr>
<td>0016</td>
<td>C * USE LOCATION SORT *TBLE 16</td>
</tr>
<tr>
<td>0017</td>
<td>C * MASTER SUMMARY TABLE *TBLE 17</td>
</tr>
<tr>
<td>0018</td>
<td>C * TBLE 18</td>
</tr>
<tr>
<td>0019</td>
<td>C * TBLE 19</td>
</tr>
<tr>
<td>0020</td>
<td>C * TBLE 20</td>
</tr>
<tr>
<td>0021</td>
<td>C * TBLE 21</td>
</tr>
<tr>
<td>0022</td>
<td>C * TBLE 22</td>
</tr>
<tr>
<td>0023</td>
<td>C * TBLE 23</td>
</tr>
<tr>
<td>0024</td>
<td>C * TBLE 24</td>
</tr>
<tr>
<td>0025</td>
<td>C * TBLE 25</td>
</tr>
<tr>
<td>0026</td>
<td>C * TBLE 26</td>
</tr>
<tr>
<td>0027</td>
<td>C * TBLE 27</td>
</tr>
<tr>
<td>0028</td>
<td>C * TBLE 28</td>
</tr>
<tr>
<td>0029</td>
<td>C * TBLE 29</td>
</tr>
<tr>
<td>0030</td>
<td>C * TBLE 30</td>
</tr>
<tr>
<td>0031</td>
<td>C * TBLE 31</td>
</tr>
<tr>
<td>0032</td>
<td>C * TBLE 32</td>
</tr>
<tr>
<td>0033</td>
<td>C * TBLE 33</td>
</tr>
<tr>
<td>0034</td>
<td>C * TBLE 34</td>
</tr>
<tr>
<td>0035</td>
<td>C * TBLE 35</td>
</tr>
<tr>
<td>0036</td>
<td>C * TBLE 36</td>
</tr>
<tr>
<td>0037</td>
<td>C * TBLE 37</td>
</tr>
<tr>
<td>0038</td>
<td>C * TBLE 38</td>
</tr>
<tr>
<td>0039</td>
<td>C * TBLE 39</td>
</tr>
<tr>
<td>0040</td>
<td>C * TBLE 40</td>
</tr>
<tr>
<td>0041</td>
<td>C * TBLE 41</td>
</tr>
<tr>
<td>0042</td>
<td>C * TBLE 42</td>
</tr>
<tr>
<td>0043</td>
<td>C * TBLE 43</td>
</tr>
<tr>
<td>0044</td>
<td>C * TBLE 44</td>
</tr>
<tr>
<td>0045</td>
<td>C * TBLE 45</td>
</tr>
<tr>
<td>0046</td>
<td>C * TBLE 46</td>
</tr>
<tr>
<td>0047</td>
<td>C * TBLE 47</td>
</tr>
<tr>
<td>0048</td>
<td>C * TBLE 48</td>
</tr>
<tr>
<td>0049</td>
<td>C * TBLE 49</td>
</tr>
<tr>
<td>0050</td>
<td>C * TBLE 50</td>
</tr>
<tr>
<td>0051</td>
<td>C * TBLE 51</td>
</tr>
<tr>
<td>0052</td>
<td>C * TBLE 52</td>
</tr>
<tr>
<td>0053</td>
<td>C * TBLE 53</td>
</tr>
<tr>
<td>0054</td>
<td>C * TBLE 54</td>
</tr>
<tr>
<td>0055</td>
<td>C * TBLE 55</td>
</tr>
<tr>
<td>0056</td>
<td>C * TBLE 56</td>
</tr>
<tr>
<td>0057</td>
<td>C * TBLE 57</td>
</tr>
<tr>
<td>0058</td>
<td>C * TBLE 58</td>
</tr>
<tr>
<td>0059</td>
<td>C * TBLE 59</td>
</tr>
</tbody>
</table>

### ROUTINES THAT USE MAIN PROC

***************TBLE 20

**MAINLINE**

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0059</td>
<td>** ROUTINES THAT USE CODEST PROC **</td>
</tr>
<tr>
<td>0060</td>
<td>***************TBLE 21</td>
</tr>
</tbody>
</table>

