# Texas Flexible Pavement Data Base

Research performed in cooperation with FHWA. Research Study Title: Texas Flexible Pavement Data Base.

This volume of the final report is the programmers manual for the Texas Flexible Pavement Database System. It describes the system from the programmer's point of view and includes a system overview, general program narratives, flow diagrams, program specifications, file layouts, report layouts, sample reports, sample screens and program listings.
TEXAS FLEXIBLE PAVEMENT DATABASE

VOLUME II. PROGRAMMER'S MANUAL

By

Rebecca Yette, Trevor Pereira and Victor Wong

Research Report
456-1F Volume II

on

Research Study Number 2-8-86-456
Texas Flexible Pavement Database

Sponsored By
Texas State Department of Highways & Public Transportation

In Cooperation with
Federal Highway Administration

August 1988

Texas Transportation Institute
Texas A&M University System
College Station, Texas
# METRIC (SI*) CONVERSION FACTORS

## APPROXIMATE CONVERSIONS TO SI UNITS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>When You Know</th>
<th>Multiply By</th>
<th>To Find</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>in</td>
<td>inches</td>
<td>2.54</td>
<td>centimetres</td>
<td>cm</td>
</tr>
<tr>
<td>ft</td>
<td>feet</td>
<td>0.3048</td>
<td>metres</td>
<td>m</td>
</tr>
<tr>
<td>yd</td>
<td>yards</td>
<td>0.914</td>
<td>metres</td>
<td>m</td>
</tr>
<tr>
<td>mi</td>
<td>miles</td>
<td>1.61</td>
<td>kilometres</td>
<td>km</td>
</tr>
</tbody>
</table>

## LENGTH

<table>
<thead>
<tr>
<th>in²</th>
<th>square inches</th>
<th>645.2</th>
<th>centimetres squared</th>
<th>cm²</th>
</tr>
</thead>
<tbody>
<tr>
<td>ft²</td>
<td>square feet</td>
<td>0.0929</td>
<td>metres squared</td>
<td>m²</td>
</tr>
<tr>
<td>yd²</td>
<td>square yards</td>
<td>0.836</td>
<td>metres squared</td>
<td>m²</td>
</tr>
<tr>
<td>mi²</td>
<td>square miles</td>
<td>2.59</td>
<td>kilometres squared</td>
<td>km²</td>
</tr>
<tr>
<td>ac</td>
<td>acres</td>
<td>0.395</td>
<td>hectares</td>
<td>ha</td>
</tr>
</tbody>
</table>

## AREA

<table>
<thead>
<tr>
<th>mm²</th>
<th>millimetres squared</th>
<th>0.0016</th>
<th>square inches</th>
<th>in²</th>
</tr>
</thead>
<tbody>
<tr>
<td>m²</td>
<td>metres squared</td>
<td>10.764</td>
<td>square feet</td>
<td>ft²</td>
</tr>
<tr>
<td>km²</td>
<td>kilometres squared</td>
<td>0.39</td>
<td>square miles</td>
<td>mi²</td>
</tr>
<tr>
<td>ha</td>
<td>hectares (10 000 m²)</td>
<td>2.53</td>
<td>acres</td>
<td>ac</td>
</tr>
</tbody>
</table>

## MASS (weight)

<table>
<thead>
<tr>
<th>g</th>
<th>grams</th>
<th>0.0353</th>
<th>ounces</th>
<th>oz</th>
</tr>
</thead>
<tbody>
<tr>
<td>kg</td>
<td>kilogramsm</td>
<td>2.205</td>
<td>pounds</td>
<td>lb</td>
</tr>
<tr>
<td>Mg</td>
<td>megagrams (1 000 kg)</td>
<td>1.103</td>
<td>short tons</td>
<td>T</td>
</tr>
</tbody>
</table>

## VOLUME

<table>
<thead>
<tr>
<th>fl oz</th>
<th>fluid ounces</th>
<th>29.57</th>
<th>millilitres</th>
<th>ml</th>
</tr>
</thead>
<tbody>
<tr>
<td>gal</td>
<td>gallons</td>
<td>3.785</td>
<td>litres</td>
<td>L</td>
</tr>
<tr>
<td>ft³</td>
<td>cubic feet</td>
<td>0.0283</td>
<td>metres cubed</td>
<td>m³</td>
</tr>
<tr>
<td>yd³</td>
<td>cubic yards</td>
<td>0.0765</td>
<td>metres cubed</td>
<td>m³</td>
</tr>
</tbody>
</table>

## TEMPERATURE (exact)

<table>
<thead>
<tr>
<th>°C</th>
<th>Fahrenheit temperature</th>
<th>9/5 (then subtracting 32)</th>
<th>Celsius temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>°F</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTE: Volumes greater than 1000 L shall be shown in m³.

**TEMPERATURE (exact)**

These factors conform to the requirement of FHWA Order 5190.1A.

* SI is the symbol for the International System of Measurements
DISCLAIMER

The statements and claims expressed in this document are statements and claims of the authors, and do not necessarily represent the official position of The State of Texas, The State Department of Highways and Public Transportation, or any political subdivision of the State or Federal Government regarding the subject matter.
ABSTRACT

This volume of the final report is the programmers manual for the Texas Flexible Pavement Database System. It describes the system from the programmer's point of view and includes a system overview, general program narratives, flow diagrams, program specifications, file layouts, report layouts, sample reports, sample screens and program listings.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INTRODUCTION</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>CHAPTER 1: MAIN MENU DRIVER PROGRAM</strong></td>
<td>9</td>
</tr>
<tr>
<td>* Program DEMAIN.PRG</td>
<td></td>
</tr>
<tr>
<td>General Narrative</td>
<td>11</td>
</tr>
<tr>
<td>Program Flow Diagram</td>
<td>12</td>
</tr>
<tr>
<td>Program Specification</td>
<td>13</td>
</tr>
<tr>
<td>Menu Screens</td>
<td>14</td>
</tr>
<tr>
<td>Program Listing</td>
<td>15</td>
</tr>
<tr>
<td>DEMAIN.PRG</td>
<td>15</td>
</tr>
<tr>
<td>ERROR.PRB</td>
<td>20</td>
</tr>
<tr>
<td><strong>CHAPTER 2: INQUIRY SUBSYSTEM</strong></td>
<td>21</td>
</tr>
<tr>
<td>* General Narrative</td>
<td>23</td>
</tr>
<tr>
<td>* Program INQUIRY.PRG</td>
<td></td>
</tr>
<tr>
<td>Program Flow Diagram</td>
<td>24</td>
</tr>
<tr>
<td>Program Specification</td>
<td>25</td>
</tr>
<tr>
<td>Inquiry Screens</td>
<td>27</td>
</tr>
<tr>
<td>Screen Format Programs</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>41</td>
</tr>
<tr>
<td>Layer Identification</td>
<td>43</td>
</tr>
<tr>
<td>Layer Thickness Across The Road</td>
<td>44</td>
</tr>
<tr>
<td>Geometric &amp; Shoulder Information</td>
<td>45</td>
</tr>
<tr>
<td>Surface</td>
<td>46</td>
</tr>
<tr>
<td>Subgrade</td>
<td>47</td>
</tr>
<tr>
<td>Visual Rating</td>
<td>48</td>
</tr>
<tr>
<td>Serviceability Index</td>
<td>50</td>
</tr>
<tr>
<td>Falling Weight SSI</td>
<td>51</td>
</tr>
<tr>
<td>Dynafect Measurement</td>
<td>53</td>
</tr>
<tr>
<td>Skid</td>
<td>54</td>
</tr>
<tr>
<td>Traffic</td>
<td>55</td>
</tr>
<tr>
<td>Weather</td>
<td>56</td>
</tr>
<tr>
<td>Environment</td>
<td>57</td>
</tr>
<tr>
<td>County Name</td>
<td>58</td>
</tr>
<tr>
<td>Material Type Classification</td>
<td>59</td>
</tr>
<tr>
<td>Type of Pavement</td>
<td>60</td>
</tr>
<tr>
<td>District Temperature Constant</td>
<td>61</td>
</tr>
<tr>
<td>Widening Flag</td>
<td>62</td>
</tr>
<tr>
<td>Layer Description</td>
<td>63</td>
</tr>
<tr>
<td>Functional Classification</td>
<td>64</td>
</tr>
<tr>
<td>Program Listings</td>
<td></td>
</tr>
<tr>
<td>INQUIRY.PRG</td>
<td>65</td>
</tr>
<tr>
<td>INQ_COLL.PRG (Procedure)</td>
<td>71</td>
</tr>
</tbody>
</table>
**CHAPTER 3: REPORT SUBSYSTEM**

* General Narrative.

* Program REPORTS.PRG
  
  Program Flow Diagram. ........................................ 80
  Program Specification ........................................ 83
  Report Screens. ................................................ 86
  Sample Reports. ................................................ 91
  Location .......................................................... 93
  Location Section ............................................... 94
  Layer Identification .......................................... 95
  Double Surface Treatment .................................... 96
  Layer Thickness Across The Road. ......................... 97
  Geometric & Shoulder Information ......................... 98
  Surface .......................................................... 99
  Subgrade .......................................................... 100
  Visual ............................................................. 101
  Serviceability Index ......................................... 102
  Falling Weight SSI ............................................. 103
  Dynaplect Measurement ....................................... 105
  Skid .............................................................. 106
  Traffic ........................................................... 107
  Weather ........................................................... 108
  Environment ...................................................... 109
  Summary Sheet .................................................. 110
  Inventory Update ............................................... 111
  County Name Table ............................................. 113
  Material Type Table .......................................... 114
  Type of Pavement Table ..................................... 115
  District Temperature Constant Table .................... 116
  Layer Description Table .................................... 117
  Widening Flag Table .......................................... 118
  Functional Classification Table ......................... 119

Program Listings

  REPORTS.PRG ..................................................... 120
  SUMMDIST.PRG ................................................... 128
  SUMMSID.PRG ..................................................... 131
  SUMMALL.PRG ..................................................... 134
  GETLOCAT.PRG ................................................... 136
  GETENVIR.PRG .................................................... 144
  GETPAVON.PRG ..................................................... 146
  GETSKID.PRG ...................................................... 150
  PRNTREPT.PRG ..................................................... 153
  SUM2DIST.PRG ..................................................... 159
  SUMM2STD.PRG ..................................................... 163
  SUM2FILE.PRG ...................................................... 164
  GETLOCT2.PRG ...................................................... 166
  PRNTREPT2.PRG ..................................................... 174
  PRNTREPT3.PRG ..................................................... 179
CHAPTER 4: EDIT & UPDATE SUBSYSTEM 

* General Narrative. ..... 185

Section 1: Edit and Update Driver. ..... 187

* Program EDITUPDT.PRG
  Program Narrative on Edit and Update Driver ..... 189
  Diagram of General Program Flow ..... 190
  Menu Screens. ..... 191
  Program Specification ..... 193
  Program Listings
    EDITUPDT.PRG ..... 195
    TABLFILE.PRG ..... 199

Section 2: Monitoring Data Update. ..... 201

* Program READPES.BAT
  Narrative on PES Tape to Disk ..... 203
  Program Specification ..... 204
  Program Listings
    READPES.BAT. ..... 205
    READPES.CMD. ..... 206

* Program PESMOD.PAS
  Narrative on PES Modification ..... 207
  Diagram of General Program Flow ..... 208
  Program Specification ..... 209
    Procedure. ..... 210
    File Layouts ..... 211
    Notes on Output Record Layouts ..... 223
    Missing Data Report Layout ..... 226
  Program Listing
    PESMOD.PAS ..... 227

* Program PESUP.PRG
  Narrative on PES Update ..... 245
  Diagram of General Program Flow ..... 246
  Program Specification ..... 247
  Missing and Inconsistent Data Report Layout ..... 257
  Program Listing
    PESUP.PRG. ..... 258
    PES_SKED.PRG ..... 261
    PES_VISL.PRG ..... 266
    PES_SSI.PRG ..... 273
    PES_MRM.PRG. ..... 278

Section 3: Inventory Data Update ..... 283

* Program ENTRLAYR.PRG
  Narrative on Adding Inventory Data. ..... 285
  Diagram of General Program Flow ..... 286
  Flowchart of Program Logic. ..... 287
  Program Specification ..... 294
Add Inventory Screens ............................................. 296
Program Listing
ENTLAYR.PRG ....................................................... 300
ADDLAYR.PRG ....................................................... 304
INV_BKUP.PRG ..................................................... 316
INV_UPDT.PRG ..................................................... 324
LOCNCHK.PRG ...................................................... 332
LAYRCHK.PRG ...................................................... 337
LAYTCHK.PRG ...................................................... 345
GEOSCHK.PRG ...................................................... 351
SURFCHK.PRG ...................................................... 355
SUBGCHEK.PRG ..................................................... 359

* Program CHNGLAYR.PRG
Narrative on Changing Inventory Data. .......................... 363
Diagram of General Program Flow ................................. 364
Flowchart of Program Logic ....................................... 365
Program Specification ............................................. 368
Change Inventory Screens ......................................... 370
Screen Format Listings
Geometric and Shoulder Information ............................. 374
Layer Identification .............................................. 375
Layer Thickness Across the Road ................................. 376
Location ........................................................... 377
Subgrade ........................................................... 379
Surface ............................................................ 380
Program Listing
CHNGLAYR.PRG .................................................... 381
CHNGBKUP.PRG .................................................... 385
CHEKGEOS.PRG .................................................... 393
CHEKLAYR.PRG .................................................... 398
CHEKLAYT.PRG .................................................... 404
CHEKLOCN.PRG .................................................... 409
CHEKSUBG.PRG ..................................................... 414
CHEKSURF.PRG ..................................................... 418

Section 4: Traffic Data Update .................................... 423

* Program READIRAF.BAT
Narrative on Traffic Tape to Disk ................................ 425
Program Specification ............................................. 426
Program Listing
READIRAF.BAT .................................................... 427
READIRAF.CMD .................................................... 428

* Program TRAFUPD.PRG
Narrative on Traffic Update ...................................... 429
Program Flow Diagram ........................................... 430
Program Specification ........................................... 431
File Record Layout .............................................. 432
Program Listing
TRAFUPD.PRG ..................................................... 434
Program Flow Diagram .......................................... 435
Section 3: Building Model File

* General Narrative.

* Program MODEL.PRG
  Diagram of General Program Flow
  Program Specification
  Model Menu Screen
  File Layouts
  Program Listing
  MODEL.PRG

* Program MODELING.PRG
  Program Specification
  Model Menu Screen
  Program Listing
  MODELING.PRG

* Program MODL_LOC.PRG
  Program Specification
  Diagram of General Program Flow
  Program Listing
  MODL_LOC.PRG

* Program MODL_DIS.PRG
  Program Specification
  Diagram of General Program Flow
  Program Listing
  MODL_DIS.PRG
  MODL_R_B.PAS

* Program MODL_LAY.PRG
  Program Specification
  Diagram of General Program Flow
  Program Listing
  MODL_LAY.PRG
  BASETHCK.PAS

* Program MODL_CEA.PRG
  Program Specification
  Diagram of General Program Flow
  Program Listing
  MODL_CEA.PRG

* Program MODL_ENV.PRG
  Program Specification
  Diagram of General Program Flow
  Program Listing
  MODL_ENV.PRG
Subprogram 1: STLOG.PAS
  Program Specification. ........................................ 436
  File Record Layout ............................................ 437
  Program Listing: STLOG.PAS .................................... 438
  Program Flow Diagram .......................................... 441
Subprogram 2: SIDITLOG.PRG
  Program Specification. ......................................... 442
  File Record Layout ............................................ 443
  Diagram of Subprogram Flow ................................... 444
  Program Listing: SIDITLOG.PRG ................................. 445
  Program Flow Diagram .......................................... 447
Subprogram 3: LOGITRAF.PRG
  Program Specification. ......................................... 448
  File Record Layout ............................................ 449
  Program Listing: LOGITRAF.PRG ................................. 450
  Program Flow Diagram .......................................... 452
Subprogram 4: NEWITRAF.PRG
  Program Specification. ......................................... 453
  Program Listing: NEWITRAF.PRG ................................. 454

CHAPTER 5: APPLICATIONS SUBSYSTEM ............................... 455

* General Narrative. ............................................. 457

Section 1: Application Driver. .................................. 459

* Program APPLICAT.PRG
  Subsystem Driver Program Flow Diagram ..................... 461
  Narrative on Application Driver ............................. 462
  Program Specification ......................................... 463
  Menu Screens for Applications ................................. 464
  Program Listing
    APPLICAT.PRG .................................................. 465

Section 2: Accumulated 18KIP vs Alligator Cracking/Rutting/PSI 467

* Program GRAPH18K.PRG
  General Narrative .............................................. 469
  Diagram of General Program Flow .............................. 470
  Program Specification ......................................... 471
    Graph Menu Screen. ........................................... 472
    File Layout. .................................................. 473
  Program Listing
    GRAPH18K.PRG .................................................. 476

* Program GRAPH18K.PAS
  Program Specification ......................................... 478
  Program Listing
    GRAPH18.PAS .................................................. 481
    DECLARE.PAS .................................................. 494
    MYGRAPH.PAS .................................................. 495
    RHOBETA.PAS .................................................. 496
Section 4: Create the Distress File. ....... 585

* Program DISTRESS.PRG
  Narrative for Distress. .............. 587
  General Program Flow Diagram. .... 588
  Program Specification .............. 589
  File Layout ........................ 590
  Program Listing
    DISTRESS.PRG .................... 592
    DISTVISL.PRG .................... 596
CHAPTER 6: DATABASE MAINTENANCE SUBSYSTEM

* General Narrative. ........................................... 601

* Reindex Files Program - REINDEX.PRG
  Program Specification ...................................... 602
  Program Listing
  REINDEX.PRG

* Installation Program - INSTDEFL.PRG
  Program Specification ...................................... 606
  Program Listing
  INSTDEFL.PRG .................................................. 607