### ROUTINES THAT USE CODEST PROC

***************TBLE 22
<table>
<thead>
<tr>
<th>Procedure</th>
<th>Description</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAINLINE</td>
<td></td>
<td>TABLE 40</td>
</tr>
<tr>
<td>MERGE</td>
<td></td>
<td>TABLE 41</td>
</tr>
<tr>
<td>SRT3</td>
<td></td>
<td>TABLE 42</td>
</tr>
<tr>
<td>SRT4</td>
<td></td>
<td>TABLE 43</td>
</tr>
<tr>
<td>SRT5</td>
<td></td>
<td>TABLE 44</td>
</tr>
<tr>
<td>SRT6</td>
<td></td>
<td>TABLE 45</td>
</tr>
<tr>
<td>SRT7</td>
<td></td>
<td>TABLE 46</td>
</tr>
<tr>
<td>SRT8</td>
<td></td>
<td>TABLE 47</td>
</tr>
<tr>
<td>SRT9</td>
<td></td>
<td>TABLE 48</td>
</tr>
<tr>
<td>SRT10</td>
<td></td>
<td>TABLE 49</td>
</tr>
<tr>
<td>SRT11</td>
<td></td>
<td>Table 50</td>
</tr>
<tr>
<td>SRT12</td>
<td></td>
<td>Table 51</td>
</tr>
<tr>
<td>SRT13</td>
<td></td>
<td>Table 52</td>
</tr>
<tr>
<td>SRT14</td>
<td></td>
<td>Table 53</td>
</tr>
<tr>
<td>SRT15</td>
<td></td>
<td>Table 54</td>
</tr>
<tr>
<td>SRT16</td>
<td></td>
<td>Table 55</td>
</tr>
<tr>
<td>SRT17</td>
<td></td>
<td>Table 56</td>
</tr>
<tr>
<td>SRT18</td>
<td></td>
<td>Table 57</td>
</tr>
<tr>
<td>SRT19</td>
<td></td>
<td>Table 58</td>
</tr>
<tr>
<td>SRT20</td>
<td></td>
<td>Table 59</td>
</tr>
<tr>
<td>SRT21</td>
<td></td>
<td>Table 60</td>
</tr>
<tr>
<td>SRT22</td>
<td></td>
<td>Table 61</td>
</tr>
<tr>
<td>SRT23</td>
<td></td>
<td>Table 62</td>
</tr>
<tr>
<td>SRT24</td>
<td></td>
<td>Table 63</td>
</tr>
<tr>
<td>SRT25</td>
<td></td>
<td>Table 64</td>
</tr>
<tr>
<td>SRT26</td>
<td></td>
<td>Table 65</td>
</tr>
<tr>
<td>SRT27</td>
<td></td>
<td>Table 66</td>
</tr>
<tr>
<td>SRT28</td>
<td></td>
<td>Table 67</td>
</tr>
<tr>
<td>SRT29</td>
<td></td>
<td>Table 68</td>
</tr>
<tr>
<td>SRT30</td>
<td></td>
<td>Table 69</td>
</tr>
<tr>
<td>SRT31</td>
<td></td>
<td>Table 70</td>
</tr>
<tr>
<td>SRT32</td>
<td></td>
<td>Table 71</td>
</tr>
<tr>
<td>SRT33</td>
<td></td>
<td>Table 72</td>
</tr>
<tr>
<td>SRT34</td>
<td></td>
<td>Table 73</td>
</tr>
<tr>
<td>SRT35</td>
<td></td>
<td>Table 74</td>
</tr>
<tr>
<td>SRT36</td>
<td></td>
<td>Table 75</td>
</tr>
<tr>
<td>SRT37</td>
<td></td>
<td>Table 76</td>
</tr>
<tr>
<td>SRT38</td>
<td></td>
<td>Table 77</td>
</tr>
<tr>
<td>SRT39</td>
<td></td>
<td>Table 78</td>
</tr>
<tr>
<td>SRT40</td>
<td></td>
<td>Table 79</td>
</tr>
<tr>
<td>SRT41</td>
<td></td>
<td>Table 80</td>
</tr>
<tr>
<td>SRT42</td>
<td></td>
<td>Table 81</td>
</tr>
<tr>
<td>SRT43</td>
<td></td>
<td>Table 82</td>
</tr>
<tr>
<td>SRT44</td>
<td></td>
<td>Table 83</td>
</tr>
<tr>
<td>SRT45</td>
<td></td>
<td>Table 84</td>
</tr>
<tr>
<td>SRT46</td>
<td></td>
<td>Table 85</td>
</tr>
<tr>
<td>SRT47</td>
<td></td>
<td>Table 86</td>
</tr>
<tr>
<td>SRT48</td>
<td></td>
<td>Table 87</td>
</tr>
<tr>
<td>SRT49</td>
<td></td>
<td>Table 88</td>
</tr>
<tr>
<td>SRT50</td>
<td></td>
<td>Table 89</td>
</tr>
<tr>
<td>SRT51</td>
<td></td>
<td>Table 90</td>
</tr>
<tr>
<td>SRT52</td>
<td></td>
<td>Table 91</td>
</tr>
<tr>
<td>SRT53</td>
<td></td>
<td>Table 92</td>
</tr>
<tr>
<td>SRT54</td>
<td></td>
<td>Table 93</td>
</tr>
<tr>
<td>SRT55</td>
<td></td>
<td>Table 94</td>
</tr>
<tr>
<td>SRT56</td>
<td></td>
<td>Table 95</td>
</tr>
<tr>
<td>SRT57</td>
<td></td>
<td>Table 96</td>
</tr>
<tr>
<td>SRT58</td>
<td></td>
<td>Table 97</td>
</tr>
<tr>
<td>SRT59</td>
<td></td>
<td>Table 98</td>
</tr>
<tr>
<td>SRT60</td>
<td></td>
<td>Table 99</td>
</tr>
<tr>
<td>SRT61</td>
<td></td>
<td>Table 100</td>
</tr>
<tr>
<td>SRT62</td>
<td></td>
<td>Table 101</td>
</tr>
<tr>
<td>SRT63</td>
<td></td>
<td>Table 102</td>
</tr>
<tr>
<td>SRT64</td>
<td></td>
<td>Table 103</td>
</tr>
<tr>
<td>SRT65</td>
<td></td>
<td>Table 104</td>
</tr>
<tr>
<td>SRT66</td>
<td></td>
<td>Table 105</td>
</tr>
<tr>
<td>SRT67</td>
<td></td>
<td>Table 106</td>
</tr>
<tr>
<td>SRT68</td>
<td></td>
<td>Table 107</td>
</tr>
<tr>
<td>SRT69</td>
<td></td>
<td>Table 108</td>
</tr>
<tr>
<td>SRT70</td>
<td></td>
<td>Table 109</td>
</tr>
<tr>
<td>SRT71</td>
<td></td>
<td>Table 110</td>
</tr>
<tr>
<td>SRT72</td>
<td></td>
<td>Table 111</td>
</tr>
<tr>
<td>SRT73</td>
<td></td>
<td>Table 112</td>
</tr>
<tr>
<td>SRT74</td>
<td></td>
<td>Table 113</td>
</tr>
<tr>
<td>SRT75</td>
<td></td>
<td>Table 114</td>
</tr>
<tr>
<td>SRT76</td>
<td></td>
<td>Table 115</td>
</tr>
<tr>
<td>SRT77</td>
<td></td>
<td>Table 116</td>
</tr>
<tr>
<td>SRT78</td>
<td></td>
<td>Table 117</td>
</tr>
<tr>
<td>SRT79</td>
<td></td>
<td>Table 118</td>
</tr>
</tbody>
</table>

**PARAMETER** MAXSRT = 8

**COMMON CODE** CODE(CODSRT(MAXSRT),NARGU,DESD2(MAXSRT),NUMBER,NUMSRT)

**END**
PROCEDURE TABLE (Continued)

PEU119
U120  AMETAB PROC  TABLE 119
U121  C  **********************  TABLE 120
U122  C  *  FLUID MEDIA PROCEDURE TABLE  TABLE 121
U123  C  *  TABLE 122
U124  C  *  THE ARRAY **TABMED** CONTAINS THE VARIOUS FLUID MEDIA CODES  TABLE 123
U125  C  *  TABLE 124
U126  C  *  VARIABLE  TABLE 125
U127  C  *  DEFINITION  TABLE 126
U128  C  *  TABLE 127
U129  C  *  NUMBDS  TABLE 128
U130  C  *  THE NUMBERS OF FLUID MEDIAS APPEARING IN  TABLE 129
U131  C  *  THIS TABLE  TABLE 130
U132  C  *  ROUTINES THAT USE AMETAB PROC  TABLE 131
U133  C  *  TABLE 132
U134  C  *  HEADRN AND SORT4  TABLE 133
U135
U136  C  D MIN D  TABLE(262)  TABLE 134
U137  MECODS  TABLE 135
U138  22  TABLE 136
U139  D ATA(TABMED(K) )  TABLE 137
U140  DATA(TABMED(K) )  TABLE 138
U141  DIMENSION  TABLE 139
U142  TABMED(K)  TABLE 140
U143  K  TABLE 141
U144  DATA(TABMED(K) )  TABLE 142
U145  DATA(TABMED(K) )  TABLE 143
U146  DATA(TABMED(K) )  TABLE 144
U147  DATA(TABMED(K) )  TABLE 145
U148  DATA(TABMED(K) )  TABLE 146
U149  DATA(TABMED(K) )  TABLE 147
U150  DATA(TABMED(K) )  TABLE 148
U151  DATA(TABMED(K) )  TABLE 149
U152  DATA(TABMED(K) )  TABLE 150
U153  DATA(TABMED(K) )  TABLE 151
U154  DATA(TABMED(K) )  TABLE 152
U155  DATA(TABMED(K) )  TABLE 153
U156  DATA(TABMED(K) )  TABLE 154
U157  DATA(TABMED(K) )  TABLE 155
U158  DATA(TABMED(K) )  TABLE 156
U159  DATA(TABMED(K) )  TABLE 157
U160  DATA(TABMED(K) )  TABLE 158
U161  DATA(TABMED(K) )  TABLE 159
U162  DATA(TABMED(K) )  TABLE 160
U163  DATA(TABMED(K) )  TABLE 161
U164  DATA(TABMED(K) )  TABLE 162
U165  DATA(TABMED(K) )  TABLE 163
U166  DATA(TABMED(K) )  TABLE 164
U167  DATA(TABMED(K) )  TABLE 165
U168  DATA(TABMED(K) )  TABLE 166
U169  DATA(TABMED(K) )  TABLE 167
U170  DATA(TABMED(K) )  TABLE 168
U171  DATA(TABMED(K) )  TABLE 169
U172  DATA(TABMED(K) )  TABLE 170
U173  DATA(TABMED(K) )  TABLE 171
U174  DATA(TABMED(K) )  TABLE 172
U175  DATA(TABMED(K) )  TABLE 173
U176  DATA(TABMED(K) )  TABLE 174
U177  DATA(TABMED(K) )  TABLE 175
U178  DATA(TABMED(K) )  TABLE 176
U179  DATA(TABMED(K) )  TABLE 177
U180  DATA(TABMED(K) )  TABLE 178
U181  DATA(TABMED(K) )  TABLE 179

END  TABLE 180

FUNTAB PROC  TABLE 181
U167  C  **********************  TABLE 182
U168  C  *  FUNCTIONAL CODE PROCEDURE TABLE  TABLE 183
U169  C  *  TABLE 184
U170  C  *  THE ARRAY **FUCOTA** CONTAINS THE VARIOUS FUNCTIONAL CODES  TABLE 185
U171  C  *  TABLE 186
U172  C  *  VARIABLE  TABLE 187
U173  C  *  DEFINITION  TABLE 188
U174  C  *  NUMBDS  TABLE 189
U175  C  *  THE NUMBER OF FUNCTIONAL CODES APPEARING IN THIS TABLE  TABLE 190
U176  C  *  ROUTINES THAT USE FUNTAB PROC  TABLE 191
U177  C  *  TABLE 192
U178  C  *  HEADRN AND SORT4  TABLE 193
U179  C  *  TABLE 194
U180  C  **********************  TABLE 195
U181  C  **********************  TABLE 196