* Backup Files Program - BACKUP.PRG ................. 608
  Program Specification ...................................... 608
  Program Listing
  BACKUP.PRG .................................................. 609
LIST OF APPENDICES

APPENDIX A: File Layouts

<table>
<thead>
<tr>
<th>File Layout</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNVTBL.DBF</td>
<td>623</td>
</tr>
<tr>
<td>DISTRESS.DBF</td>
<td>624</td>
</tr>
<tr>
<td>DISTTEMP.DBF</td>
<td>625</td>
</tr>
<tr>
<td>DYNAXL.DBF</td>
<td>626</td>
</tr>
<tr>
<td>ENV.DBF</td>
<td>627</td>
</tr>
<tr>
<td>FALLWTGT.DBF</td>
<td>628</td>
</tr>
<tr>
<td>FUNCLBL.DBF</td>
<td>630</td>
</tr>
<tr>
<td>GEOSHO.DBF</td>
<td>631</td>
</tr>
<tr>
<td>LAYER.DBF</td>
<td>632</td>
</tr>
<tr>
<td>LAYERTBL.DBF</td>
<td>633</td>
</tr>
<tr>
<td>LAYTHICK.DBF</td>
<td>634</td>
</tr>
<tr>
<td>LOCATION.DBF</td>
<td>635</td>
</tr>
<tr>
<td>MATTBL.DBF</td>
<td>636</td>
</tr>
<tr>
<td>MODEL.DBF</td>
<td>637</td>
</tr>
<tr>
<td>PAVETYPE.DBF</td>
<td>638</td>
</tr>
<tr>
<td>SI.DBF</td>
<td>639</td>
</tr>
<tr>
<td>SKID.DBF</td>
<td>640</td>
</tr>
<tr>
<td>SUBGRADE.DBF</td>
<td>641</td>
</tr>
<tr>
<td>SURFACE.DBF</td>
<td>642</td>
</tr>
<tr>
<td>TRAFFIC.DBF</td>
<td>643</td>
</tr>
<tr>
<td>VISUAL.DBF</td>
<td>644</td>
</tr>
<tr>
<td>WEATHER.DBF</td>
<td>645</td>
</tr>
<tr>
<td>WIDENFG.DBF</td>
<td>646</td>
</tr>
</tbody>
</table>

APPENDIX B: Variable Descriptions For All Master Files............ 647

APPENDIX C: Minimum System Configuration.......................... 653

APPENDIX D: Initial Installation.................................... 657
# List of Exhibits

<table>
<thead>
<tr>
<th>Exhibit</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Location File Report</td>
<td>93</td>
</tr>
<tr>
<td>2</td>
<td>Test Section Location File Report</td>
<td>94</td>
</tr>
<tr>
<td>3</td>
<td>Layer Identification File Report</td>
<td>95</td>
</tr>
<tr>
<td>4</td>
<td>SID Numbers with Two Course Surface (DST) Report</td>
<td>96</td>
</tr>
<tr>
<td>5</td>
<td>Layer Thickness Across The Road File Report</td>
<td>97</td>
</tr>
<tr>
<td>6</td>
<td>Geometric &amp; Shoulder Information File Report</td>
<td>98</td>
</tr>
<tr>
<td>7</td>
<td>Surface File Report</td>
<td>99</td>
</tr>
<tr>
<td>8</td>
<td>Subgrade File Report</td>
<td>100</td>
</tr>
<tr>
<td>9</td>
<td>Visual Rating File Report</td>
<td>101</td>
</tr>
<tr>
<td>10</td>
<td>Serviceability Index File Report</td>
<td>102</td>
</tr>
<tr>
<td>11</td>
<td>Falling Weight SSI File Report - Sections 1 &amp; 2</td>
<td>103</td>
</tr>
<tr>
<td>12</td>
<td>Dynaflect Measurement File Report</td>
<td>105</td>
</tr>
<tr>
<td>13</td>
<td>Skid Measurement File Report</td>
<td>106</td>
</tr>
<tr>
<td>14</td>
<td>Traffic File Report</td>
<td>107</td>
</tr>
<tr>
<td>15</td>
<td>Weather File Report</td>
<td>108</td>
</tr>
<tr>
<td>16</td>
<td>Environment File Report</td>
<td>109</td>
</tr>
<tr>
<td>17</td>
<td>Summary Report</td>
<td>110</td>
</tr>
<tr>
<td>18</td>
<td>Inventory Update Report</td>
<td>111</td>
</tr>
<tr>
<td>19</td>
<td>County Name Table Report</td>
<td>113</td>
</tr>
<tr>
<td>20</td>
<td>Material Type Classification Table Report</td>
<td>114</td>
</tr>
<tr>
<td>21</td>
<td>Type of Pavement Table Report</td>
<td>115</td>
</tr>
<tr>
<td>22</td>
<td>District Temperature Constant Table Report</td>
<td>116</td>
</tr>
<tr>
<td>23</td>
<td>Layer Description Table Report</td>
<td>117</td>
</tr>
<tr>
<td>24</td>
<td>Widening Table Report</td>
<td>118</td>
</tr>
<tr>
<td>25</td>
<td>Functional Classification Table Report</td>
<td>119</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>System Overview Diagram</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Relational Database Index Structure</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Program Flow Diagram Symbols</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>Subdirectory Structure</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>Main Menu Driver - Program Flow Diagram</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>Inquiry Subsystem - Program Flow Diagram</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>Report Subsystem - Program Flow Diagram</td>
<td>80</td>
</tr>
<tr>
<td>8</td>
<td>Report Subsystem - Program Flow Diagram</td>
<td>81</td>
</tr>
<tr>
<td>9</td>
<td>Report Subsystem - Program Flow Diagram</td>
<td>82</td>
</tr>
<tr>
<td>10</td>
<td>Edit and Update - Program Flow Diagram</td>
<td>189</td>
</tr>
<tr>
<td>11</td>
<td>PES Modification - Program Flow Diagram</td>
<td>208</td>
</tr>
<tr>
<td>12</td>
<td>PES Update - Program Flow Diagram</td>
<td>246</td>
</tr>
<tr>
<td>13</td>
<td>Inventory Data - Add Process</td>
<td>286</td>
</tr>
<tr>
<td>14</td>
<td>Inventory Data - Add Process Program Flow Chart</td>
<td>287</td>
</tr>
<tr>
<td>15</td>
<td>Inventory Data - Add Process Program Flow Diagram</td>
<td>288</td>
</tr>
<tr>
<td>16</td>
<td>Inventory Data - Change Process</td>
<td>364</td>
</tr>
<tr>
<td>17</td>
<td>Inventory Data - Change Process Program Flow Chart</td>
<td>365</td>
</tr>
<tr>
<td>18</td>
<td>Inventory Data - Change Process Program Flow Diagram</td>
<td>366</td>
</tr>
<tr>
<td>19</td>
<td>Traffic Update - Program Flow Diagram</td>
<td>430</td>
</tr>
<tr>
<td>20</td>
<td>Subprogram STLOG.PAS - Program Flow Diagram</td>
<td>435</td>
</tr>
<tr>
<td>21</td>
<td>Subprogram SDIGLOG.PRG - Program Flow Diagram</td>
<td>441</td>
</tr>
<tr>
<td>22</td>
<td>Subprogram LOGTGRAF.PRG - Program Flow Diagram</td>
<td>447</td>
</tr>
<tr>
<td>23</td>
<td>Subprogram NEWTGRAF.PRG - Program Flow Diagram</td>
<td>452</td>
</tr>
<tr>
<td>24</td>
<td>Application Subsystem Driver Program Flow</td>
<td>461</td>
</tr>
<tr>
<td>25</td>
<td>Program Flow for Graphing 18KTP vs Area Distress or PSI</td>
<td>470</td>
</tr>
<tr>
<td>26</td>
<td>Program Flow for Building Model File Automatically by</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>Choosing Option 1 of Mode.Prg Menu</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Program Flow for Building Model File by Individual Data</td>
<td>513</td>
</tr>
<tr>
<td></td>
<td>Files by Choosing Option 2 of Model.Prg Menu</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Program Flow for Mod1_Loc.Prg</td>
<td>529</td>
</tr>
<tr>
<td>29</td>
<td>Program Flow for Mod1_Dis.Prg</td>
<td>533</td>
</tr>
<tr>
<td>30</td>
<td>Program Flow for Mod1_Lay.Prg</td>
<td>549</td>
</tr>
<tr>
<td>31</td>
<td>Program Flow for Mod1_Wea.Prg</td>
<td>556</td>
</tr>
<tr>
<td>32</td>
<td>Program Flow for Mod1_ENV.Prg</td>
<td>559</td>
</tr>
<tr>
<td>33</td>
<td>Program Flow for Mod1_Trf.Prg</td>
<td>562</td>
</tr>
<tr>
<td>34</td>
<td>Program Flow for Mod1_Sub.Prg</td>
<td>565</td>
</tr>
<tr>
<td>35</td>
<td>Program Flow for Mod1_Sho.Prg</td>
<td>568</td>
</tr>
<tr>
<td>36</td>
<td>Program Flow for Mod1_Suf.Prg</td>
<td>571</td>
</tr>
<tr>
<td>37</td>
<td>Program Flow for Mod1_Tmp.Prg</td>
<td>575</td>
</tr>
<tr>
<td>38</td>
<td>Program Flow for Mod1_Dyn.Prg</td>
<td>578</td>
</tr>
<tr>
<td>39</td>
<td>Program Flow for Mod1_Fal.Prg</td>
<td>582</td>
</tr>
<tr>
<td>40</td>
<td>Program Flow for Mod1_Distress.Prg</td>
<td>588</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>List of Master, Table, and Primary Auxiliary Files</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Standard File Extensions</td>
<td>4</td>
</tr>
</tbody>
</table>
INTRODUCTION

The development of the Texas Flexible Pavement Database System was sponsored by the Texas Department of Highways and Public Transportation and conducted by Texas Transportation Institute. Whereas Volume 1 of this report explains the purpose of the system and provides a User's Manual, Volume 2 is a detailed explanation of the system from the programmer's point of view. Volume 2 assumes the reader is familiar with Volume 1 of this report, and with dBASE III PLUS and Borland TURBO Pascal.

The Texas Flexible Pavement Database System is a microcomputer relational database system written for an IBM XT in dBASE III PLUS and Pascal. The dBASE database contains location, inventory, environmental, pavement condition (monitoring), and traffic information for randomly selected sections of pavement in Texas. Sections average two (2) miles in length. The system is used to edit and update, report, display on the monitor, and analyze the data. A system overview diagram is shown in Figure 1. The following are primary inputs to the system:

a) **Pavement Evaluation System (PES) data** - used to update the Skid, Serviceability Index, Falling Weight, Visual Rating, and Location Files.

b) **Roadway Information (RIFILE) data** - used to update the Traffic File.

c) **Roadlife maps and District Maintenance records** - used to update the Location, Geometric and Shoulder, Layer Identification, Surface, Subgrade, and Layer Thickness Across the Road Files. Unlike the PES and RIFILE data, this information must be input through data entry screens.

The following are primary outputs of the system:

a) **Data Inquiry Screens** - any information stored in the database can be viewed on the monitor.

b) **Performance versus Accumulated 18 KIP Equivalent Axle Loads Graph** - alligator cracking, rutting, or PSI versus 18 KEAL points are plotted on a graph which is displayed on the monitor. A curve is then fit through these points and also displayed on the monitor.

c) **Model Development File** - this file contains rho and beta constants which describe the shape of the degradation curve mentioned in (b) and independent variables (layer thickness, environmental factors, traffic levels, etc.) which are needed to develop performance models. This file can be used directly by the SAS programming language.

d) **Various Reports** including the following:

1) "Raw" data listings - listings of all information contained in any of the master files and table files

2) Summary listings - information about a section printed on a single page

3) **Inventory Update Forms** - a two-part report in which the first page contains inventory information about
TEXAS FLEXIBLE PAVEMENT DATABASE SYSTEM
a section on one or two sheets, and the second page is a blank form which can be used by the District Offices to record maintenance activities to update inventory data in the database.

4) Edit and Update Reports - reports which list errors or inconsistencies in data to be added to or changed in the database.

Table 1 lists the relational database master files, the table files, and two primary auxiliary files.

**Table 1**

**List of Master Files**

<table>
<thead>
<tr>
<th>dBASE III Name</th>
<th>Type of Information</th>
<th>Data Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>DYNAFIELD</td>
<td>Dynaflect</td>
<td>Monitoring</td>
</tr>
<tr>
<td>ENV</td>
<td>Environment</td>
<td>Environmental</td>
</tr>
<tr>
<td>FALLWEIGHT</td>
<td>Falling Weight</td>
<td>Monitoring</td>
</tr>
<tr>
<td>GEOGHO</td>
<td>Geometric and Shoulder</td>
<td>Inventory</td>
</tr>
<tr>
<td>LAYER</td>
<td>Layer Identification</td>
<td>Inventory</td>
</tr>
<tr>
<td>LAYTHICK</td>
<td>Layer Thickness Across the Road</td>
<td>Inventory</td>
</tr>
<tr>
<td>LOCATION</td>
<td>Location</td>
<td>Monitoring</td>
</tr>
<tr>
<td>SI</td>
<td>Serviceability Index</td>
<td>Monitoring</td>
</tr>
<tr>
<td>SKID</td>
<td>Skid Measurements</td>
<td>Monitoring</td>
</tr>
<tr>
<td>SUBGRADE</td>
<td>Subgrade Layer</td>
<td>Inventory</td>
</tr>
<tr>
<td>SURFACE</td>
<td>Surface Layer</td>
<td>Inventory</td>
</tr>
<tr>
<td>TRAFFIC</td>
<td>Traffic</td>
<td>Traffic</td>
</tr>
<tr>
<td>VISUAL</td>
<td>Visual Rating</td>
<td>Monitoring</td>
</tr>
<tr>
<td>WEATHER</td>
<td>Weather</td>
<td>Environmental</td>
</tr>
</tbody>
</table>

**List of Table Files**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNTYBL</td>
<td>County Name</td>
</tr>
<tr>
<td>DISTTEMP</td>
<td>District Temperature Constant</td>
</tr>
<tr>
<td>FUNCTBL</td>
<td>Functional Classification Table</td>
</tr>
<tr>
<td>LAYERBL</td>
<td>Layer Description Table</td>
</tr>
<tr>
<td>MATTBL</td>
<td>Material Type Table</td>
</tr>
<tr>
<td>PAVETYPE</td>
<td>Pavement Type Table</td>
</tr>
<tr>
<td>WIDENFLAG</td>
<td>Widening Flag Table</td>
</tr>
</tbody>
</table>

**List of Primary Auxiliary Files**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISTRESS</td>
<td>Distress and 18 KIP</td>
</tr>
<tr>
<td>MODEL</td>
<td>Model (used to develop performance equations)</td>
</tr>
</tbody>
</table>

Record layouts for the files listed in Table 1 are provided in Appendix A. These files are stored in the subdirectory \PAVEDB\FILES. All dBASE
master files are indexed by the key fields which are indicated on the file layouts. The index files are stored in the subdirectory \PAVEDB\INDEXES. Figure 2 shows the master index structure for the database master files. Appendix B lists in alphabetical order all fields contained in the master files and in which files the fields are found.

The system is composed of a main menu program and five subsystems which are as follows:

a) Inquiry - displays on the monitor master file and table file data
b) Report - produces hard copies of data contained in the files
c) Edit and Update - modifies the data in the database
d) Applications - produces graphs and a model file both of which can be used in the development of performance models
e) Database Maintenance - reindexes the master files, sets up the defaults for the printer and path, and copies the master files onto floppy diskettes for backup purposes

The main menu initiates each subsystem.

This manual is organized in the same order as the subsystems are listed above. Each subsystem is contained in a separate chapter. The following is provided for each subsystem:

General Narratives
Program Flow Diagrams
Program Specifications
Screen Layouts
File Layouts
Program Listings

Sample reports are provided in Chapter 3. File layouts for files listed in Table 1 are not listed in the chapters since they can be found in Appendix A. Table 2 explains the standard file extensions used in the Texas Flexible Pavement System to identify the various types of files.

Table 2
Standard File Extensions

<table>
<thead>
<tr>
<th>Extension</th>
<th>File Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>.BAT</td>
<td>Batch File</td>
</tr>
<tr>
<td>.DAT</td>
<td>Data File NOT Identified by Other Extensions</td>
</tr>
<tr>
<td>.DBF</td>
<td>dBASE III PLUS File</td>
</tr>
<tr>
<td>.DLM</td>
<td>Comma Delimited File</td>
</tr>
<tr>
<td>.FMT</td>
<td>dBASE III PLUS Screen Format File</td>
</tr>
<tr>
<td>.FRM</td>
<td>dBASE III PLUS Report Form File</td>
</tr>
<tr>
<td>.MEM</td>
<td>dBASE III PLUS Memory Variable File</td>
</tr>
<tr>
<td>.NDX</td>
<td>dBASE III PLUS Index File</td>
</tr>
<tr>
<td>.PAS</td>
<td>TURBO Pascal Program File</td>
</tr>
<tr>
<td>.PRG</td>
<td>dBASE III PLUS Program File</td>
</tr>
<tr>
<td>.VUE</td>
<td>dBASE III PLUS View File</td>
</tr>
</tbody>
</table>
TEXAS FLEXIBLE PAVEMENT SYSTEM
RELATIONAL DATABASE INDEX STRUCTURE

FIGURE 2
Figure 3 illustrates and explains the symbols used in the manual to diagram program flow. Figure 4 shows the subdirectory structure of the system. The minimum system configuration is listed in Appendix C. The initial Texas Flexible Pavement System installation programs are documented in Appendix D.
CHAPTER 1

MAIN MENU DRIVER PROGRAM
Main Menu Driver Program Narrative

The program DEMAIN.PRG is the first program called by dBASE. The Flexible Pavement System is activated by FLEXPAVE.BAT which consists of the following command: DBASE DEMAIN. This command starts up dBASE and calls the main driver program DEMAIN.PRG.