END  TABLE 197

FUCOTAIh)  TABLE 198

END  TABLE 199

89
**PROCEDURE TABLE (Continued)**

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U180</td>
<td>U181</td>
<td>DATA FUCOTA, VAC PU, VACUUM PUMP</td>
</tr>
<tr>
<td>U182</td>
<td>U183</td>
<td>REFRIG, REFRIG, REFRIG</td>
</tr>
<tr>
<td>U184</td>
<td>U185</td>
<td>SERVICE, SERVICE, SERVICE</td>
</tr>
<tr>
<td>U186</td>
<td>U187</td>
<td>PNEUMA, PNEUMA, PNEUMA</td>
</tr>
<tr>
<td>U188</td>
<td>U189</td>
<td>TESTCY, TEST CY, TEST CY</td>
</tr>
<tr>
<td>U190</td>
<td>U191</td>
<td>ACCESSY, ACCESSY, ACCESSY</td>
</tr>
<tr>
<td>U192</td>
<td>U193</td>
<td>TRANSPY, TRANSPORTATION</td>
</tr>
<tr>
<td>U194</td>
<td>U195</td>
<td>GASDETY, GAS DETECTOR</td>
</tr>
</tbody>
</table>

---

**USLOCPTO PROC**

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U200</td>
<td>U201</td>
<td>USE LOCATION PROCEDURE TABLE</td>
</tr>
<tr>
<td>U202</td>
<td>U203</td>
<td>THE ARRAY <em>PLTAB</em> CONTAINS THE VARIOUS USE LOCATION NAMES</td>
</tr>
<tr>
<td>U204</td>
<td>U205</td>
<td>VARIABLE DEFINITION</td>
</tr>
<tr>
<td>U206</td>
<td>U207</td>
<td>THE NUMBER OF USE LOCATION CODES APPEARING IN THIS TABLE</td>
</tr>
<tr>
<td>U208</td>
<td>U209</td>
<td>ROUTINES THAT USE USLOCPTO PROC</td>
</tr>
<tr>
<td>U210</td>
<td>U211</td>
<td>HEADRT AND SORTY</td>
</tr>
</tbody>
</table>

---

**USLOCPTO PROC**

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U220</td>
<td>U221</td>
<td>USE LOCATION PROCEDURE TABLE</td>
</tr>
<tr>
<td>U222</td>
<td>U223</td>
<td>THE ARRAY <em>PLTAB</em> CONTAINS THE VARIOUS USE LOCATION NAMES</td>
</tr>
<tr>
<td>U224</td>
<td>U225</td>
<td>VARIABLE DEFINITION</td>
</tr>
<tr>
<td>U226</td>
<td>U227</td>
<td>THE NUMBER OF USE LOCATION CODES APPEARING IN THIS TABLE</td>
</tr>
<tr>
<td>U228</td>
<td>U229</td>
<td>ROUTINES THAT USE USLOCPTO PROC</td>
</tr>
<tr>
<td>U230</td>
<td>U231</td>
<td>HEADRT AND SORTY</td>
</tr>
</tbody>
</table>