DEMAIN.PRG checks to see if all the master files are present. It also verifies that the memory variable file INSTALL.MEM with the installation defaults is present. If it is not, the installation menu is run automatically before the main menu is brought up. If the file is present, DEMAIN.PRG displays the main menu on the screen and calls the appropriate program for the choice selected.

DEMAIN.PRG also activates the program ERROR.PRG which is called when an error occurs in the system. These programs and INSTALL.MEM can be found in subdirectory \PAVEDB.
Main Menu Driver - Program Flow Diagram

DBMAIN.PRG

INQUIRY.PRG

REPORTS.PRG

EDITUPDT.PRG

BACKUP.PRG

APPLICAT.PRG

INSTDEFL.PRG

REINDEX.PRG

FIGURE 5
Program Specification

Program Name: DEMAIN.PRG

Purpose: To check for the existence of the Master dBASE files, to check for the Installation default file and to call the appropriate programs from the main menu.

Input\Output Files:

INSTALL.MEM

dBASE Programs Called (See Program Flow Diagram):

The following programs called are subsystems by themselves. For more information on each program, refer to the appropriate chapter of this manual.

INQUIRY.PRG
REPORTS.PRG
EDITUPDT.PRG
BACKUP.PRG
APPLICAT.PRG
INSTDEFL.PRG
REINDEX.PRG
ERROR.PRG - This program is called only when an error occurs
The following screen is produced by DBMAIN.PRG.

```
TEXAS FLEXIBLE PAVEMENT DATABASE
MAIN MENU

1 - Inquiry
2 - Reports
3 - Edit & Update
4 - Applications
5 - Backup
6 - Installation
7 - Reindex Master Files

Q - QUIT

OPTION ===>  
```
PROGRAM LISTING

* MAIN PROGRAM DRIVER
* PROGRAM NAME: DBMAIN.PRG 06/01/88
* MODIFIED ON: 09/19/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TII
* AUTHOR: TREvor X. PEREIRA
* PURPOSE: TO SET UP THE MAIN MENU FOR THE DATABASE
* MAINTAINENCE SYSTEM.
*
SET TALK OFF
SET ECHO OFF
SET STAT OFF
SET BELL OFF
SET HELP OFF
SET CONFIRM ON
SET ESCAPE OFF
SET SAFETY ON
SET TYPE TO 20
ON ERROR DO \PAVEDB\ERROR.PRG
CLOSE ALL
PUBLIC MDRIVE, MDRIVE2, MPRT1, MPRT2
SET FORMAT TO
STORE " " TO MPTION
STORE .T. TO NOTFINI
CLEAR
* Get drive that is being used
IF FILE("\PAVEDB\INSTALL.MEM")
   RESTORE FROM \PAVEDB\INSTALL ADDI
   MDRIVE = IDRIVE
   MDRIVE2 = IDRIVE2
   MPRT1 = IPRT1
   MPRT2 = IPRT2
   RELE ALL LIKE I*
   SET DEFAULT TO &MDRIVE
ELSE
   CLEAR
   @ 10, 5 SAY "INSTALL.MEM File not found. Please go through the "
   @ 11, 5 SAY " Installation Menu and choose the defaults."
   WAIT
   IDRIVE = " "
   IDRIVE2 = " "
   IPRT1 = " "
   IPRT2 = " "
   DO \PAVEDB\INSIDEFL
ENDIF

? "Checking to see if Master files are present. Please Wait . . ."
IF .NOT. FILE("\PAVEDB\FILES\LOCATION.DBF")
   ? "LOCATION file is not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE('PAVEDB\FILES\LAYER.DBF')
    ?"LAYER file is not found. Please Check . . ."
    WAIT
    RETURN
ENDIF
IF .NOT. FILE('PAVEDB\FILES\LAYTHICK.DBF')
    ?"LAYTHICK file is not found. Please Check . . ."
    WAIT
    RETURN
ENDIF
IF .NOT. FILE('PAVEDB\FILES\GEOSH0.DBF')
    ?"GEOSH0 file is not found. Please Check . . ."
    WAIT
    RETURN
ENDIF
IF .NOT. FILE('PAVEDB\FILES\SURFACE.DBF')
    ?"SURFACE file is not found. Please Check . . ."
    WAIT
    RETURN
ENDIF
IF .NOT. FILE('PAVEDB\FILES\SUBGRADE.DBF')
    ?"SUBGRADE file is not found. Please Check . . ."
    WAIT
    RETURN
ENDIF
IF .NOT. FILE('PAVEDB\FILES\VISUAL.DBF')
    ?"VISUAL file is not found. Please Check . . ."
    WAIT
    RETURN
ENDIF
IF .NOT. FILE('PAVEDB\FILES\DYNAILD.DBF')
    ?"DYNAILD file is not found. Please Check . . ."
    WAIT
    RETURN
ENDIF
IF .NOT. FILE('PAVEDB\FILES\SI.DBF')
    ?"SI file is not found. Please Check . . ."
    WAIT
    RETURN
ENDIF
IF .NOT. FILE('PAVEDB\FILES\TRAFFIC.DBF')
    ?"TRAFFIC file is not found. Please Check . . ."
    WAIT
    RETURN
ENDIF
IF .NOT. FILE('PAVEDB\FILES\WEATHER.DBF')
    ?"WEATHER file is not found. Please Check . . ."
    WAIT
    RETURN
ENDIF
IF .NOT. FILE('PAVEDB\FILES\FALLWGT.DBF')
? "FALLWGT file is not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE('PAVEDB\FILES\ENV.DBF')
? "ENV file is not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE('PAVEDB\FILES\SKID.DBF')
? "SKID file is not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE('PAVEDB\INDEXES\LOCSID.NDX')
? "LOCATION INDEX is not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE('PAVEDB\INDEXES\LAYNDX.NDX')
? "LAYER INDEX is not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE('PAVEDB\INDEXES\LAYTHICK.NDX')
? "LAYTHICK INDEX is not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE('PAVEDB\INDEXES\GEOSHO.NDX')
? "GEOSHO INDEX is not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE('PAVEDB\INDEXES\SURFNDX.NDX')
? "SURFACE INDEX is not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE('PAVEDB\INDEXES\SUBGNDX.NDX')
? "SUBGRADE INDEX is not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE('PAVEDB\INDEXES\VISUAL.NDX')
? "VISUAL INDEX is not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE('PAVEDB\INDEXES\DYNAPL.LD.NDX')
? "DYNAPL.LD INDEX is not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE('PAVEDEI\INDEXES\SI.NDX')
  ? "SI INDEX is not found. Please Check . . ."
  WAIT
  RETURN
ENDIF
IF .NOT. FILE('PAVEDEI\INDEXES\TRAFFIC.NDX')
  ? "TRAFFIC INDEX is not found. Please Check . . ."
  WAIT
  RETURN
ENDIF
IF .NOT. FILE('PAVEDEI\INDEXES\WEATHER.NDX')
  ? "WEATHER INDEX is not found. Please Check . . ."
  WAIT
  RETURN
ENDIF
IF .NOT. FILE('PAVEDEI\INDEXES\FALLWEIGHT.NDX')
  ? "FALLWEIGHT INDEX is not found. Please Check . . ."
  WAIT
  RETURN
ENDIF
IF .NOT. FILE('PAVEDEI\INDEXES\ENV.NDX')
  ? "ENV INDEX is not found. Please Check . . ."
  WAIT
  RETURN
ENDIF
IF .NOT. FILE('PAVEDEI\INDEXES\SKID.NDX')
  ? "SKID INDEX is not found. Please Check . . ."
  WAIT
  RETURN
ENDIF

DO WHILE NOTFINI
  STORE " " TO MOPTION
  CLOSE ALL
  * set up the database maintenance main menu screen and do the
  * loop until MOPTION is 1-7 or Q
  DO WHILE .NOT. (MOPTION $ '1234567Q')
    @ 0, 0 CLEAR
    @ 3, 15 SAY " TEXAS FLEXIBLE PAVEMENT DATABASE"
    @ 4, 32 SAY "MAIN MENU"
    @ 7, 19 SAY "1 - Inquiry"
    @ 8, 19 SAY "2 - Reports"
    @ 9, 19 SAY "3 - Edit & Update"
    @ 10, 19 SAY "4 - Applications"
    @ 11, 19 SAY "5 - Backup"
    @ 12, 19 SAY "6 - Installation"
    @ 13, 19 SAY "7 - Reindex Master Files"
    @ 15, 19 SAY "Q - QUIT"
    @ 17, 44 SAY "OPTION ———> " GET MOPTION PICTURE "+"
    @ 2, 9 TO 19, 67 DOUBLE
    READ
    * check for MOPTION equal to 1 - 7 or Q
    CLEAR TYPE
    IF .NOT. (MOPTION $ '1234567Q')
@ 20, 10 SAY "Please enter 1, 2, 3, 4, 5, 6, 7 or Q"
    WAIT
    ENDF
    ENDDO
    DO CASE
        CASE MOPTION = "1"
            DO \PAVEDB\INQUIRY\INQUIRY
        CASE MOPTION = "2"
            DO \PAVEDB\REPORTS\REPORTS
        CASE MOPTION = "3"
            DO \PAVEDB\EDITUPDT\EDITUPDT
        CASE MOPTION = "4"
            DO \PAVEDB\APPLICAT\APPLICAT
        CASE MOPTION = "5"
            DO \PAVEDB\BACKUP\BACKUP
        CASE MOPTION = "6"
            DO \PAVEDB\INSTDEFI
        CASE MOPTION = "7"
            MRESPONSE = " "
            @ 10,10 SAY "You are about to REINDEX all the files - takes about 2 hours."
            @ 11,11 SAY "Want to continue (Y/N)? " GET MRESPONSE
            READ
            IF MRESPONSE = "Y"
                DO \PAVEDB\REINDEX
            ENDF
        CASE MOPTION = "Q"
            QUIT
    ENDCASE
    ENDDO
* PROGRAM LISTING

* PROGRAM NAME: ERROR.PRG 07/03/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO GIVE A MESSAGE IF AN ERROR OCCURS ANY WHERE IN THE SYSTEM
*

CLEAR TYPE
SET TYPE TO 0
CLEAR
CLOSE ALL
@ 10, 10 SAY " An error has occurred in the system."
@ 12, 10 SAY " Please contact The Texas Transportation Institute.
" at (409) 845-8408"
@ 14, 10 SAY " Ask for: Rebecca Yette or" " Trevor Pereira"
@ 15, 10 SAY ""
@ 16, 10 SAY ""
CANCEL
CHAPTER 2

INQUIRY SUBSYSTEM
GENERAL NARRATIVE

The Inquiry Menu System which is written in dBase III Plus allows the user to view the data in any of the 14 master files and the 7 tables. Changes to the data cannot be made using this part of the system. The Inquiry main menu has the choices Inventory, Monitoring, Traffic, Environment and Tables available. Depending on the file chosen, a dBase form is displayed on the screen with the required information for a particular SID Number. The user can scan through subsequent records or exit the screen.

The program INQUIRY.PRG displays the appropriate menus on the screen and the user chooses the file and SID number he wants to view. The program then uses the appropriate dBase screen format file (form) to display the data. dBase screen format files use the .FMT extension as a naming convention. INQUIRY.PRG and all .FMT files are stored in the subdirectory \PAVEDB\INQUIRY.

This chapter contains the following information about the Inquiry Subsystem:

- Program Flow Diagram
- Program Specification
- Inquiry Screens
- Screen Format Program Listings
- Program Listings.
**Program Specification**

**Program Name:** INQUIRY.PRG

**Purpose:** To display the data to the screen for the appropriate files and tables on user's request.

**Input Files:** The following files are used along with their indices:

<table>
<thead>
<tr>
<th>Data File</th>
<th>Index File</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCATION.DBF</td>
<td>LOCSID.NDX</td>
</tr>
<tr>
<td>LAYER.DBF</td>
<td>LAYNDX.NDX</td>
</tr>
<tr>
<td>GEOSHO.DBF</td>
<td>GEONDX.NDX</td>
</tr>
<tr>
<td>SURFACE.DBF</td>
<td>SURFNDX.NDX</td>
</tr>
<tr>
<td>SUBGRADE.DBF</td>
<td>SUBGNX.NDX</td>
</tr>
<tr>
<td>LAYTHICK.DBF</td>
<td>LAYTNX.NDX</td>
</tr>
<tr>
<td>VISUAL.DBF</td>
<td>VIUSD.NDX</td>
</tr>
<tr>
<td>SI.DBF</td>
<td>SI.NDX</td>
</tr>
<tr>
<td>FALLWGT.DBF</td>
<td>FALLWGT.NDX</td>
</tr>
<tr>
<td>DYNACLID.DBF</td>
<td>DYNACLID.NDX</td>
</tr>
<tr>
<td>SKID.DBF</td>
<td>SKID.NDX</td>
</tr>
<tr>
<td>ENV.DBF</td>
<td>ENV.NDX</td>
</tr>
<tr>
<td>WEATHER.DBF</td>
<td>WEATHER.NDX</td>
</tr>
<tr>
<td>TRAFFIC.DBF</td>
<td>TRAFFIC.NDX</td>
</tr>
<tr>
<td>CTYTBLSN.DBF</td>
<td>CTYTBLSN.NDX</td>
</tr>
<tr>
<td>MATLTLB.DBF</td>
<td>MATLTLB.NDX</td>
</tr>
<tr>
<td>PAVETYPD.DBF</td>
<td>PAVETYPD.NDX</td>
</tr>
<tr>
<td>DISTTTEMP.DBF</td>
<td>DISTTTEMP.NDX</td>
</tr>
<tr>
<td>WIDENFLG.DBF</td>
<td>WIDENFLG.NDX</td>
</tr>
<tr>
<td>LAVERTLB.DBF</td>
<td>LAVERTLB.NDX</td>
</tr>
<tr>
<td>FUNCLTLB.DBF</td>
<td>FUNCLTLB.NDX</td>
</tr>
</tbody>
</table>

**dBase Procedure File Called:**

INQ COLL.PRG - This procedure displays on the screen data for individual files.
PROGRAM SPECIFICATION (continued)

Output (Using dBASE III Format Files):

The following format files (forms) are used to display data to the screen:

<table>
<thead>
<tr>
<th>Format File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INQ LOCN.FMT</td>
<td>Location</td>
</tr>
<tr>
<td>INQ LAYR.FMT</td>
<td>Layer Identification</td>
</tr>
<tr>
<td>INQ LAYT.FMT</td>
<td>Layer Thickness</td>
</tr>
<tr>
<td>INQ GEO0.FMT</td>
<td>Geometric &amp; Shoulder</td>
</tr>
<tr>
<td>INQ SURF.FMT</td>
<td>Surface</td>
</tr>
<tr>
<td>INQ SUBG.FMT</td>
<td>Subgrade</td>
</tr>
<tr>
<td>INQ VISL.FMT</td>
<td>Visual</td>
</tr>
<tr>
<td>INQ SI.FMT</td>
<td>Serviceability Index</td>
</tr>
<tr>
<td>INQ FALL.FMT</td>
<td>Falling Weight</td>
</tr>
<tr>
<td>INQ_DYNA.FMT</td>
<td>Dynaflect</td>
</tr>
<tr>
<td>INQ SKID.FMT</td>
<td>Skid</td>
</tr>
<tr>
<td>INQ ENV.FMT</td>
<td>Environment</td>
</tr>
<tr>
<td>INQ WEAT.FMT</td>
<td>Weather</td>
</tr>
<tr>
<td>INQ TRAF.FMT</td>
<td>Traffic</td>
</tr>
<tr>
<td>INQ_CNTY.FMT</td>
<td>County Name</td>
</tr>
<tr>
<td>INQ_MATL.FMT</td>
<td>Material Type Class</td>
</tr>
<tr>
<td>INQ_FVMT.FMT</td>
<td>Type of Pavement</td>
</tr>
<tr>
<td>INQ_DIST.FMT</td>
<td>District Temperature</td>
</tr>
<tr>
<td>INQ_WIDN.FMT</td>
<td>Widening Flag</td>
</tr>
<tr>
<td>INQ_LITBL.FMT</td>
<td>Layer Description</td>
</tr>
<tr>
<td>INQ_FUNC.FMT</td>
<td>Functional Class</td>
</tr>
</tbody>
</table>
INQUIRY SCREENS

TEXAS FLEXIBLE PAVEMENT DATABASE 1.0
Inquiry

1 - Inventory Data
2 - Monitoring Data
3 - Traffic Data
4 - Environmental Data
5 - Tables

OPTION ➔

Main Menu of the Inquiry Subsystem
INQUIRY SCREENS (continued)

TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
Inventory Data

1 - Location
2 - Layer ID
3 - Geometric & Shoulder
4 - Surface
5 - Subgrade
6 - Layer Thickness Across the Road

OPTION ➔

Inventory Menu Screen

TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY - INVENTORY
Location File

SID Number  13
Highway Ident. US  82
Mile Post  22 + 0  TO 24 + 0
Mile Point  22.000  TO 24.000
HPMS Sample Number
Functional Classification  0
Active  T
Inactive Date  0/0
Previous SID  0
Next SID  0
Number of Lanes  1
HPMS Section Subdivision  0
District  1
County  92
Control/Section  45/4
Lane Identification R
Mile Point Date  6/75

Inventory - Location File Data Screen Display
TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
Inventory Data - Layer Identification

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID Number</td>
<td>13</td>
</tr>
<tr>
<td>Structure Number</td>
<td>1</td>
</tr>
<tr>
<td>Layer Number</td>
<td>1</td>
</tr>
<tr>
<td>Layer Description</td>
<td>7</td>
</tr>
<tr>
<td>Center Thickness (Inches)</td>
<td></td>
</tr>
<tr>
<td>Layer Material Classification</td>
<td>44</td>
</tr>
<tr>
<td>Job Completed Date (MM/YY)</td>
<td>6/31</td>
</tr>
<tr>
<td>Widening Date (MM/YY)</td>
<td>0/0</td>
</tr>
</tbody>
</table>

Inventory - Layer File Data Screen Display

TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
Inventory Data - Geometric And Shoulder

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID Number</td>
<td>13</td>
</tr>
<tr>
<td>Structure Number</td>
<td>1</td>
</tr>
<tr>
<td>Pavement Type</td>
<td>34</td>
</tr>
<tr>
<td>Lane Width one way - (ft)</td>
<td>12.0</td>
</tr>
<tr>
<td>Outside Shoulder Width (ft)</td>
<td>0.0</td>
</tr>
<tr>
<td>Shoulder Surface Type</td>
<td>1</td>
</tr>
<tr>
<td>Shoulder Base Type</td>
<td>0</td>
</tr>
<tr>
<td>Shoulder Surface Thick (in.)</td>
<td>0.00</td>
</tr>
<tr>
<td>Shoulder Base Thick (in.)</td>
<td>0.00</td>
</tr>
<tr>
<td>Widening Flag</td>
<td>1</td>
</tr>
</tbody>
</table>

Inventory - Geometric & Shoulder File Data Screen Display
TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
Inventory Data - Surface

SID Number 13
Structure Number 1
Layer Number 3

Aggregate Application Rate (S.Y./C.Y.) 0
Admixture Type AC
Admixture Percent (%) 5.70
Asphalt Application Rate (Gal/S.Y.) 0.00

Inventory - Surface File Data Screen Display

TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
Inventory Data - Subgrade

SID Number 13
Structure Number 1
Layer Number 1

Percent Passing No. 200 Sieve (%) 88.8
Plasticity Index 40.4
Liquid Limit 64.5
Texas Triaxial Class 5.3
Permeability Index 0.23

Inventory - Subgrade File Data Screen Display
**TEXAS FLEXIBLE PAVEMENT DATABASE INQUIRY**

**Inventory Data - Layer Thickness Across the Road**

<table>
<thead>
<tr>
<th>SID Number</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure Number</td>
<td>1</td>
</tr>
<tr>
<td>Layer Number</td>
<td>2</td>
</tr>
</tbody>
</table>

| Thickness - 3rd Pos From Center (in.) | 10.00 |
| Thickness - 2nd Pos From Center (in.) | 10.00 |
| Thickness - 1st Pos From Center (in.) | 6.00  |
| Thickness - Center (in.)              | 6.00  |
| Distance From Center - 3rd Position (ft.) | 12.00 |
| Distance From Center - 2nd Position (ft.) | 9.0   |
| Distance From Center - 1st Position (ft.) | 5.0   |

Inventory - Layer Thickness Across The Road File Data Screen Display
INQUIRY SCREENS (continued)

TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
Monitoring Data

1 - Visual
2 - Serviceability Index
3 - Falling Weight
4 - Dynaffect
5 - Skid

OPTION =>

Enter SID Number => 0

Monitoring Menu Screen

TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
Monitoring Data - Visual Rating File

Actual Date of Measurement  9/80
SID Number  13
Structure Number  2
Layer Number  5

Rutting  Block Cr  Alligtr Cr  Longitud Cr  Transv Cr
1S 0M OSV  0S 0M OSV  0S 0M OSV  1S 0M OSV  0S 1M OSV

Seal Code  Patching  Failures/Mi  Pavement Rat Scr  78
1  0G 0F 0P  0  PES Pavement Rat Scr 0.00

Unwght Vis. Rat Scr 0.00

Monitoring - Visual File Data Screen Display
### TEXAS FLEXIBLE PAVEMENT DATABASE
### INQUIRY
### Monitoring Data - Serviceability Index

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID Number</td>
<td>13</td>
</tr>
<tr>
<td>Structure Number</td>
<td>2</td>
</tr>
<tr>
<td>Layer Number</td>
<td>5</td>
</tr>
<tr>
<td>Date</td>
<td>9/16/80</td>
</tr>
<tr>
<td>Count of Observation</td>
<td>8</td>
</tr>
<tr>
<td>Mean</td>
<td>3.17500</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.38079</td>
</tr>
<tr>
<td>Low Value</td>
<td>2.5</td>
</tr>
<tr>
<td>High Value</td>
<td>3.6</td>
</tr>
</tbody>
</table>

### Monitoring - Serviceability Index File Data Screen Display

### TEXAS FLEXIBLE PAVEMENT DATABASE
### INQUIRY
### Monitoring Data - Falling Weight SSI

<table>
<thead>
<tr>
<th>Date</th>
<th>SID Number</th>
<th>Structure Number</th>
<th>Layer Number</th>
<th>Average SSI</th>
<th>Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/0/0</td>
<td>26</td>
<td>1</td>
<td>5</td>
<td>0.0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Geophone</th>
<th>Reading 1</th>
<th>Reading 2</th>
<th>Reading 3</th>
<th>Reading 4</th>
<th>Reading 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### Monitoring - Falling Weight SSI File Data Screen Display
INQUIRY SCREENS (continued)

TEXAS FLEXIBLE PAVEMENT DATABASE INQUIRY
Monitoring Data - Dynaflect Measurement

| SID Number | 13 |
| Structure Number | 1 |
| Layer Number | 4 |
| Date | 8/10/76 |
| STATION | 1 |
| Reading for Sensor 1 | 0.570 |
| Reading for Sensor 2 | 0.540 |
| Reading for Sensor 3 | 0.470 |
| Reading for Sensor 4 | 0.380 |
| Reading for Sensor 5 | 0.320 |

Monitoring - Dynaflect Measurement File Data Screen Display

TEXAS FLEXIBLE PAVEMENT DATABASE INQUIRY
Monitoring Data - Skid Measurement

| SID Number | 13 |
| Structure Number | 1 |
| Layer Number | 4 |
| Date | 4/71 |
| Mean | 38 |
| High | 42 |
| Low | 32 |

Monitoring - Skid Measurement File Data Screen Display
INQUIRY SCREENS (continued)

TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
Traffic Data

Please Enter SID Number: 0

Traffic - Traffic File Screen to Enter the SID number

TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
Traffic Data

<table>
<thead>
<tr>
<th>SID Number</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>1933</td>
</tr>
<tr>
<td>Annual Average Daily Traffic</td>
<td>386</td>
</tr>
<tr>
<td>Annual Cummulative 18 Keal - one way</td>
<td>54259</td>
</tr>
<tr>
<td>Percent trucks</td>
<td>17.7</td>
</tr>
</tbody>
</table>

Traffic - Traffic File Data Screen Display
INQUIRY SCREENS (continued)

TEXAS FLEXIBLE PAVEMENT DATABASE 1.4
INQUIRY
Environmental Data

1 - Environment
2 - Weather

OPTION ———>

Enter county Number ———> 0

Environment Menu Screen
### Texas Flexible Pavement Database Inquiry

#### Weather Measurement

<table>
<thead>
<tr>
<th></th>
<th>No. of Yrs Avg</th>
<th>Mean</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precipitation</td>
<td>19</td>
<td>3.179</td>
<td>2.053</td>
</tr>
<tr>
<td>Total Freeze Thaw Cycle</td>
<td>18</td>
<td>9.111</td>
<td>3.160</td>
</tr>
<tr>
<td>Wet Freeze Thaw Cycle</td>
<td>18</td>
<td>1.889</td>
<td>1.183</td>
</tr>
<tr>
<td>Maximum Temperature</td>
<td>18</td>
<td>55.611</td>
<td>3.791</td>
</tr>
<tr>
<td>Averaged Temperature</td>
<td>18</td>
<td>45.056</td>
<td>3.369</td>
</tr>
</tbody>
</table>

Environment - Weather Measurement File Data Screen Display

### Texas Flexible Pavement Database Inquiry

#### Environment Measurement

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>County Number</td>
<td>1</td>
</tr>
<tr>
<td>Thornthwaite Index Mean</td>
<td>12.510</td>
</tr>
<tr>
<td>Thornthwaite Index</td>
<td></td>
</tr>
<tr>
<td>- No. of Years Averaged</td>
<td>20</td>
</tr>
<tr>
<td>Thornthwaite Index</td>
<td></td>
</tr>
<tr>
<td>- Standard Deviation</td>
<td>26.102</td>
</tr>
</tbody>
</table>

Environment - Environment Measurement File Data Screen Display
INQUIRY SCREENS (continued)

<table>
<thead>
<tr>
<th>TEXAS FLEXIBLE PAVEMENT DATABASE</th>
<th>1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>INQUIRY</td>
<td></td>
</tr>
<tr>
<td>Tables</td>
<td></td>
</tr>
</tbody>
</table>

1 - County Name
2 - Material Type
3 - Type of Pavement
4 - District Temperature Constant
5 - Widening Flag

OPTION ➔

Table Menu Screen
INQUIRY SCREENS (continued)

TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
County Table

County Number 1
County Name ANDERSON

Tables - County Name File Data Screen Display

TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
Material Type Classification Table

Material Code 1
Material Description HOT MIX - HOT LAID
Material Short Form HMAC
Layer Description S

Tables - Material Type Classification File Data Screen Display

TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
Type of Pavement Table

Pavement Code 1
Type of Base GRANULAR BASE
Surface Thickness SURFACE TREATED
Surface Seal

Tables - Type of Pavement File Data Screen Display
INQUIRY SCREENS (continued)

TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
District Temperature Table

District Number  1
Temperature Constant  21

Tables - District Temperature File Data Screen Display

TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
Widening Table

Code    0
Description No Widening
Comments:
        Center Thickness CAN be used

Tables - Widening Flag File Data Screen Display

TEXAS FLEXIBLE PAVEMENT DATABASE
INQUIRY
Layer Description Table

Layer Code    1
Code Description OVL
Four Letter Code Overlay

Tables - Layer Description File Data Screen Display
Format Programs For Screens
Location

@ 2, 23 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 3, 23 SAY "INQUIRY"
@ 4, 23 SAY "Inventory Data - Location File"
@ 6, 6 SAY "SID Number"
@ 6, 18 SAY LOCATION->SID_NO PICTURE "9999"
@ 6, 44 SAY "District"
@ 6, 53 SAY LOCATION->HWYDIST PICTURE "99"
@ 6, 59 SAY "County"
@ 6, 66 SAY LOCATION->CITYNUM PICTURE "999"
@ 8, 6 SAY "Highway Ident."
@ 8, 21 SAY LOCATION->HWYPREFIX
@ 8, 24 SAY LOCATION->HWYNUM PICTURE "9999"
@ 8, 29 SAY LOCATION->HWYSUFFIX
@ 8, 44 SAY "Control/Section"
@ 8, 60 SAY LOCATION->CONTROL PICTURE "9999"
@ 8, 64 SAY "/"
@ 8, 65 SAY LOCATION->SECTION PICTURE "99"
@ 10, 6 SAY "Mile Post"
@ 10, 17 SAY LOCATION->BEGINPST PICTURE "999"
@ 10, 20 SAY LOCATION->EDITOR
@ 10, 21 SAY IF(LEN(LTRIM(STR(EMPSTDIS,2))) = 1,"0" + LTRIM(STR(EMPSTDIS,2)), STR(EMPSTDIS,2))
@ 10, 24 SAY "TO"
@ 10, 27 SAY LOCATION->ENDEMPST PICTURE "999"
@ 10, 30 SAY LOCATION->EDITOR
@ 10, 31 SAY IF(LEN(LTRIM(STR(EMPSTDIS,2))) = 1,"0" + LTRIM(STR(EMPSTDIS,2)), STR(EMPSTDIS,2))
@ 10, 44 SAY "Lane Identification"
@ 10, 64 SAY LOCATION->LANEID
@ 12, 6 SAY "Mile Point"
@ 12, 17 SAY LOCATION->BEGINPNT PICTURE "99.999"
@ 12, 24 SAY "TO"
@ 12, 27 SAY LOCATION->ENDEMPNT PICTURE "99.999"
@ 12, 44 SAY "Mile Point Date"
@ 12, 61 SAY LOCATION->MAXIMPNT PICTURE "99"
@ 12, 63 SAY "/"
@ 12, 64 SAY LOCATION->MINPNT PICTURE "99"
@ 14, 6 SAY "HPMS Sample Number"
@ 14, 25 SAY LOCATION->HPMSSAM
@ 14, 44 SAY "HPMS Section Subdivision"
@ 14, 69 SAY LOCATION->HPMSSEC PICTURE "9"
@ 16, 6 SAY "Functional Classification"
@ 16, 31 SAY LOCATION->FUNCTION PICTURE "99"
@ 16, 44 SAY "Number of Lanes"
@ 16, 61 SAY LOCATION->NUMLANES PICTURE "99"
@ 18, 6 SAY "Active ?"
@ 18, 15 SAY LOCATION->ACTIVEFLAG PICTURE "L"
@ 18, 19 SAY "Inactive Date"
@ 18, 33 SAY LOCATION->INACTIV PICTURE "99"
@ 18, 35  SAY "/"
@ 18, 36  SAY LOCATION->INACTYR PICTURE "99"
@ 18, 41  SAY "Previous SID"
@ 18, 54  SAY LOCATION->PREVSID PICTURE "9999"
@ 18, 61  SAY "Next SID"
@ 18, 70  SAY LOCATION->NEXTSID PICTURE "9999"
@ 20, 6   SAY "Comment"
@ 20, 15  SAY LOCATION->COMMENT PICTURE
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 1, 2 TO 21, 77   DOUBLE
Format Programs For Screens
Layer Identification

@ 3, 25 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 4, 37 SAY "INQUIRY"
@ 5, 22 SAY "Inventory Data - Layer Identification"
@ 8, 37 SAY "SID Number"
@ 8, 55 SAY LAYER->SID NO
@ 9, 37 SAY "Structure Number"
@ 9, 57 SAY LAYER->STRUCNUM
@ 10, 37 SAY "Layer Number"
@ 10, 57 SAY LAYER->LAYNUM
@ 12, 23 SAY "Layer Description"
@ 12, 57 SAY LAYER->LAYDESC FUNCTION "Z"
@ 13, 23 SAY "Center Thickness (inches)"
@ 13, 54 SAY LAYER->CENTHKL FUNCTION "Z"
@ 14, 23 SAY "Layer Material Classification"
@ 14, 57 SAY LAYER->LAYMATCL FUNCTION "Z"
@ 15, 23 SAY "Job Completed Date (MM/YY)"
@ 15, 54 SAY LAYER->JOCMPMO FUNCTION "Z"
@ 15, 56 SAY "/"
@ 15, 57 SAY LAYER->JOCMPYR FUNCTION "Z"
@ 16, 23 SAY "Widening Date (MM/YY)"
@ 16, 54 SAY LAYER->WIDENLMO
@ 16, 56 SAY "/"
@ 16, 57 SAY LAYER->WIDENLYR FUNCTION "B"
@ 2, 15 TO 18, 65 DOUBLE
Format Programs For Screens
Layer Thickness Across The Road

@ 3, 24 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 4, 35 SAY "INQUIRY"
@ 5, 15 SAY "Inventory Data - Layer Thickness Across The Road"
@ 8, 36 SAY "SID Number"
@ 9, 36 SAY "SID Number"
@ 8, 58 SAY LAYTHICK->SID NO
@ 9, 36 SAY "Structure Number"
@ 9, 60 SAY LAYTHICK->STRUCNUM
@ 10, 36 SAY "Layer Number"
@ 10, 60 SAY LAYTHICK->LAYNUM
@ 12, 14 SAY "Thickness - 3rd Pos. From Center (in.)"
@ 12, 57 SAY LAYTHICK->FC3THK
@ 13, 14 SAY "Thickness - 2nd Pos. From Center (in.)"
@ 13, 57 SAY LAYTHICK->FC2THK
@ 14, 14 SAY "Thickness - 1st Pos. From Center (in.)"
@ 14, 57 SAY LAYTHICK->FC1THK
@ 15, 14 SAY "Thickness - Center (in.)"
@ 15, 57 SAY LAYTHICK->CENTTHK
@ 16, 14 SAY "Distance From Center - 3rd Position (ft)"
@ 16, 58 SAY LAYTHICK->FC3DIS
@ 17, 14 SAY "Distance From Center - 2nd Position (ft)"
@ 17, 58 SAY LAYTHICK->FC2DIS
@ 18, 14 SAY "Distance From Center - 1st Position (ft)"
@ 18, 58 SAY LAYTHICK->FC1DIS
@ 2, 9 TO 20, 68 DOUBLE
Format Programs For Screens
Geometric & Shoulder Information

3, 24 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
4, 35 SAY "INQUIRY"
5, 20 SAY "Inventory Data - Geometric And Shoulder"
8, 36 SAY "SID Number"
8, 54 SAY GEOSHO->SID_NO
9, 36 SAY "Structure Number"
9, 56 SAY GEOSHO->STRUCNUM
11, 20 SAY "Pavement Type"
11, 52 SAY GEOSHO->PAVETYP
12, 20 SAY "Lane Width - one way (ft)"
12, 50 SAY GEOSHO->LANEWID
13, 20 SAY "Outside Shoulder Width (ft)"
13, 50 SAY GEOSHO->OUTSHOWD
14, 20 SAY "Shoulder Surface Type"
14, 53 SAY GEOSHO->SHOSFTYP
15, 20 SAY "Shoulder Base Type"
15, 52 SAY GEOSHO->SHOBSFTYP
16, 20 SAY "Shoulder Surface Thick (in.)"
16, 49 SAY GEOSHO->SHOSFTHK
17, 20 SAY "Shoulder Base Thick (in.)"
17, 49 SAY GEOSHO->SHOBSTHK
18, 20 SAY "Widening Flag"
18, 53 SAY GEOSHO->WIDENFLG
2, 14 TO 20, 64 DOUBLE
@ 4, 23 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 5, 35 SAY "INQUIRY"
@ 6, 26 SAY "Inventory Data - Surface"
@ 9, 36 SAY "SID Number"
@ 9, 54 SAY SURFACE->SID NO
@ 10, 36 SAY "Structure Number"
@ 10, 56 SAY SURFACE->STRUCNUM
@ 11, 36 SAY "Layer Number"
@ 11, 56 SAY SURFACE->LAYNUM
@ 13, 20 SAY "Aggregate Appl. Rate (S.Y./C.Y.)"
@ 13, 55 SAY SURFACE->AGAPPLRT PICTURE "999"
@ 14, 20 SAY "Admixture Type"
@ 14, 46 SAY SURFACE->ADMXTYP
@ 15, 20 SAY "Admixture Percent (%)"
@ 15, 53 SAY SURFACE->ADMXPCT
@ 16, 20 SAY "Asphalt Appl. Rate (Gal/S.Y.)"
@ 16, 54 SAY SURFACE->ASAPPLRT
@ 3, 14 TO 18, 64 DOUBLE
@ 4, 24 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 5, 35 SAY "INQUIRY"
@ 6, 27 SAY "Inventory Data - Subgrade"
@ 9, 36 SAY "SID Number"
@ 9, 54 SAY SUBGRADE->SID_NO
@ 10, 36 SAY "Structure Number"
@ 10, 56 SAY SUBGRADE->STRUCNUM
@ 11, 36 SAY "Layer Number"
@ 11, 56 SAY SUBGRADE->LAYNUM
@ 13, 20 SAY "Percent Passing No. 200 Sieve (%)"
@ 13, 54 SAY SUBGRADE->PPSV200
@ 14, 20 SAY "Plasticity Index"
@ 14, 54 SAY SUBGRADE->PLASTIX
@ 15, 20 SAY "Liquid Limit"
@ 15, 54 SAY SUBGRADE->LIQLIM
@ 16, 20 SAY "Texas Triaxial Class"
@ 16, 55 SAY SUBGRADE->TXTRIAXL
@ 17, 20 SAY "Permeability Index"
@ 17, 53 SAY SUBGRADE->PERMIX
@ 3, 14 TO 19, 64 DOUBLE
Format Programs For Screens
Visual Rating