---

**USLOCPTO PROC**

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U232</td>
<td>U233</td>
<td>USE LOCATION PROCEDURE TABLE</td>
</tr>
<tr>
<td>U234</td>
<td>U235</td>
<td>THE ARRAY <em>PLTAB</em> CONTAINS THE VARIOUS USE LOCATION NAMES</td>
</tr>
<tr>
<td>U236</td>
<td>U237</td>
<td>VARIABLE DEFINITION</td>
</tr>
<tr>
<td>U238</td>
<td>U239</td>
<td>THE NUMBER OF USE LOCATION CODES APPEARING IN THIS TABLE</td>
</tr>
<tr>
<td>U240</td>
<td>U241</td>
<td>ROUTINES THAT USE USLOCPTO PROC</td>
</tr>
<tr>
<td>U242</td>
<td>U243</td>
<td>HEADRT AND SORTY</td>
</tr>
</tbody>
</table>
### Procedure Table (Continued)

```
PROCEDURE TABLE

<table>
<thead>
<tr>
<th>List Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U240</td>
<td></td>
</tr>
<tr>
<td>U241</td>
<td></td>
</tr>
<tr>
<td>U242</td>
<td></td>
</tr>
<tr>
<td>U243</td>
<td></td>
</tr>
<tr>
<td>U244</td>
<td></td>
</tr>
<tr>
<td>U245</td>
<td></td>
</tr>
<tr>
<td>U246</td>
<td></td>
</tr>
<tr>
<td>U247</td>
<td></td>
</tr>
<tr>
<td>U248</td>
<td></td>
</tr>
<tr>
<td>U249</td>
<td></td>
</tr>
<tr>
<td>U250</td>
<td></td>
</tr>
<tr>
<td>U251</td>
<td></td>
</tr>
<tr>
<td>U252</td>
<td></td>
</tr>
<tr>
<td>U253</td>
<td></td>
</tr>
<tr>
<td>U254</td>
<td></td>
</tr>
<tr>
<td>U255</td>
<td></td>
</tr>
<tr>
<td>U256</td>
<td></td>
</tr>
<tr>
<td>U257</td>
<td></td>
</tr>
<tr>
<td>U258</td>
<td></td>
</tr>
<tr>
<td>U259</td>
<td></td>
</tr>
<tr>
<td>U260</td>
<td></td>
</tr>
<tr>
<td>U261</td>
<td></td>
</tr>
<tr>
<td>U262</td>
<td></td>
</tr>
<tr>
<td>U263</td>
<td></td>
</tr>
<tr>
<td>U264</td>
<td></td>
</tr>
<tr>
<td>U265</td>
<td></td>
</tr>
<tr>
<td>U266</td>
<td></td>
</tr>
<tr>
<td>U267</td>
<td></td>
</tr>
<tr>
<td>U268</td>
<td></td>
</tr>
<tr>
<td>U269</td>
<td></td>
</tr>
<tr>
<td>U270</td>
<td></td>
</tr>
<tr>
<td>U271</td>
<td></td>
</tr>
<tr>
<td>U272</td>
<td></td>
</tr>
<tr>
<td>U273</td>
<td></td>
</tr>
<tr>
<td>U274</td>
<td></td>
</tr>
<tr>
<td>U275</td>
<td></td>
</tr>
<tr>
<td>U276</td>
<td></td>
</tr>
<tr>
<td>U277</td>
<td></td>
</tr>
<tr>
<td>U278</td>
<td></td>
</tr>
<tr>
<td>U279</td>
<td></td>
</tr>
<tr>
<td>U280</td>
<td></td>
</tr>
<tr>
<td>U281</td>
<td></td>
</tr>
<tr>
<td>U282</td>
<td></td>
</tr>
<tr>
<td>U283</td>
<td></td>
</tr>
<tr>
<td>U284</td>
<td></td>
</tr>
<tr>
<td>U285</td>
<td></td>
</tr>
<tr>
<td>U286</td>
<td></td>
</tr>
<tr>
<td>U287</td>
<td></td>
</tr>
<tr>
<td>U288</td>
<td></td>
</tr>
<tr>
<td>U289</td>
<td></td>
</tr>
<tr>
<td>U290</td>
<td></td>
</tr>
<tr>
<td>U291</td>
<td></td>
</tr>
<tr>
<td>U292</td>
<td></td>
</tr>
<tr>
<td>U293</td>
<td></td>
</tr>
<tr>
<td>U294</td>
<td></td>
</tr>
<tr>
<td>U295</td>
<td></td>
</tr>
<tr>
<td>U296</td>
<td></td>
</tr>
<tr>
<td>U297</td>
<td></td>
</tr>
<tr>
<td>U298</td>
<td></td>
</tr>
<tr>
<td>U299</td>
<td></td>
</tr>
<tr>
<td>U300</td>
<td></td>
</tr>
</tbody>
</table>
```
PROCEDURE TABLE (Concluded)

<table>
<thead>
<tr>
<th>COMMON PROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE 301</td>
</tr>
<tr>
<td>TABLE 302</td>
</tr>
<tr>
<td>TABLE 303</td>
</tr>
<tr>
<td>TABLE 304</td>
</tr>
<tr>
<td>TABLE 305</td>
</tr>
<tr>
<td>TABLE 306</td>
</tr>
<tr>
<td>TABLE 307</td>
</tr>
<tr>
<td>TABLE 308</td>
</tr>
<tr>
<td>TABLE 309</td>
</tr>
<tr>
<td>TABLE 310</td>
</tr>
<tr>
<td>TABLE 311</td>
</tr>
<tr>
<td>TABLE 312</td>
</tr>
<tr>
<td>TABLE 313</td>
</tr>
<tr>
<td>TABLE 314</td>
</tr>
<tr>
<td>TABLE 315</td>
</tr>
<tr>
<td>TABLE 316</td>
</tr>
<tr>
<td>TABLE 317</td>
</tr>
<tr>
<td>TABLE 318</td>
</tr>
<tr>
<td>TABLE 319</td>
</tr>
<tr>
<td>TABLE 320</td>
</tr>
<tr>
<td>TABLE 321</td>
</tr>
<tr>
<td>TABLE 322</td>
</tr>
<tr>
<td>TABLE 323</td>
</tr>
<tr>
<td>TABLE 324</td>
</tr>
<tr>
<td>TABLE 325</td>
</tr>
<tr>
<td>TABLE 326</td>
</tr>
<tr>
<td>TABLE 327</td>
</tr>
<tr>
<td>TABLE 328</td>
</tr>
<tr>
<td>TABLE 329</td>
</tr>
<tr>
<td>TABLE 330</td>
</tr>
<tr>
<td>TABLE 331</td>
</tr>
<tr>
<td>TABLE 332</td>
</tr>
<tr>
<td>TABLE 333</td>
</tr>
<tr>
<td>TABLE 334</td>
</tr>
<tr>
<td>TABLE 335</td>
</tr>
<tr>
<td>TABLE 336</td>
</tr>
<tr>
<td>TABLE 337</td>
</tr>
<tr>
<td>TABLE 338</td>
</tr>
<tr>
<td>TABLE 339</td>
</tr>
<tr>
<td>TABLE 340</td>
</tr>
<tr>
<td>TABLE 341</td>
</tr>
<tr>
<td>TABLE 342</td>
</tr>
<tr>
<td>TABLE 343</td>
</tr>
<tr>
<td>TABLE 344</td>
</tr>
<tr>
<td>TABLE 345</td>
</tr>
<tr>
<td>TABLE 346</td>
</tr>
<tr>
<td>TABLE 347</td>
</tr>
</tbody>
</table>