@ 2, 24 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 3, 35 SAY "INQUIRY"
@ 4, 22 SAY "Monitoring Data - Visual Rating File"
@ 7, 7 SAY "Actual Date of Measurement"
@ 7, 34 SAY VISUAL->ACTIMONTH PICTURE "99"
@ 7, 36 SAY "/"
@ 7, 37 SAY VISUAL->ACTYEAR
@ 7, 45 SAY "SID Number"
@ 7, 63 SAY VISUAL->SID NO
@ 8, 45 SAY "Structure Number"
@ 8, 65 SAY VISUAL->STRUCNUM
@ 9, 45 SAY "Layer Number"
@ 9, 65 SAY VISUAL->LAYNUM
@ 12, 7 SAY "Rutting          Block Cr  Alligtr Cr  Longitud Cr  Transv Cr"
@ 13, 7 SAY VISUAL->RUTTLS
@ 13, 8 SAY "S"
@ 13, 10 SAY VISUAL->RUTIMD
@ 13, 11 SAY "M"
@ 13, 13 SAY VISUAL->RUTTSV
@ 13, 14 SAY "S"
@ 13, 20 SAY VISUAL->BLKCRKSL
@ 13, 21 SAY "S"
@ 13, 23 SAY VISUAL->BLKCRKMD
@ 13, 24 SAY "M"
@ 13, 26 SAY VISUAL->BLKCRKSV
@ 13, 27 SAY "S"
@ 13, 33 SAY VISUAL->ALLGCRSL
@ 13, 34 SAY "S"
@ 13, 37 SAY VISUAL->ALLGCRMD
@ 13, 38 SAY "M"
@ 13, 41 SAY VISUAL->ALLGCRSV
@ 13, 42 SAY "S"
@ 13, 47 SAY VISUAL->LONGCRSL
@ 13, 48 SAY "S"
@ 13, 51 SAY VISUAL->LONGCRMD
@ 13, 52 SAY "M"
@ 13, 55 SAY VISUAL->LONGCRSV
@ 13, 56 SAY "S"
@ 13, 61 SAY VISUAL->TRANCRSL
@ 13, 62 SAY "S"
@ 13, 64 SAY VISUAL->TRANCRMD
@ 13, 65 SAY "M"
@ 13, 67 SAY VISUAL->TRANCRSV
@ 13, 68 SAY "S"
@ 16, 7 SAY "Seal Code Patching Failures/Mi Pavement Rat Scr"
@ 16, 67 SAY VISUAL->PRS
@ 17, 11 SAY VISUAL->SEALCRCD
@ 17, 18 SAY VISUAL->PATCHGD
@ 17, 19 SAY "G"
@ 17, 21 SAY VISUAL->PATCHFR
@ 17, 22 SAY "F"
@ 17, 24 SAY VISUAL->PATCHFR
@ 17, 25 SAY "P"
@ 17, 35 SAY VISUAL->FAILMILE
@ 17, 45 SAY "PES Pavement Rat Scr"
@ 17, 66 SAY VISUAL->PESPVTRES
@ 18, 46 SAY "Unwght Vis. Rat Scr"
@ 18, 66 SAY VISUAL->UVURS
@ 1, 2 TO 20, 73 DOUBLE
Format Programs For Screens
Serviceability Index

0 3, 24 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
0 4, 35 SAY "INQUIRY"
0 5, 21 SAY "Monitoring Data - Serviceability Index"
0 7, 36 SAY "SID Number"
0 7, 54 SAY SI->SID_NO
0 8, 36 SAY "Structure Number"
0 8, 56 SAY SI->STRNUM
0 9, 36 SAY "Layer Number"
0 9, 56 SAY SI->LAYNUM
0 10, 36 SAY "Date"
0 10, 50 SAY SI->ACTMONTH FUNCTION "Z"
0 10, 52 SAY "/"
0 10, 53 SAY SI->ACTDAY FUNCTION "Z"
0 10, 55 SAY "/"
0 10, 56 SAY SI->ACTYEAR FUNCTION "Z"
0 12, 22 SAY "Count of Observation"
0 12, 55 SAY SI->SICOUNT
0 13, 22 SAY "Mean"
0 13, 51 SAY SI->SIMEAN
0 14, 22 SAY "Standard Deviation"
0 14, 51 SAY SI->SISD
0 15, 22 SAY "Low Value"
0 15, 55 SAY SI->SILOWVAL
0 16, 22 SAY "High Value"
0 16, 55 SAY SI->SIHIVAL
0 2, 14 TO 18, 64  DOUBLE
Format Programs For Screens
Falling Weight SSI

@ 2, 24 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 3, 35 SAY "INQUIRY"
@ 4, 21 SAY "Monitoring Data - Falling Weight SSI"
@ 7,  6 SAY "Date"
@ 7, 12 SAY FALLWGT->MONTH
@ 7, 14 SAY "/"
@ 7, 15 SAY FALLWGT->DAY
@ 7, 17 SAY "/"
@ 7, 18 SAY FALLWGT->YEAR
@ 7, 25 SAY "SID Number"
@ 7, 43 SAY FALLWGT->SID_NO
@ 7, 57 SAY "Average SSI"
@ 7, 70 SAY FALLWGT->RSSSTIAVG
@ 8, 11 SAY "(MM/DD/YY) Structure Number"
@ 8, 45 SAY FALLWGT->STRUCTNUM
@ 8, 57 SAY "Temperature"
@ 8, 71 SAY FALLWGT->SSITEMP
@ 9, 25 SAY "Layer Number"
@ 9, 45 SAY FALLWGT->LAYNUM
@ 11, 18 SAY "Reading 1  Reading 2  Reading 3  Reading 4  Reading 5"
@ 12,  5 SAY "Geophone 1"
@ 12, 20 SAY FALLWGT->SSIGP11
@ 12, 32 SAY FALLWGT->SSIGP21 PICTURE "99.99"
@ 12, 44 SAY FALLWGT->SSIGP31
@ 12, 56 SAY FALLWGT->SSIGP41
@ 12, 68 SAY FALLWGT->SSIGP51
@ 13,  5 SAY "Geophone 2"
@ 13, 20 SAY FALLWGT->SSIGP12
@ 13, 32 SAY FALLWGT->SSIGP22
@ 13, 44 SAY FALLWGT->SSIGP32
@ 13, 56 SAY FALLWGT->SSIGP42
@ 13, 68 SAY FALLWGT->SSIGP52
@ 14,  5 SAY "Geophone 3"
@ 14, 20 SAY FALLWGT->SSIGP13
@ 14, 32 SAY FALLWGT->SSIGP23 PICTURE "99.99"
@ 14, 44 SAY FALLWGT->SSIGP33
@ 14, 56 SAY FALLWGT->SSIGP43
@ 14, 68 SAY FALLWGT->SSIGP53
@ 15,  5 SAY "Geophone 4"
@ 15, 20 SAY FALLWGT->SSIGP14
@ 15, 32 SAY FALLWGT->SSIGP24
@ 15, 44 SAY FALLWGT->SSIGP34
@ 15, 56 SAY FALLWGT->SSIGP44
@ 15, 68 SAY FALLWGT->SSIGP54
@ 16,  5 SAY "Geophone 5"
@ 16, 20 SAY FALLWGT->SSIGP15
@ 16, 32 SAY FALLWGT->SSIGP25
@ 16, 44 SAY FALLWGT->SSIGP35
@ 16, 56 SAY FALLWGT->SSIGP45
@ 16, 68 SAY FALLWGT->SSIGP55
@ 17,  5 SAY "Geophone 6"
@ 17, 20 SAY FALLWGT->SSIGP16
@ 17, 32 SAY FALLWGT->SSIGP26
@ 17, 44 SAY FALLWGT->SSIGP36
@ 17, 56 SAY FALLWGT->SSIGP46
@ 17, 68 SAY FALLWGT->SSIGP56
@ 18,  5 SAY "Geophone 7"
@ 18, 20 SAY FALLWGT->SSIGP17
@ 18, 32 SAY FALLWGT->SSIGP27
@ 18, 44 SAY FALLWGT->SSIGP37
@ 18, 56 SAY FALLWGT->SSIGP47
@ 18, 68 SAY FALLWGT->SSIGP57
@  1,  2 TO 20, 77    DOUBLE
Format Programs For Screens
Dynaflect Measurement

3, 24 SAY "TENAS FLEXIBLE PAVEMENT DATABASE"
4, 35 SAY "INQUIRY"
5, 20 SAY "Monitoring Data - Dynaflect Measurement"
7, 38 SAY "SID Number"
7, 55 SAY DYNADFLD->STD_NO
8, 38 SAY "Structure Number"
8, 57 SAY DYNADFLD->STRUCNUM
9, 38 SAY "Layer Number"
9, 57 SAY DYNADFLD->LAYNUM
10, 38 SAY "Date"
10, 51 SAY DYNADFLD->MONTH
10, 53 SAY "/
10, 54 SAY DYNADFLD->DAY
10, 56 SAY "/
10, 57 SAY DYNADFLD->YEAR
12, 21 SAY "STATION"
12, 53 SAY DYNADFLD->STATION
13, 21 SAY "Reading for Sensor 1"
13, 50 SAY DYNADFLD->SENS1RD
14, 21 SAY "Reading for Sensor 2"
14, 50 SAY DYNADFLD->SENS2RD
15, 21 SAY "Reading for Sensor 3"
15, 50 SAY DYNADFLD->SENS3RD
16, 21 SAY "Reading for Sensor 4"
16, 50 SAY DYNADFLD->SENS4RD
17, 21 SAY "Reading for Sensor 5"
17, 50 SAY DYNADFLD->SENS5RD
2, 14 TO 19, 64   DOUBLE
Format Programs For Screens
Skid

@ 3, 24 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 4, 35 SAY "INQUIRY"
@ 5, 23 SAY "Monitoring Data - Skid Measurement"
@ 8, 34 SAY "SID Number"
@ 8, 52 SAY SKID->SID_NO
@ 9, 34 SAY "Structure Number"
@ 9, 54 SAY SKID->STRUCTNUM
@ 10, 34 SAY "Layer Number"
@ 10, 54 SAY SKID->LAYNUM
@ 11, 34 SAY "Date"
@ 11, 51 SAY SKID->MONTH
@ 11, 53 SAY "/"
@ 11, 54 SAY SKID->YEAR
@ 13, 29 SAY "Mean"
@ 13, 41 SAY SKID->SKIDNUMM
@ 14, 29 SAY "High"
@ 14, 41 SAY SKID->SKIDNUMH
@ 15, 29 SAY "Low"
@ 15, 41 SAY SKID->SKIDNUML
@ 2, 16 TO 17, 63 DOUBLE
Print Program for Screens
Traffic

@ 3, 24 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 4, 35 SAY "INQUIRY"
@ 5, 32 SAY "Traffic Data"
@ 7, 39 SAY "SID Number"
@ 7, 50 SAY TRAFFIC->SID_NO
@ 8, 39 SAY "Year"
@ 8, 50 SAY TRAFFIC->YEAR
@ 10, 21 SAY "Annual Average Daily Traffic"
@ 10, 53 SAY TRAFFIC->AADT1WAY
@ 11, 21 SAY "Annual Cumulative 18 Keal"
@ 12, 28 SAY "- one way"
@ 12, 50 SAY TRAFFIC->A18KEAL
@ 13, 21 SAY "Percent trucks"
@ 13, 55 SAY TRAFFIC->PCTTRK
@ 2, 14 TO 15, 65 DOUBLE
Format Programs For Screens
Weather

@ 4, 24 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 5, 35 SAY "INQUIRY"
@ 6, 29 SAY "Weather Measurement"
@ 8, 9 SAY "County Number"
@ 8, 26 SAY WEATHER->CNTYNUM
@ 9, 9 SAY "Month"
@ 9, 27 SAY WEATHER->MOON
@ 11, 32 SAY "No. of Yrs Avg Mean Std. Dev."
@ 12, 9 SAY "Precipitation"
@ 12, 38 SAY WEATHER->PRECYRS
@ 12, 49 SAY WEATHER->PRECMN
@ 12, 60 SAY WEATHER->PRECSD
@ 13, 9 SAY "Total Freeze Thaw Cycle"
@ 13, 38 SAY WEATHER->TFTCYRS
@ 13, 49 SAY WEATHER->TFTCMN
@ 13, 60 SAY WEATHER->TFTCSD
@ 14, 9 SAY "Wet Freeze Thaw Cycle"
@ 14, 38 SAY WEATHER->WFTCYRS
@ 14, 49 SAY WEATHER->WFTCMN
@ 14, 60 SAY WEATHER->WFTCSD
@ 15, 9 SAY "Maximum Temperature"
@ 15, 38 SAY WEATHER->MIMYRS
@ 15, 49 SAY WEATHER->IMMPMN
@ 15, 60 SAY WEATHER->IMMFSD
@ 16, 9 SAY "Averaged Temperature"
@ 16, 38 SAY WEATHER->ATMPYRS
@ 16, 49 SAY WEATHER->ATMPMN
@ 16, 60 SAY WEATHER->ATMFSD
@ 3, 4 TO 18, 72 DOUBLE
@ 5, 24  SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 6, 35  SAY "INQUIRY"
@ 7, 27  SAY "Environment Measurement"
@ 10, 23  SAY "County Number"
@ 10, 53  SAY ENV->CNYNUM
@ 11, 23  SAY "Thornthwaite Index Mean"
@ 11, 50  SAY ENV->THORNMIN PICTURE "9999999.999"
@ 12, 23  SAY "Thornthwaite Index"
@ 13, 28  SAY "- No. of Years Averaged"
@ 13, 55  SAY ENV->THORNYRS
@ 14, 23  SAY "Thornthwaite Index"
@ 15, 28  SAY "- Standard Deviation"
@ 15, 51  SAY ENV->THORNSD
@ 4, 14  TO 17, 64  DOUBLE
Format Programs For Screens
County Name

@ 7, 23  SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 8, 34  SAY "INQUIRY"
@ 9, 32  SAY "County Table"
@ 12, 24  SAY "County Number"
@ 12, 40  SAY CNTYTbl->CNTYNUM
@ 13, 24  SAY "County Name"
@ 13, 40  SAY CNTYTbl->CNTYNAME
@  6, 15  TO 15, 61  DOUBLE
@ 15, 36  TO 15, 36
Format Programs For Screens
Material Type Classification

@ 4, 24 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 5, 35 SAY "INQUIRY"
@ 6, 23 SAY "Material Type Classification Table"
@ 9, 21 SAY "Material Code"
@ 9, 42 SAY MATLTLB->MATCODE
@ 10, 21 SAY "Material Description"
@ 10, 42 SAY MATLTLB->MATDESC
@ 11, 21 SAY "Material Short Form"
@ 11, 42 SAY MATLTLB->MATSHRT
@ 12, 21 SAY "Layer Description"
@ 12, 42 SAY MATLTLB->LAYRDES
@ 3, 14 TO 14, 64 DOUBLE
Format Programs For Screens
Type of Pavement

@ 4, 24 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 5, 35 SAY "INQUIRY"
@ 6, 27 SAY "Type of Pavement Table"
@ 10, 20 SAY "Pavement Code"
@ 10, 36 SAY PAVE->PAVECODE
@ 11, 20 SAY "Type of Base"
@ 11, 36 SAY PAVE->BASETYPE
@ 12, 20 SAY "Surface Thickness"
@ 12, 36 SAY PAVE->BASETHK
@ 13, 20 SAY "Base Seal"
@ 13, 36 SAY PAVE->BASESEAL
@ 3, 14 TO 15, 64 DOUBLE
Format Programs For Screens
District Temperature Constant

@  6, 24 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@  7, 36 SAY "INQUIRY"
@  8, 27 SAY "District Temperature Table"
@ 11, 27 SAY "District Number"
@ 11, 51 SAY DISTTEMP->DISTRICT
@ 12, 27 SAY "Temperature Constant"
@ 12, 51 SAY DISTTEMP->TEMPOONS
@  5, 17 TO 14, 62   DOUBLE
@ 5, 24  SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 6, 35  SAY "INQUIRY"
@ 7, 31  SAY "Widening Table"
@ 10,  8  SAY "Code"
@ 10, 21  SAY WIDENFLG->WIDNCODE
@ 11,  8  SAY "Description"
@ 11, 21  SAY WIDENFLG->WIDNDESC
@ 12,  8  SAY "Comments:
@ 13, 10  SAY WIDENFLG->COMMENTS
@  4,  4  TO 15, 73  DOUBLE
@ 4, 21 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 5, 33 SAY "INQUIRY"
@ 6, 26 SAY "Layer Description Table"
@ 9, 16 SAY "Code"
@ 9, 33 SAY LAYERTBL->CODE
@ 10, 16 SAY "Short Description"
@ 10, 33 SAY LAYERTBL->CODE_DESC
@ 11, 16 SAY "Description"
@ 11, 33 SAY LAYERTBL->CODE_LAYR
@ 3, 12 TO 13, 61 DOUBLE
Format Programs For Screens
Functional Classification

@ 6, 25 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 7, 37 SAY "INQUIRY"
@ 8, 25 SAY "Functional Classification Table"
@ 11, 13 SAY "Code"
@ 11, 27 SAY FUNCLTBL->CODE PICTURE "9"
@ 12, 13 SAY "Description"
@ 12, 27 SAY FUNCLTBL->DESCRIPT
@ 5, 10 TO 14, 69 DOUBLE
* SUBSYSTEM: INQUIRY MAIN MENU
* PROGRAM NAME: INQUIRY.PRG 05/31/88
* MODIFIED ON: 09/26/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO RETRIEVE DATA TO THE SCREEN FOR THE FOLLOWING:
  * Monitoring data
  * Environment data
  * Inventory data
  * Traffic data
  * Tables

* set parameters and initialize variables
STORE 13 TO MSID NO
CLOSE DATABASES
SET FORMAT TO
SET PROCEDURE TO \PAVEDB\INQUIRY\INQ_COLL
MREPEAT2 = .T.

DO WHILE MREPEAT2
  STORE " " TO Inqpick
  * set up the INQUIRY menu screen and do the loop until INQPICK is 1-5
  DO WHILE .NOT. (INQPICK $ '12345')
    @ 0, 0 CLEAR
    @ 4, 19 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE 1.0"
    @ 5, 32 SAY "Inquiry"
    @ 9, 17 SAY "1 - Inventory Data"
    @ 10, 17 SAY "2 - Monitoring Data"
    @ 11, 17 SAY "3 - Traffic Data"
    @ 12, 17 SAY "4 - Environmental Data"
    @ 13, 17 SAY "5 - Tables"
    @ 16, 37 SAY "OPTION ←→ " GET INQPICK
    @ 3, 9 TO 18, 65 DOUBLE
    READ
    IF READKEY() = 12
      RETURN TO MASTER
    ENDIF
  ENDDO

IF MREPEAT2
  * according to the response received from the INQUIRY menu, the appropriate
  * commands are executed
  DO CASE
    * Display the Inventory data to the screen
    CASE INQPICK = "1"
      MREPEAT = .T.
      DO WHILE WHILE MREPEAT

65
CLEAR
INQCOLL = ""
MSID_NO = 13
DO WHILE .NOT. (INQCOLL $ '123456')
  @ 3, 21 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE 1.1"
  @ 4, 33 SAY "INQUIRY"
  @ 5, 30 SAY "Inventory Data"
  @ 7, 16 SAY "1 - Location"
  @ 8, 16 SAY "2 - Layer ID"
  @ 9, 16 SAY "3 - Geometric & Shoulder"
  @ 10, 16 SAY "4 - Surface"
  @ 11, 16 SAY "5 - Subgrade"
  @ 12, 16 SAY "6 - Layer Thickness Across the Road"
  @ 15, 37 SAY "OPTION " GET INQCOLL
  @ 17, 27 SAY "Enter SID Number " GET MSID_NO PICTURE "9999"
  @ 2, 9 TO 19, 65 DOUBLE
READ
IF READKEY() = 12
  MREPEAT = .F.
  EXIT
ENDIF
ENDO
IF MREPEAT
  DO COLLECT1
  CLEA TYPE
  ENDIF
  SET FORMAT TO
ENDO

* Display the Monitoring data to the screen if the choice is 2 on the
* Inquiry main menu
CASE INQCOLL = "2"
  MREPEAT = .T.
  DO WHILE MREPEAT
    CLEAR
    INQCOLL = ""
    MSID_NO = 13
    DO WHILE .NOT. (INQCOLL $ '123456')
      @ 3, 22 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE 1.2"
      @ 4, 34 SAY "INQUIRY"
      @ 5, 30 SAY "Monitoring Data"
      @ 8, 15 SAY "1 - Visual"
      @ 9, 15 SAY "2 - Serviceability Index"
      @ 10, 15 SAY "3 - Falling Weight"
      @ 11, 15 SAY "4 - Dynaflect"
      @ 12, 15 SAY "5 - Skid"
      @ 14, 40 SAY "OPTION " GET INQCOLL
      @ 16, 30 SAY "Enter SID Number " GET MSID_NO PICTURE "9999"
      @ 2, 9 TO 18, 65 DOUBLE
READ
IF READKEY() = 12
  MREPEAT = .F.
  EXIT
ENDIF
ENDDO
IF MREPEAT
   DO COLLECT2
      CLEA TYPE
   ENDF
   SET FORMAT TO
ENDDO

* If choice 3 is chosen, display data for the traffic file
CASE INQPICK = "3"
   CLEAR
   * Display TRAFFIC data for a SID number to the screen
   MREPEAT = .T.
   STORE "Traffic Data" TO MENUHEAD
   MENUHEAD = TRIM(MENUHEAD)
   MCENTER = 37 - (LEN(MENUHEAD)/2)
   MSID_NO = 13
   DO SID_INQ
      IF MREPEAT
         USE \PAVEDB\FILES\TRAFFIC INDE \PAVEDB\INDEXES\TRAFFIC
         SEEK STR(MSID_NO,4)
         IF FOUND()
            SET FORMAT TO \PAVEDB\INQUIRY\INQ_TRAF
            CHANGE
         ELSE
            @ 20, 15 SAY "SID number not found"
            WAIT
         ENDF
      ENDF
      USE
      SET FORMAT TO
   ENDDO

* If choice 4 is chosen, display one of the 5 INQUIRY table screens
CASE INQPICK = "4"
   MREPEAT = .T.
   DO WHILE MREPEAT
      CLEAR
      INQCOLL = " "
      MONTYNUM = 1
      DO WHILE .NOT. (INQCOLL $ '12')
         @ 4, 21 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE 1.4"
         @ 5, 33 SAY "INQUIRY"
         @ 6, 28 SAY "Environmental Data"
         @ 9, 15 SAY "1 - Environment"
         @ 10, 15 SAY "2 - Weather"
         @ 14, 40 SAY "OPTION ==> " GET INQCOLL
         @ 16, 27 SAY "Enter county Number ==> " GET MONTYNUM PICTURE "999"
         @ 3, 9 TO 18, 65 DOUBLE
         READ
         IF READKEY() = 12
            MREPEAT = .F.
         EXIT
      ENDF
   ENDDO
IF MREPEAT
  DO CASE
  CASE INQCOLL = "1"
    * Display ENVIRONMENT data for a SID number to the screen
    USE \PAVEDB\FILES\ENV INDE \PAVEDB\INDEXES\ENV
    SEEK MCNTYNUM
    IF FOUND()
      MFOUND = .T.
    ENDIF
    IF MCNTYNUM = 0
      MFOUND = .T.
      GOTO TOP
    ENDIF
    IF MFOUND
      SET FORMAT TO \PAVEDB\INQUIRY\INQ_ENV
      CHANGE
    ELSE
      @ 20, 15 SAY "COUNTY number not found"
      WAIT
    ENDIF
    USE
  
  CASE INQCOLL = "2"
    * Display WEATHER data for a SID number to the screen
    USE \PAVEDB\FILES\WEATHER INDE \PAVEDB\INDEXES\WEATHER
    SEEK STR(MCINTYNUM, 3)
    IF FOUND()
      MFOUND = .T.
    ENDIF
    IF MCNTYNUM = 0
      MFOUND = .T.
      GOTO TOP
    ENDIF
    IF MFOUND
      SET FORMAT TO \PAVEDB\INQUIRY\INQ_WEAT
      CHANGE
    ELSE
      @ 20, 15 SAY "COUNTY number not found"
      WAIT
    ENDIF
  ENDCASE
  ENDF
  SET FORMAT TO
  USE
ENDO

* if choice 5 is chosen, display one of the 5 INQUIRY table screens
CASE INQPICK = "5"
  DO \PAVEDB\EDITUPDT\TABLFILE
  MREPEAT = .T.
  DO WHILE MREPEAT
    CLEAR
    INQCOLL = " "
    MCNTYNUM = 1
  END
DO WHILE .NOT. (INQCOLL $ '1234567')
  @ 3, 21 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE" 1.5"
  @ 4, 33 SAY "INQUIRY"
  @ 5, 33 SAY "Tables"
  @ 8, 16 SAY "1 - County Name"
  @ 9, 16 SAY "2 - Material Type"
  @ 10, 16 SAY "3 - Type of Pavement"
  @ 11, 16 SAY "4 - District Temperature Constant"
  @ 12, 16 SAY "5 - Widening Flag"
  @ 13, 16 SAY "6 - Layer Description"
  @ 14, 16 SAY "7 - Functional Classification"
  @ 17, 42 SAY "OPTION ------> " GET INQCOLL
  @ 2, 9 TO 18, 65 DOUBLE
READ
  IF READKEY() = 12
    MREPEAT = .F.
    EXIT
  ENDF
ENDDO
IF MREPEAT
DO CASE
CASE INQCOLL = "1"
  @ 19, 15 SAY "Please enter the COUNTY number: "
  @ 19, 49 GET MONTYNUM PICTURE "999"
READ
CLEAR
USE \PAVEDB\FILES\CNYTBL INDEX \PAVEDB\INDEXES\CNYTELNO
LOCATE FOR CNTYNUM = MONTYNUM
IF MONTYNUM = 0
  GO TO TOP
ENDIF
SET FORMAT TO \PAVEDB\INQUIRY\INQ_CNY
CHANGE
USE
CASE INQCOLL = "2"
CLEAR
USE \PAVEDB\FILES\MATTLBL
SET FORMAT TO \PAVEDB\INQUIRY\INQ_MAT
CHANGE
USE
CASE INQCOLL = "3"
CLEAR
USE \PAVEDB\FILES\PAVETYPE
SET FORMAT TO \PAVEDB\INQUIRY\INQ_PVMT
CHANGE
USE
CASE INQCOLL = "4"
MDIST = 1
  @ 19, 15 SAY "Please enter the DISTRICT number: "
  @ 19, 49 GET MDIST PICTURE "99"
READ
CLEAR
USE \PAVEDB\FILES\DISTTEMP
LOCATE FOR DISTRICT = MDIST

69
IF MDIST = 0
    GOTO TOP
ENDIF
SET FORMAT TO \PAVEDB\INQUIRY\INQ_DIST
CHANGE
USE
CASE INQCOLL = "5"
    CLEAR
    USE \PAVEDB\FILES\WIDENFLG
    SET FORMAT TO \PAVEDB\INQUIRY\INQ_WIDN
    CHANGE
    USE
CASE INQCOLL = "6"
    CLEAR
    USE \PAVEDB\FILES\LAYERBL
    SET FORMAT TO \PAVEDB\INQUIRY\INQ_LITBL
    CHANGE
    USE
CASE INQCOLL = "7"
    CLEAR
    USE \PAVEDB\FILES\FUNCLBL
    SET FORMAT TO \PAVEDB\INQUIRY\INQ_FUNC
    CHANGE
    USE
ENDCASE
ENDIF
SET FORMAT TO
ENDDO
ENDCASE
MREPEAT2 = .F.
ENDIF
MREPEAT2 = .T.
ENDDO
SET FORMAT TO
SET PROCEDURE TO
CLOSE DATABASES
CLEAR
RETURN TO MASTER
Program Listing

* SYSTEM: INQUIRY OF INVENTORY AND MONITORING DATA
* PROGRAM NAME: INQ.Collection.PRG
* MODIFIED ON: 09/26/88
* CALLED FROM: INQUIRY.PRG
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TII
* AUTHOR: TREvor X. PEREIRA
* PURPOSE: THIS IS A PROCEDURE FILE FOR THE MAIN INQUIRY PROGRAM. RESPECTIVE PROCEDURES ARE CALLED TO DISPLAY DATA ON THE SCREEN FOR INDIVIDUAL FILES.
*

***********************
* PROCEDURE GET S ID NUMBER FOR INQUIRY
***********************
PROCEDURE SID INQ
STORE 0 TO MSID_NO, VAR1, VAR2, VAR3, COMPARE
STORE .F. TO VALID

* display Sid number entry screen
DO WHILE .NOT. VALID
  @ 0, 0 CLEAR
  @ 6, 22 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
  @ 7, 34 SAY "INQUIRY"
  @ 8, MCENTER SAY MENUHEAD
  @ 11, 24 SAY "Please Enter SID Number:"
  @ 11, 49 GET MSID_NO PICTURE "9999"
  @ 5, 15 TO 13, 60 - DOUBLE

* get Sid Number
READ

* calculates the correct Sid Number
VAR1 = INT(MSID_NO/1000)
VAR2 = MOD(INT(MSID_NO/100),10)
VAR2 = VAR2 * 2
VAR3 = MOD(INT(MSID_NO/10),10)
VAR3 = VAR3 * 3
VAR4 = MOD(VAR1,10)+VAR2+VAR3
COMPARE = MOD(VAR4,10)

* sets up the escape key if want to abort
IF READKEY() = 12
  MREPEAT = .F.
  RETURN
ENDIF

* compares calculated Sid number with Sid number entered
IF COMPARE = MOD(MSID_NO,10)
STORE .T. TO VALID
ELSE
   @ 20, 10 SAY "Invalid SID Number"
   WAIT
ENDIF

* verifies that Sid number has been entered
IF MSID NO = 0
   STORE .F. TO VALID
   @ 20, 10 SAY "Invalid SID Number"
   WAIT
ENDIF
ENDDO

* returns to the calling program
RETURN

********************************************************************************
* PROCEDURE COLLECT 1
********************************************************************************

PROCEDURE COLLECT1
MFOUND = .F.
DO CASE
CASE INQCOLL = "1"
   * Display location data for a SID number to the screen
   USE \PAVEDB\FILES\LOCATION INDE \PAVEDB\INDEXES\LOCSID
   SEEK MSID NO
   IF FOUND()
      MFOUND = .T.
  ENDIF
   IF MSID NO = 0
      MFOUND = .T.
      GOTO TOP
   ENDIF
   IF MFOUND
      SET FORMAT TO \PAVEDB\INQUIRY\INQ_LOCN
      CHANGE
   ELSE
      @ 20, 15 SAY "SID number not found"
      WAIT
   ENDIF
CASE INQCOLL = "2"
   * Display IDENTIFICATION data for a SID number to the screen
   USE \PAVEDB\FILES\LAYER INDE \PAVEDB\INDEXES\LAYINDX
   SEEK STR(MSID NO,4)
   IF FOUND()
      MFOUND = .T.
   ENDIF
   IF MSID NO = 0
      MFOUND = .T.
      GOTO TOP
   ENDIF
   IF MFOUND .AND. (.NOT. EOF())
      SET FORMAT TO \PAVEDB\INQUIRY\INQ_LAYR
CHANGE ELSE @ 20, 15 SAY "SID number not found" WAIT ENDIF
CASE INQCOLL = "3"
* Display GEOMETRIC AND SHOULDER data for a SID number to the screen
USE \\PAVEDB\\FILES\\GEOGHO INDE \\PAVEDB\\INDEXES\\GEOINDX SEEK STR(MSID_NO,4) IF FOUND()
   MFOUND = .T.
ENDIF IF MSID NO = 0 MFOUND = .T. GOTO TOP ENDIF IF MFOUND .AND. (.NOT. EOF())
   SET FORMAT TO \\PAVEDB\\INQUIRY\\INQ_GEO
   CHANGE ELSE @ 20, 15 SAY "SID number not found"
WAIT ENDIF
CASE INQCOLL = "4"
* Display SURFACE data for a SID number to the screen
USE \\PAVEDB\\FILES\\SURFACE INDE \\PAVEDB\\INDEXES\\SURFNINX SEEK STR(MSID_NO,4) IF FOUND()
   MFOUND = .T.
ENDIF IF MSID NO = 0 MFOUND = .T. GOTO TOP ENDIF IF MFOUND .AND. (.NOT. EOF())
   SET FORMAT TO \\PAVEDB\\INQUIRY\\INQ_SURF
   CHANGE ELSE @ 20, 15 SAY "SID number not found"
WAIT ENDIF
CASE INQCOLL = "5"
* Display SUBGRADE data for a SID number to the screen
USE \\PAVEDB\\FILES\\SUBGRADE INDE \\PAVEDB\\INDEXES\\SUBGNDX SEEK STR(MSID_NO,4) IF FOUND()
   MFOUND = .T.
ENDIF IF MSID NO = 0 MFOUND = .T. GOTO TOP ENDIF IF MFOUND .AND. (.NOT. EOF())
   SET FORMAT TO \\PAVEDB\\INQUIRY\\INQ_SUBG
CHANGE
ELSE
   @ 20, 15 SAY "SID number not found"
   WAIT
ENDIF
CASE INQCOLL = "6"
   * Display LAYER THICKNESS ACROSS THE ROAD data for a SID number to the screen
   USE \PAVEDB\FILES\LAYTHICK INDE \PAVEDB\INDEXES\LAYINDX
   SEEK SIR(MSID_NO,4)
   IF FOUND()
      MFOUNT = .T.
   ENDIF
   IF MSID_NO = 0
      MFOUNT = .T.
      GOTO TOP
   ENDIF
   IF MFOUNT .AND. (.NOT. EOF())
      SET FORMAT TO \PAVEDB\INQUIRY\INQ_LAYT
      CHANGE
   ELSE
      @ 20, 15 SAY "SID number not found"
      WAIT
   ENDIF
ENDCASE
CLEAR
RETURN