Routines that use COMMON PROC

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPRO</td>
<td>PROGRAM CODE</td>
</tr>
<tr>
<td>LINUM</td>
<td>END ITEM NUMBER</td>
</tr>
<tr>
<td>TITLE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>DESCN</td>
<td>DESIGN RESPONSIBILITY</td>
</tr>
<tr>
<td>NCODE</td>
<td>CLASS CODE</td>
</tr>
<tr>
<td>QUAN</td>
<td>QUANTITY</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>WEIGHT</td>
</tr>
<tr>
<td>MOD</td>
<td>STAGENAME</td>
</tr>
<tr>
<td>USLOC</td>
<td>USE LOCATION</td>
</tr>
<tr>
<td>FTPCDE</td>
<td>FUNCTIONAL CODE</td>
</tr>
<tr>
<td>LENGTH</td>
<td>LENGTH</td>
</tr>
<tr>
<td>WIDTH</td>
<td>WIDTH</td>
</tr>
<tr>
<td>HIGHT</td>
<td>HEIGHT</td>
</tr>
<tr>
<td>UNUST</td>
<td>UNIT COST</td>
</tr>
<tr>
<td>MEDIA</td>
<td>FLUID MEDIA</td>
</tr>
<tr>
<td>APDOC</td>
<td>APPLICABLE DOCUMENT</td>
</tr>
<tr>
<td>TITLE</td>
<td>DUMMY VARIABLE (NOT USED)</td>
</tr>
<tr>
<td>DATE</td>
<td>DATE</td>
</tr>
<tr>
<td>PROCM</td>
<td>PROGRAM TITLE</td>
</tr>
<tr>
<td>NLMINS</td>
<td>THE NUMBER OF LINES OF DESCRIPTION</td>
</tr>
<tr>
<td>NPAGE</td>
<td>PAGE COUNT</td>
</tr>
<tr>
<td>LADD</td>
<td>LINE COUNTER</td>
</tr>
<tr>
<td>DATX</td>
<td>INPUT DATE</td>
</tr>
</tbody>
</table>

COMMON

| COMMON | COMMON | COMMON |
|        | NPRO, LNUM, NCDE, MOD, QUAN, TITLE, DESCN | USLOC(12), APDOC(24), PROCM(24), NLMINS, DATE |
|        | COMMON MEDIA, NPRO, LNUM, NCDE, MOD, QUAN, TITLE, DESCN | COMMON MEDIA, NPRO, LNUM, NCDE, MOD, QUAN, TITLE, DESCN, FTPCDE |
|        | COMMON MEDIA, NPRO, LNUM, NCDE, MOD, QUAN, TITLE, DESCN | COMMON MEDIA, NPRO, LNUM, NCDE, MOD, QUAN, TITLE, DESCN |

END

END

INTEGER QUAN, WIDTH, HIGHT, WEIGHT

COMMON NPRO, LNUM(3), NCDE, MOD, QUAN, TITLE(72), DESCN(2), USLOC(12), APDOC(24), PROCM(24), NLMINS, DATE(24)

COMMON MEDIA(12), HIGHT, LENGTH, WIDTH, HIGHT, UNUST(3), FTPCDE

COMMON TITLE(120), DATX(24), NPAGE, LADD

END
SUBROUTINE UPDATE

C ********************************************************************** UPDT 1
C * SUBROUTINE UPDATE - THIS ROUTINE UPDATES THE MASTER DATA FILE WITH * UPDT 2
C * THE NEW INPUT INFORMATION SUPPLIED BY THE PROGRAM ANALYST * UPDT 3
C ********************************************************************** UPDT 4

SUBROUTINE UPDATE UPDT 5

#include COMMON UPDT 6
#include EQULY UPDT 7
integer R UPDT 8
DIMENSION WORDS(WORDS) UPDT 9
DATA NCND/'se/' UPDT 10
DATA NC/H/I// UPDT 11
DATA NCh2/12/ UPDT 12
DATA NC/1409/ UPDT 13
DATA NCRA/'ate/' UPDT 14
DATA BLANK/f UPDT 15
DATA Nw2 UPDT 16
DATA NRm2 UPDT 17
DATA NRRUR=O UPDT 18
DATA N0U T O . O UPDT 19
DATA NTOT.O UPDT 20
DATA NNDATA=O UPDT 21
DATA NODATA=U UPDT 22
rewind 1 UPDT 23
rewind 2 UPDT 24
rewind 3 UPDT 25
rewind 4 UPDT 26
rewind 5 UPDT 27
rewind 6 UPDT 28
read tape data from file UPDT 29
continue UPDT 30
read(1,ern=90, end=30) words UPDT 31
nout=3 UPDT 32
write (3,eq=91) words UPDT 33
go to 10 UPDT 34
continue UPDT 35
nlines=1 UPDT 36
up 33 K=7,120 UPDT 37
title(K)=blank UPDT 38
up 34 K=13,24
up 35 apdoc(K)=blank UPDT 39
date(1)=blank UPDT 40
date(2)=blank UPDT 41
c UPDT 42
read card data from file UPDT 43
read card type 1 UPDT 44
c UPDT 45
c UPDT 46
c UPDT 47
c UPDT 48
ncard=ncard+1 UPDT 49
read(nrr,200,errno=95,end=85)ncr,npro,einum,(title(1),i=1,6) UPDT 50
format(i1,2x,11,2x,2a6,a1,2x,6a6) UPDT 51
if(ncr.ne.1) go to 95 UPDT 52
if(npro.eq.nnhy=2 UPDT 53
if(npro.eq.1601go to 8u UPDT 54
ncard=ncard+1 UPDT 55
nflag=0 UPDT 56
read(r,201,errno=95,end=85)ncr UPDT 57
format(a1) UPDT 58
if(ncr.eq.ncrd1go to 90 UPDT 59
if(ncr.eq.ncr21go to 95 UPDT 60
if(ncr.eq.ncr21go to 9b UPDT 61
c UPDT 62
c UPDT 63
c UPDT 64
SUBROUTINE UPDATE (Continued)