*******************************
* PROCEDURE COLLECT2
*******************************
PROCEDURE COLLECT2
   MFOUNT = .F.
DO CASE
CASE INQCOLL = "1"
   * Display VISUAL data for a SID number to the screen
   USE \PAVEDB\FILES\VISUAL INDE \PAVEDB\INDEXES\VISUAL
   SEEK SIR(MSID_NO,4)
   IF FOUND()
      MFOUNT = .T.
   ENDIF
   IF MSID_NO = 0
      MFOUNT = .T.
      GOTO TOP
   ENDIF
   IF MFOUNT .AND. (.NOT. EOF())
      SET FORMAT TO \PAVEDB\INQUIRY\INQ_VISL
      CHANGE
   ELSE
      @ 20, 15 SAY "SID number not found"
      WAIT
   ENDIF
CASE INQCOLL = "2"
   * Display SERVICEABILITY INDEX data for a SID number to the screen
USE \PAVEDB\FILES\SI INDE \PAVEDB\INDEXES\SI
SEEK SIR(MSID_NO,4)
IF FOUND()
   MFOUND = .T.
ENDIF
IF MSID_NO = 0
   MFOUND = .T.
   GOTO TOP
ENDIF
IF MFOUND .AND. (.NOT. EOF())
   SET FORMAT TO \PAVEDB\INQUIRY\INQ_SI
   CHANGE
ELSE
   @ 20, 15 SAY "SID number not found"
   WAIT
ENDIF
CASE INQCOLL = "3"
   * Display FALLING WEIGHT data for a SID number to the screen
   USE \PAVEDB\FILES\FALLWEIGHT INDE \PAVEDB\INDEXES\FALLWEIGHT
   SEEK SIR(MSID_NO,4)
   IF FOUND()
      MFOUND = .T.
   ENDIF
   IF MSID_NO = 0
      MFOUND = .T.
      GOTO TOP
   ENDIF
   IF MFOUND .AND. (.NOT. EOF())
      SET FORMAT TO \PAVEDB\INQUIRY\INQ_FALL
      CHANGE
   ELSE
      @ 20, 15 SAY "SID number not found"
      WAIT
   ENDIF
CASE INQCOLL = "4"
   * Display DYNATECT data for a SID number to the screen
   USE \PAVEDB\FILES\DYNATEC INDE \PAVEDB\INDEXES\DYNATEC
   SEEK SIR(MSID_NO,4)
   IF FOUND()
      MFOUND = .T.
   ENDIF
   IF MSID_NO = 0
      MFOUND = .T.
      GOTO TOP
   ENDIF
   IF MFOUND .AND. (.NOT. EOF())
      SET FORMAT TO \PAVEDB\INQUIRY\INQ_DYNA
      CHANGE
   ELSE
      @ 20, 15 SAY "SID number not found"
      WAIT
   ENDIF
CASE INQCOIL = "5"
* Display SKID data for a SID number to the screen
USE \PAVEDB\FILES\SKID INDE \PAVEDB\INDEXES\SKID
SEEK STR(MSID_NO,4)
IF FOUND()
    MFOUND = .T.
ENDIF
IF MSID_NO = 0
    MFOUND = .T.
    GOTO TOP
ENDIF
IF MFOUND .AND. (.NOT. EOF())
    SET FORMAT TO \PAVEDB\INQUIRY\INQ_SKID
    CHANGE
ELSE
    @ 20, 15 SAY "SID number not found"
    WAIT
ENDIF
ENDCASE
CLEAR
RETURN
CHAPTER 3

REPORT SUBSYSTEM
The Report Menu System written in dBase III Plus allows the user to print reports for any of the 14 master files and 7 tables. In addition to these 23 general reports, two specific reports have been set up - Summary and Inventory Update. The reports have been divided into 7 categories: Summary, Inventory Update, Inventory Data, Monitoring Data, Traffic Data, Environment Data and Tables. When a report is chosen, the program accesses the appropriate file and retrieves the data. The report is then sent to the printer as well as the screen. The reports program (REPORTS.PRG) generates all the reports except Summary and Inventory Update for which it calls separate programs. REPORTS.PRG also produces all of the screens contained in the Report Subsystem. All of the Report Subsystem programs are stored in the subdirectory \PAVEDB\REPORTS. The data and view files are stored in the subdirectory \PAVEDB\FILES. The index files are stored in \PAVEDB\INDEXES.

The Summary and Inventory Update Reports have the option of being printed by Individual SID number (SUMMSID.PRG or SUMM2SID.PRG), by District (SUMMDIST.PRG or SUM2DIST.PRG) or by ALL SID numbers (SUMMALL.PRG or SUM2FILE.PRG). For the individual SID numbers, the program gets the SID number from the user, accesses the appropriate files to obtain the data and prints the report. For reports by district, the program obtains the district number from the user, gets the SID numbers for that district, accesses the applicable files to obtain the data and prints the report. For reports on all SID numbers, the program automatically gets the SID number, retrieves the data from the pertinent files and prints out the report.

The following information about the Report Subsystem is provided in this chapter:

- Program Flow Diagram
- Program Specifications
- Menu Screens
- Sample Reports
- Program Listings.
Report Subsystem - Program Flow Diagram

LOCATION.DBF
LAYER.DBF
SUBGRADE.DBF
SURFACE.DBF
GEOSHO.DBF
LAYTHICK.DBF

SUM2DIST.PRG
SUM2FILE.PRG
SUM2SID.PRG

GETLOCT2.PRG
PRTREPT2.PRG

Inventory Update Report
Inventory Update Blank Form

FIGURE 8
PROGRAM SPECIFICATION

Program Name: REPORTS.PRG

Purpose: Display report request screens and produce user requested reports.

Input Files: The following files are used along with their indices:

<table>
<thead>
<tr>
<th>Data File</th>
<th>Index File</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCATION.DBF</td>
<td>LOCSID.NDX</td>
</tr>
<tr>
<td>LAYER.DBF</td>
<td>IAYNDX.NDX</td>
</tr>
<tr>
<td>GEOSSHO.DBF</td>
<td>GEONDX.NDX</td>
</tr>
<tr>
<td>SURFACE.DBF</td>
<td>SURFNDX.NDX</td>
</tr>
<tr>
<td>SUBGRADE.DBF</td>
<td>SUBGNDX.NDX</td>
</tr>
<tr>
<td>LAYTHICK.DBF</td>
<td>LAYINDX.NDX</td>
</tr>
<tr>
<td>VISUAL.DBF</td>
<td>VISUAL.NDX</td>
</tr>
<tr>
<td>ST.DBF</td>
<td>ST.NDX</td>
</tr>
<tr>
<td>FALLWGT.DBF</td>
<td>FALLWGT.NDX</td>
</tr>
<tr>
<td>DYNAPL.ID.DBF</td>
<td>DYNAFLID.NDX</td>
</tr>
<tr>
<td>SKID.DBF</td>
<td>SKID.NDX</td>
</tr>
<tr>
<td>ENV.DBF</td>
<td>ENV.NDX</td>
</tr>
<tr>
<td>WEATHER.DBF</td>
<td>WEATHER.NDX</td>
</tr>
<tr>
<td>TRAFFIC.DBF</td>
<td>TRAFFIC.NDX</td>
</tr>
<tr>
<td>CNTYTB1.DBF</td>
<td>CNTYB1NO.NDX</td>
</tr>
</tbody>
</table>

dBase Programs Called (See Program Flow Diagram):

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMSID.PRG</td>
<td>Gets SID Number for the Summary Report</td>
</tr>
<tr>
<td>SUMMDIST.PRG</td>
<td>Gets SID Numbers of a District for the Summary Report</td>
</tr>
<tr>
<td>SUMMALL.PRG</td>
<td>Gets all SID Number's for the Summary</td>
</tr>
<tr>
<td>GETSKID.PRG</td>
<td>Gets Skid Data for Summary Report</td>
</tr>
<tr>
<td>GETENVIR.PRG</td>
<td>Gets Environment Data for Summary Report</td>
</tr>
<tr>
<td>GETLOCAT.PRG</td>
<td>Gets Location Data for Summary Report</td>
</tr>
<tr>
<td>GETPAVCON.PRG</td>
<td>Gets Pavement Condition Data for the Summary Report</td>
</tr>
</tbody>
</table>
dBase Programs Called (See Program Flow Diagram): (continued)

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRNTREPT.PRG</td>
<td>Prints the Summary Report</td>
</tr>
<tr>
<td>SUMM2SID.PRG</td>
<td>Gets SID Number for the Inventory Update Report</td>
</tr>
<tr>
<td>SUMM2DIST.PRG</td>
<td>Gets SID Numbers for a District for Inventory Update Report</td>
</tr>
<tr>
<td>SUM2FILE.PRG</td>
<td>Gets all SID Numbers for the Inventory Update Report</td>
</tr>
<tr>
<td>GETIOCT2.PRG</td>
<td>Gets Data for the Inventory Update Report.</td>
</tr>
<tr>
<td>PRTREPT2.PRG</td>
<td>Prints the Inventory Update Report.</td>
</tr>
<tr>
<td>PRTREPT3.PRT</td>
<td>Prints a Blank Inventory Update Form.</td>
</tr>
</tbody>
</table>

NOTE: REPORTS.PRG produces all of the reports except the Summary Report and the Inventory Update Report.

Output Files: None

Reports (Using dBASE III Procedures):

The following reports are printed on request:

- Location
- Location Section
- Layer Identification
- Layer Thickness
- Geometric & Shoulder Info
- Surface
- Subgrade
- Double Surface Treated

Summary
- For All SID Numbers
- For Individual SID Number
- For a Particular District

Inventory Update
- For Individual SID numbers
- For a Particular District
- For All SID Numbers

Monitoring Data
- Visual
- Serviceability Index
- Falling Weight
- Dynaflect
- Skid

Environment Data
- Environment
- Weather

Traffic Data
- Traffic

Tables
- County Name
- Material Type

REPORT FILES

SEC_LOCN.FRM
LOCATION.FRM
LAYER.FRM
LAYTHICK.FRM
GEOSHO.FRM
SURFACE.FRM
SUBGRADE.FRM
DSTREPT.FRM
SUMMALL.PRG
SUMMSTD.PRG
SUMMDIST.PRG
SUM2SID.PRG
SUM2DIST.PRG
FILE.PRG
VISUAL.FRM
SI.FRM
FALLNGHT.FRM &
FALLNGHT2.FRM
DYNFAFLD.FRM
SKID.FRM
ENV.FRM
WEATHER.FRM
TRAFFIC.FRM
COUNTYBL.FRM
MATERIAL.FRM

84
Reports (Using dBASE III Procedures): (continued)

- Type of Pavement
- District Temperature Constant
- Widening Flag
- Functional Classification
- Layer Description

REPORT FILES

PAVETYPE.FRM
DISTTEMP.FRM
WIDENFLG.FRM
FUNCLITBL.FRM
LAYERITBL.FRM
REPORT SCREENS

Other than the Main Menu, all of the screens illustrated in this section are produced by REPORTS.PRG.

**TEXAS FLEXIBLE PAVEMENT DATABASE**
**MAIN MENU**

1 - Inquiry  
2 - Reports  
3 - Edit & Update  
4 - Applications  
5 - Backup  
6 - Installation  
7 - Reindex Master Files  

Q - QUIT

OPTION ———>

Above is the Main Menu of the Texas Flexible Pavement Database System. When the Reports option is chosen, the next screen 2.0 is displayed. Choice 1 - Summary displays the screen 2.1 on the next page while choice 2 - Inventory Update Forms displays screen 2.2.

**TEXAS FLEXIBLE PAVEMENT DATABASE**
**Reports**

2.0

1 - Summary  
2 - Inventory Update Forms  
3 - Inventory Data  
4 - Monitoring Data  
5 - Traffic Data  
6 - Environmental Data  
7 - Tables

OPTION ———>
REPORT SCREENS (continued)

**TEXAS FLEXIBLE PAVEMENT DATABASE**  
**REPORTS**  
Summary Report

1 - By SID Number  
2 - By District  
3 - ALL SID Numbers

OPTION ➞

---

**TEXAS FLEXIBLE PAVEMENT DATABASE**  
**REPORTS**  
Inventory Update Forms

1 - By District  
2 - By SID Number  
3 - ALL SID Numbers

OPTION ➞
Choice 3 - Inventory Data on the Reports Menu brings up the above screen 2.3, while choice 4 - Monitoring Data brings up the next screen 2.4.
Choice 6 - Environmental Data displays the above screen 2.6 while choice 7 - Tables displays the next screen 2.7.
REPORT SUBSYSTEM
SAMPLE REPORTS
<table>
<thead>
<tr>
<th>SID No.</th>
<th>Actv Highway Flag</th>
<th>Highway Number</th>
<th>Control/ Section</th>
<th>Milepoint Begin to End</th>
<th>Prev Next Inact</th>
<th>No. of Lanes</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>T</td>
<td>92</td>
<td>22+00</td>
<td>US 82</td>
<td>R 45/4</td>
<td>6/75</td>
</tr>
<tr>
<td>26</td>
<td>T</td>
<td>92</td>
<td>4+00</td>
<td>6+00 FM 2729</td>
<td>R 2798/3</td>
<td>6/75</td>
</tr>
<tr>
<td>39</td>
<td>T</td>
<td>117</td>
<td>106+00</td>
<td>SH 30</td>
<td>R 9/13</td>
<td>6/75</td>
</tr>
<tr>
<td>42</td>
<td>T</td>
<td>117</td>
<td>28+00</td>
<td>SH 34</td>
<td>R 173/6</td>
<td>6/75</td>
</tr>
<tr>
<td>55</td>
<td>T</td>
<td>117</td>
<td>2+00</td>
<td>6+00 FM 2736</td>
<td>R 2732/1</td>
<td>6/75</td>
</tr>
<tr>
<td>68</td>
<td>T</td>
<td>117</td>
<td>2+00</td>
<td>8+00 FM 2717</td>
<td>R 136/8</td>
<td>6/75</td>
</tr>
<tr>
<td>84</td>
<td>F</td>
<td>139</td>
<td>14+00</td>
<td>14+16 FM 905</td>
<td>L 730/3</td>
<td>6/75</td>
</tr>
<tr>
<td>97</td>
<td>T</td>
<td>139</td>
<td>14+00</td>
<td>16+00 FM 79</td>
<td>L 730/3</td>
<td>6/75</td>
</tr>
<tr>
<td>102</td>
<td>T</td>
<td>190</td>
<td>4+11</td>
<td>8+00 US 69</td>
<td>R 203/3</td>
<td>6/75</td>
</tr>
<tr>
<td>115</td>
<td>T</td>
<td>190</td>
<td>2+00</td>
<td>4+00 FM 779</td>
<td>R 2605/1</td>
<td>6/75</td>
</tr>
<tr>
<td>128</td>
<td>T</td>
<td>73</td>
<td>8+00</td>
<td>10+00 SH 6</td>
<td>R 256/1</td>
<td>6/75</td>
</tr>
<tr>
<td>131</td>
<td>T</td>
<td>73</td>
<td>4+00</td>
<td>6+00 FM 2157</td>
<td>R 1990/1</td>
<td>6/75</td>
</tr>
<tr>
<td>144</td>
<td>T</td>
<td>120</td>
<td>36+00</td>
<td>38+00 US 281</td>
<td>L 249/7</td>
<td>6/75</td>
</tr>
<tr>
<td>157</td>
<td>T</td>
<td>120</td>
<td>6+00</td>
<td>4+00 FM 206</td>
<td>L 391/7</td>
<td>6/75</td>
</tr>
<tr>
<td>160</td>
<td>T</td>
<td>127</td>
<td>28+00</td>
<td>30+00 US 67</td>
<td>R 259/4</td>
<td>6/75</td>
</tr>
<tr>
<td>173</td>
<td>F</td>
<td>127</td>
<td>2+00</td>
<td>4+00 FM 917</td>
<td>R 1181/2</td>
<td>6/75</td>
</tr>
<tr>
<td>186</td>
<td>F</td>
<td>220</td>
<td>0+00</td>
<td>2+00 US 377</td>
<td>R 80/7</td>
<td>6/75</td>
</tr>
<tr>
<td>199</td>
<td>T</td>
<td>220</td>
<td>0+00</td>
<td>2+00 SH 3035</td>
<td>L 2208/1</td>
<td>6/75</td>
</tr>
<tr>
<td>204</td>
<td>T</td>
<td>220</td>
<td>2+00</td>
<td>4+00 FM 1709</td>
<td>L 1603/3</td>
<td>6/75</td>
</tr>
<tr>
<td>220</td>
<td>T</td>
<td>127</td>
<td>18+00</td>
<td>20+00 SH 35W</td>
<td>L 14/4</td>
<td>6/75</td>
</tr>
<tr>
<td>233</td>
<td>T</td>
<td>39</td>
<td>4+00</td>
<td>6+00 SH 79</td>
<td>L 282/2</td>
<td>6/75</td>
</tr>
<tr>
<td>248</td>
<td>T</td>
<td>39</td>
<td>10+00</td>
<td>12+00 FM 4197</td>
<td>L 1350/1</td>
<td>6/75</td>
</tr>
<tr>
<td>259</td>
<td>T</td>
<td>169</td>
<td>20+00</td>
<td>22+00 SH 59</td>
<td>L 229/2</td>
<td>6/75</td>
</tr>
<tr>
<td>262</td>
<td>T</td>
<td>169</td>
<td>6+00</td>
<td>8+00 FM 455</td>
<td>L 845/1</td>
<td>6/75</td>
</tr>
<tr>
<td>275</td>
<td>T</td>
<td>224</td>
<td>0+00</td>
<td>2+00 FM 2651</td>
<td>L 2645/1</td>
<td>6/75</td>
</tr>
<tr>
<td>288</td>
<td>F</td>
<td>224</td>
<td>28+00</td>
<td>30+00 US 183</td>
<td>L 147/1</td>
<td>6/75</td>
</tr>
<tr>
<td>291</td>
<td>F</td>
<td>234</td>
<td>10+00</td>
<td>10+20 FM 917</td>
<td>L 702/1</td>
<td>6/75</td>
</tr>
<tr>
<td>306</td>
<td>T</td>
<td>118</td>
<td>6+00</td>
<td>8+00 SH 152</td>
<td>L 1557/2</td>
<td>6/75</td>
</tr>
<tr>
<td>319</td>
<td>F</td>
<td>118</td>
<td>2+18</td>
<td>2+00 FM 1598</td>
<td>L 1515/3</td>
<td>6/75</td>
</tr>
<tr>
<td>322</td>
<td>T</td>
<td>118</td>
<td>2+18</td>
<td>2+00 SH 305</td>
<td>L 582/1</td>
<td>6/75</td>
</tr>
<tr>
<td>348</td>
<td>T</td>
<td>148</td>
<td>2+00</td>
<td>4+00 SH 305</td>
<td>L 461/13</td>
<td>6/75</td>
</tr>
<tr>
<td>351</td>
<td>T</td>
<td>148</td>
<td>2+00</td>
<td>30+01 SH 23</td>
<td>L 439/4</td>
<td>6/75</td>
</tr>
<tr>
<td>377</td>
<td>T</td>
<td>148</td>
<td>28+00</td>
<td>30+01 SH 23</td>
<td>L 439/4</td>
<td>6/75</td>
</tr>
<tr>
<td>393</td>
<td>T</td>
<td>180</td>
<td>20+00</td>
<td>22+00 SH 40</td>
<td>L 1041/1</td>
<td>6/75</td>
</tr>
<tr>
<td>408</td>
<td>T</td>
<td>180</td>
<td>4+00</td>
<td>6+00 US 385</td>
<td>L 229/2</td>
<td>6/75</td>
</tr>
<tr>
<td>411</td>
<td>T</td>
<td>180</td>
<td>4+00</td>
<td>16+00 SH 214</td>
<td>L 461/13</td>
<td>6/75</td>
</tr>
<tr>
<td>424</td>
<td>T</td>
<td>104</td>
<td>34+00</td>
<td>36+00 US 54</td>
<td>L 439/4</td>
<td>6/75</td>
</tr>
<tr>
<td>437</td>
<td>T</td>
<td>96</td>
<td>24+00</td>
<td>26+00 US 87</td>
<td>L 67/6</td>
<td>6/75</td>
</tr>
<tr>
<td>440</td>
<td>T</td>
<td>96</td>
<td>6+00</td>
<td>8+00 SH 194</td>
<td>L 439/4</td>
<td>6/75</td>
</tr>
<tr>
<td>453</td>
<td>T</td>
<td>96</td>
<td>26+00</td>
<td>29+00 FM 451</td>
<td>L 1041/1</td>
<td>6/75</td>
</tr>
<tr>
<td>466</td>
<td>T</td>
<td>96</td>
<td>0+00</td>
<td>2+00 FM 1612</td>
<td>L 2332/2</td>
<td>6/75</td>
</tr>
</tbody>
</table>