65 40  N J = N L I N E S + 6 + 1
66  MJ = N L I N E S + 6 + 2
67  IF( N L I N E S .GE. 19) GO TO 78
68  HEAD(I:202, Err=95, End=85) TITLE(J, J=MJ, MJ)
69  N L I N E S = N L I N E S + 1
70  GO TO 36
71
72  C
73  C******READ CARD TYPE 2
74
75  45  READ(1,203, ERR=95, END=85) DESC, QUAN, LNGTH, MOD
76  203 FORMAT(3X, 1A6, 2X, 10A6, 2X, 1A6)
77
78  C
79  C******READ CARD TYPE 3
80
81  50  RCARD = NCARD + 1
82  READ(1,204, Err=95, End=85) ICR, USLOC, FCYCD, LNGTH, ITH, HGT, UNCT
83  204 FORMAT(3X, 1A6, 2X, 10A6, 2X, 1A6)
84  NCRK = NCR
85  IF(NCR.EQ.3) GO TO 95
86
87  C
88  C******READ CARD TYPE 4
89
90  55  NCR = NCARD + 1
91  READ(1,205, Err=95, End=85) ICR
92  IF(NCR.EQ.4) NCARD = NCARD
93  IF(NCR.EQ.4) GO TO 70
94  IF(NCR.EQ.4) NCR = 0
95  IF(NCR.EQ.4) NCARD = NCARD
96  IF(NCR.EQ.4) GO TO 80
97  GO TO 95
98  60  J = J + 2
99  READ(1,206, Err=95, End=85) ICR
100  205 FORMAT(3X, 1A6, 2X, 10A6, 2X, 1A6)
101  NCRK = BLANK
102  GO TO 50
103  70  J = J + 3
104  READ(1,207, Err=95, End=85) ICR
105  207 FORMAT(3X, 1A6, 2X, 10A6, 2X, 1A6)
106  GO TO 60
107  104  IC = 24
108
109  C
110  C******ERROR MESSAGES
111
112  90  WRITE(6,100) ***ERROR WHILE READING INPUT FILE***
113  100 FORMAT(1X,10A6) ***ERROR WHILE READING INPUT FILE***
114  99  STOP
115  91  WRITE(6,101) ***EXECUTION TERMINATED***
116  101 FORMAT(1X,10A6) ***EXECUTION TERMINATED***
117  98  GO TO 99
SUBROUTINE UPDATE (Continued)

95 CONTINUE
96   IF (NFLAG.EQ.1) GO TO 30
97   NFLAG = 1
98   NHR = 2
99   IF (ICARD.LT.1 .AND. NHR.EQ.0 .AND. EINUM(2).EQ.BLANK) GO TO 30
100  IF (NFLA6*EL.GT.I) GO TO 30
101  NFLAGI = 1
102  NRH2 = 1
103  IF (ITCARDU.EQ.'IAN'. .AND. EINUM(2).EQ.BLANK) GO TO 30
104  WR = 1
105  FORM 10 = '***ERROR WHILE READING INPUT FILE 2***' 
106  * IS OUT OF ORDER OR IS PUNCHED INCORRECT
107  IF (NERRUR.GT.30) GO TO 96
108  GO TO 30
109  WRITE(6,109)
110  FORM 11 = '***ERROR WHILE READING INPUT FILE 3***'
111  EXECUTION TERMINATED
112  EXECUTION TERMINATED
113  WRITE(6,113)
114  WRITE(3,LEHR=9)ORDS
115  READ(J3ERH97,EN-894)ORDS
116  CALL SRTHLLORDS)
117  CALL SRTSHET(*ORUS,SI6U)
118  CONTINUE
119  IF (DIATLI1 .EQ. BLANK .AND. DATE(2) .EQ. BLANK) GO TO 86
120  GO TO 153
121  WRITE(6,121)EINUM
122  WRITE(3,LEHR=9)ORDS
123  READ(J3ERH97,EN-894)ORDS
124  CALL SRTHLLORDS)
125  CALL SRTSHET(*ORUS,SI6U)
126  CONTINUE
127  END FILE 3
128  REWIND 3
129  CALL RSLW(IORDS)
130  CALL KEYAI203,1U8so'A''A'lI)
131  CALL KEYIWl2g,3S,72oA's'A
132  CALL SRTH(ORDS)
133  CALL SRTSHT
134  CONTINUE
135  IF (NFLA6*EL.GT.I) GO TO 30
SUBROUTINE UPDATE (Concluded)

175 153 NTOT = NTOT + 1
176 NUOT = 4
177 WRITE(4, ERR=911) WORDS
178 GO TO 89
179 CONTINUE
180 DO 87 I = 1, L
181 WRITE(4, ERR=911) WORDS
182 CALL SHRTRT(WORDS, $151)
183 DO 88 I = 1, 3
184 IF(IDORDS(I) + 1) NE. EINUM(I) GO TO 150
185 CONTINUE
186 IF(WORDS(I), EQ, 0) NDDATA = NDDATA + 1
187 IF(WORDS(I), EQ, 0) GO TO 89
188 UORDS(127) = DATX(I)
189 UORDS(128) = DATX(I)
190 NUDATA = NUDATA + 1
191 NTOT = NTOT + 1
192 NUOT = 4
193 WRITE(4, ERR=911) WORDS
194 GO TO 89
195 CONTINUE
196 UORDS(127) = DATX(I)
197 UORDS(128) = DATX(I)
198 NUDATA = NUDATA + 1
199 NTOT = NTOT + 1
200 NUOT = 4
201 WRITE(4, ERR=911) WORDS
202 GO TO 81
203 CONTINUE
204 IF(UDATE(I) .EQ. BLANK. AND. DATE(2) .EQ. BLANK) GO TO 152
205 NTOT = NTOT + 1
206 NUOT = 4
207 WRITE(4, ERR=911) WORDS
208 CONTINUE
209 END FILE 4
210 WRITE(6, 117) NDDATA, NDDATA, NDDATA, NTOT
211 107 FORMAT(* **GSE ATTRIBUTE DATA FILE HAS BEEN UPDATED***)
212 **NEW ENTITIES*%
213 **UPDATED ENTITIES**
214 **DELETED ENTITIES**
215 **TOTAL ENTITIES USED DURING THIS RUN**
216 RETURN
217 END
SUBROUTINE WGTMET

** DIMENSION IA(6) **

** C*******FLO IS A LIBRARY FUNCTION TO OBTAIN BITS FROM ALPHANUMERIC NUMBERS **

** C*******CALCULATE INTEGER NUMBER (DEPENDING ON NUMBER OF DIGITS) **

** C*******CONVERT REAL NUMBER TO METRIC **

** C*******CALL SUBROUTINE TO CONVERT REAL NUMBER TO ALPHANUMERIC NUMBER **

** C*******RELATIVELY SELECTED METRIC **

** C*******END **
APPENDIX C. SUPPLEMENT RUN PROCEDURES

PREVIOUS PAGE BLANK NOT FILMED
APPENDIX C. SUPPLEMENT RUN PROCEDURES
(The Univac 1108 Run Cards)

Before a run can be submitted to a computer operator at the Univac
1108 remotes or at central site (Computation Laboratory), two special
computer cards must appear in front of all other cards in the card-deck
runstream. They are simply referred to as Run Card No. 1 (Fig. C-1) and
Run Card No. 2 (Fig. C-2).

The following procedure should aid in completing these forms
correctly:

1. Punch Run Card No. 1 exactly as shown in Figure C-1 except that
   keypuncher punches the first six letters of his last name in card columns 24
   through 29. If the name is less than six characters, the space should be filled
   with some arbitrary letters.

2. Punch Run Card No. 2 exactly as shown in Figure C-2 on the
   preprinted green run card except for any comment. Note that one really needs
   only to duplicate Run Card No. 2 from Run Card No. 1 except card columns
   65 through 80 are left blank.

3. Printed information must be added to Run Card No. 2 as shown in
   Figure C-2. Write "32" for CORESIZE and "207" or "225" for BIN NO. Write
   RUN "1 of 1." Mark an "X" in the NO columns of PUNCH$ and PLOTS.
   Write the current tape number (4 or 5 digits) under INPUT TAPES. Remem-
   ber that these two cards precede all other cards in the runstream.
Figure C-1. Sample Run Card No. 1.
**Figure C-2. Sample Run Card No. 2.**

<table>
<thead>
<tr>
<th>OUTPUT ITEMS</th>
<th>YES</th>
<th>NO</th>
<th>OPER. NO.</th>
<th>RUN 1 or 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUNCH $</td>
<td>X</td>
<td></td>
<td></td>
<td>INPUT TAPES</td>
</tr>
<tr>
<td>PLOTS</td>
<td>X</td>
<td></td>
<td></td>
<td>BBBBBB</td>
</tr>
</tbody>
</table>

**Table:**

<table>
<thead>
<tr>
<th>CORE SIZE</th>
<th>BIN NO.</th>
<th>CLOCK NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 K</td>
<td>225</td>
<td></td>
</tr>
</tbody>
</table>

**Remarks:**

- AP1121 BSC
- RUN CARD NO. 2
The information in this report has been reviewed for security classification. Review of any information concerning Department of Defense or Atomic Energy Commission programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

This document has also been reviewed and approved for technical accuracy.

O. M. HARDAGE
Chief, Operations Planning and Analysis Branch

F. KURTZ
Chief, Operations Development Division

H. E. THOMASON
Systems Analysis and Integration Laboratory