**EXHIBIT 1**

**SAMPLE LOCATION FILE**
<table>
<thead>
<tr>
<th>Standard ID Number</th>
<th>Highway District Number</th>
<th>County District Number</th>
<th>Highway Number</th>
<th>Beginning Milepost</th>
<th>Ending Milepost</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1</td>
<td>92</td>
<td>US 82</td>
<td>22+00</td>
<td>24+00</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>92</td>
<td>FM2729</td>
<td>4+00</td>
<td>6+00</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>117</td>
<td>IH 30</td>
<td>106+00</td>
<td>109+00</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>117</td>
<td>SH 34</td>
<td>28+00</td>
<td>30+00</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>117</td>
<td>FM1566</td>
<td>2+00</td>
<td>4+00</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
<td>117</td>
<td>FM2736</td>
<td>2-20</td>
<td>2+00</td>
</tr>
<tr>
<td>71</td>
<td>1</td>
<td>139</td>
<td>US 271</td>
<td>6+00</td>
<td>8+00</td>
</tr>
<tr>
<td>84</td>
<td>1</td>
<td>139</td>
<td>FM 905</td>
<td>14+00</td>
<td>14+16</td>
</tr>
<tr>
<td>97</td>
<td>1</td>
<td>139</td>
<td>FM 79</td>
<td>14+00</td>
<td>16+00</td>
</tr>
<tr>
<td>102</td>
<td>1</td>
<td>190</td>
<td>US 69</td>
<td>4+11</td>
<td>8+00</td>
</tr>
<tr>
<td>115</td>
<td>1</td>
<td>190</td>
<td>FM 779</td>
<td>2+00</td>
<td>4+00</td>
</tr>
<tr>
<td>128</td>
<td>2</td>
<td>73</td>
<td>SH 6</td>
<td>8+00</td>
<td>10+00</td>
</tr>
<tr>
<td>131</td>
<td>2</td>
<td>73</td>
<td>FM2157</td>
<td>4+00</td>
<td>6+00</td>
</tr>
<tr>
<td>144</td>
<td>2</td>
<td>120</td>
<td>US 281</td>
<td>36+00</td>
<td>38+00</td>
</tr>
<tr>
<td>157</td>
<td>2</td>
<td>120</td>
<td>FM 206</td>
<td>6+00</td>
<td>4+00</td>
</tr>
<tr>
<td>160</td>
<td>2</td>
<td>127</td>
<td>US 67</td>
<td>28+00</td>
<td>30+00</td>
</tr>
<tr>
<td>173</td>
<td>2</td>
<td>127</td>
<td>FM 917</td>
<td>2+00</td>
<td>4+00</td>
</tr>
<tr>
<td>186</td>
<td>2</td>
<td>220</td>
<td>US 377</td>
<td>0+00</td>
<td>2+00</td>
</tr>
<tr>
<td>199</td>
<td>2</td>
<td>220</td>
<td>SH 303S</td>
<td>0+00</td>
<td>2+00</td>
</tr>
<tr>
<td>204</td>
<td>2</td>
<td>220</td>
<td>FM1709</td>
<td>2+00</td>
<td>4+00</td>
</tr>
<tr>
<td>217</td>
<td>2</td>
<td>127</td>
<td>IH 35W</td>
<td>18+00</td>
<td>20+00</td>
</tr>
<tr>
<td>220</td>
<td>3</td>
<td>39</td>
<td>SH 79</td>
<td>4+00</td>
<td>6+00</td>
</tr>
<tr>
<td>233</td>
<td>3</td>
<td>39</td>
<td>FM1197</td>
<td>10+00</td>
<td>12+00</td>
</tr>
<tr>
<td>246</td>
<td>3</td>
<td>169</td>
<td>SH 59</td>
<td>20+00</td>
<td>22+00</td>
</tr>
<tr>
<td>259</td>
<td>3</td>
<td>169</td>
<td>FM 455</td>
<td>6+00</td>
<td>8+00</td>
</tr>
<tr>
<td>262</td>
<td>3</td>
<td>224</td>
<td>US 183</td>
<td>34+00</td>
<td>34+20</td>
</tr>
<tr>
<td>275</td>
<td>3</td>
<td>224</td>
<td>FM2651</td>
<td>0-01</td>
<td>2+00</td>
</tr>
<tr>
<td>288</td>
<td>3</td>
<td>244</td>
<td>US 183</td>
<td>28+00</td>
<td>30+00</td>
</tr>
<tr>
<td>291</td>
<td>3</td>
<td>244</td>
<td>FM 91</td>
<td>10+00</td>
<td>10+20</td>
</tr>
<tr>
<td>306</td>
<td>4</td>
<td>33</td>
<td>IH 40</td>
<td>104+00</td>
<td>106+00</td>
</tr>
<tr>
<td>319</td>
<td>4</td>
<td>33</td>
<td>US 60</td>
<td>26+00</td>
<td>26+18</td>
</tr>
<tr>
<td>322</td>
<td>4</td>
<td>33</td>
<td>FM1342</td>
<td>10+00</td>
<td>12+08</td>
</tr>
<tr>
<td>335</td>
<td>4</td>
<td>104</td>
<td>US 87</td>
<td>4+00</td>
<td>6+00</td>
</tr>
<tr>
<td>348</td>
<td>4</td>
<td>104</td>
<td>FM 998</td>
<td>2+00</td>
<td>4+00</td>
</tr>
<tr>
<td>351</td>
<td>4</td>
<td>118</td>
<td>SH 152</td>
<td>6+00</td>
<td>8+00</td>
</tr>
<tr>
<td>364</td>
<td>4</td>
<td>118</td>
<td>FM1598</td>
<td>2-18</td>
<td>2+00</td>
</tr>
<tr>
<td>377</td>
<td>4</td>
<td>148</td>
<td>SH 305</td>
<td>2+00</td>
<td>4+00</td>
</tr>
</tbody>
</table>

**SAMPLE TEST LOCATION FILE REPORT**

**EXHIBIT 2**

94
Texas Flexible Database
Layer Identification File

<table>
<thead>
<tr>
<th>SID</th>
<th>Struc Layer No.</th>
<th>Layer No.</th>
<th>Layer Description</th>
<th>Layer Center Thickness</th>
<th>Material Type Class.</th>
<th>Date Job Completed</th>
<th>Layer Widened</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>0.00</td>
<td>44</td>
<td>6/31</td>
<td>0/0</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>6.00</td>
<td>17</td>
<td>6/31</td>
<td>8/46</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2.00</td>
<td>1</td>
<td>7/47</td>
<td>0/0</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1.30</td>
<td>1</td>
<td>1/63</td>
<td>0/0</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>0.40</td>
<td>11</td>
<td>8/79</td>
<td>0/0</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>0.00</td>
<td>44</td>
<td>10/65</td>
<td>0/0</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>6.00</td>
<td>21</td>
<td>10/65</td>
<td>0/0</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5.20</td>
<td>21</td>
<td>6/66</td>
<td>0/0</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>0.36</td>
<td>6</td>
<td>6/66</td>
<td>0/0</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>0.25</td>
<td>6</td>
<td>6/66</td>
<td>0/0</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>0.00</td>
<td>44</td>
<td>9/52</td>
<td>0/0</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>6.00</td>
<td>31</td>
<td>9/52</td>
<td>0/0</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>10.00</td>
<td>17</td>
<td>9/52</td>
<td>9/52</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2.50</td>
<td>1</td>
<td>8/67</td>
<td>0/0</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>1.50</td>
<td>1</td>
<td>8/67</td>
<td>0/0</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>0.00</td>
<td>44</td>
<td>6/45</td>
<td>0/0</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>6.00</td>
<td>25</td>
<td>6/45</td>
<td>0/0</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>3</td>
<td>10</td>
<td>0.25</td>
<td>5</td>
<td>6/45</td>
<td>0/0</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>4.00</td>
<td>21</td>
<td>12/54</td>
<td>0/0</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>0.26</td>
<td>6</td>
<td>12/54</td>
<td>0/0</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>6</td>
<td>10</td>
<td>0.26</td>
<td>6</td>
<td>12/54</td>
<td>0/0</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>7</td>
<td>5</td>
<td>4.00</td>
<td>21</td>
<td>7/64</td>
<td>0/0</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>8</td>
<td>10</td>
<td>0.36</td>
<td>6</td>
<td>7/64</td>
<td>0/0</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>9</td>
<td>10</td>
<td>0.25</td>
<td>6</td>
<td>7/64</td>
<td>0/0</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>10</td>
<td>3</td>
<td>0.30</td>
<td>11</td>
<td>9/72</td>
<td>0/0</td>
</tr>
<tr>
<td>42</td>
<td>2</td>
<td>11</td>
<td>2</td>
<td>0.40</td>
<td>11</td>
<td>7/82</td>
<td>0/0</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>0.00</td>
<td>44</td>
<td>9/51</td>
<td>0/0</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>4.00</td>
<td>21</td>
<td>9/51</td>
<td>0/0</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>3</td>
<td>10</td>
<td>0.20</td>
<td>6</td>
<td>9/51</td>
<td>0/0</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>0.20</td>
<td>6</td>
<td>9/51</td>
<td>0/0</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>0.20</td>
<td>11</td>
<td>7/56</td>
<td>0/0</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>6</td>
<td>3</td>
<td>0.33</td>
<td>11</td>
<td>6/63</td>
<td>0/0</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>0.00</td>
<td>44</td>
<td>9/65</td>
<td>0/0</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>6.00</td>
<td>21</td>
<td>9/65</td>
<td>0/0</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
<td>3</td>
<td>10</td>
<td>0.36</td>
<td>6</td>
<td>9/65</td>
<td>0/0</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>0.25</td>
<td>6</td>
<td>9/65</td>
<td>0/0</td>
</tr>
</tbody>
</table>

SAMPLE LAYER IDENTIFICATION FILE REPORT

EXHIBIT 3
<table>
<thead>
<tr>
<th>Section Identification Number</th>
<th>Date Job Completed (Month/Year)</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>6/66</td>
</tr>
<tr>
<td>26</td>
<td>6/66</td>
</tr>
<tr>
<td>42</td>
<td>12/54</td>
</tr>
<tr>
<td>42</td>
<td>12/54</td>
</tr>
<tr>
<td>42</td>
<td>7/64</td>
</tr>
<tr>
<td>42</td>
<td>7/64</td>
</tr>
<tr>
<td>55</td>
<td>9/51</td>
</tr>
<tr>
<td>55</td>
<td>9/51</td>
</tr>
<tr>
<td>68</td>
<td>9/65</td>
</tr>
<tr>
<td>68</td>
<td>9/65</td>
</tr>
<tr>
<td>84</td>
<td>6/71</td>
</tr>
<tr>
<td>84</td>
<td>6/71</td>
</tr>
<tr>
<td>97</td>
<td>8/55</td>
</tr>
<tr>
<td>102</td>
<td>6/66</td>
</tr>
<tr>
<td>102</td>
<td>6/66</td>
</tr>
<tr>
<td>115</td>
<td>7/62</td>
</tr>
<tr>
<td>115</td>
<td>7/62</td>
</tr>
<tr>
<td>128</td>
<td>5/40</td>
</tr>
<tr>
<td>128</td>
<td>5/40</td>
</tr>
<tr>
<td>144</td>
<td>8/53</td>
</tr>
<tr>
<td>144</td>
<td>8/53</td>
</tr>
<tr>
<td>160</td>
<td>5/41</td>
</tr>
<tr>
<td>160</td>
<td>5/41</td>
</tr>
<tr>
<td>186</td>
<td>7/47</td>
</tr>
<tr>
<td>186</td>
<td>7/47</td>
</tr>
<tr>
<td>199</td>
<td>9/60</td>
</tr>
<tr>
<td>199</td>
<td>9/60</td>
</tr>
<tr>
<td>217</td>
<td>8/66</td>
</tr>
<tr>
<td>217</td>
<td>8/66</td>
</tr>
<tr>
<td>220</td>
<td>11/41</td>
</tr>
<tr>
<td>220</td>
<td>11/41</td>
</tr>
<tr>
<td>246</td>
<td>5/42</td>
</tr>
<tr>
<td>246</td>
<td>5/42</td>
</tr>
<tr>
<td>259</td>
<td>6/49</td>
</tr>
<tr>
<td>SID</td>
<td>Struc No</td>
</tr>
<tr>
<td>-----</td>
<td>----------</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
</tr>
<tr>
<td>71</td>
<td>1</td>
</tr>
<tr>
<td>71</td>
<td>1</td>
</tr>
<tr>
<td>71</td>
<td>1</td>
</tr>
<tr>
<td>71</td>
<td>1</td>
</tr>
<tr>
<td>84</td>
<td>1</td>
</tr>
<tr>
<td>84</td>
<td>1</td>
</tr>
</tbody>
</table>

**Sample Layer Thickness Across the Road File Report**

EXHIBIT 5
<table>
<thead>
<tr>
<th>SID</th>
<th>Struc No</th>
<th>Pavmtn Width</th>
<th>Outside Shoulder</th>
<th>Shoulder Surface Type</th>
<th>Base Surface Type</th>
<th>Shoulder Widen</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>2</td>
<td>12.0</td>
<td>0.0</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>10.0</td>
<td>0.0</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>12.0</td>
<td>17.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>12.0</td>
<td>10.0</td>
<td>5</td>
<td>21</td>
<td>0.36</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>9.0</td>
<td>0.0</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
<td>10.0</td>
<td>0.0</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>71</td>
<td>1</td>
<td>12.0</td>
<td>15.0</td>
<td>2</td>
<td>31</td>
<td>0.00</td>
</tr>
<tr>
<td>84</td>
<td>1</td>
<td>10.0</td>
<td>0.0</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>97</td>
<td>1</td>
<td>10.0</td>
<td>0.0</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>102</td>
<td>1</td>
<td>12.0</td>
<td>10.0</td>
<td>5</td>
<td>21</td>
<td>0.36</td>
</tr>
<tr>
<td>115</td>
<td>1</td>
<td>10.0</td>
<td>0.0</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>128</td>
<td>1</td>
<td>11.0</td>
<td>6.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>128</td>
<td>2</td>
<td>11.0</td>
<td>6.0</td>
<td>5</td>
<td>21</td>
<td>0.30</td>
</tr>
<tr>
<td>128</td>
<td>3</td>
<td>11.0</td>
<td>6.0</td>
<td>5</td>
<td>21</td>
<td>0.60</td>
</tr>
<tr>
<td>131</td>
<td>1</td>
<td>10.0</td>
<td>3.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>131</td>
<td>2</td>
<td>10.0</td>
<td>3.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>131</td>
<td>3</td>
<td>10.0</td>
<td>3.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>144</td>
<td>1</td>
<td>12.0</td>
<td>7.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>144</td>
<td>2</td>
<td>12.0</td>
<td>7.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>157</td>
<td>1</td>
<td>8.0</td>
<td>4.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>157</td>
<td>2</td>
<td>10.0</td>
<td>0.0</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>157</td>
<td>3</td>
<td>10.0</td>
<td>0.0</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>160</td>
<td>1</td>
<td>12.0</td>
<td>10.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>160</td>
<td>2</td>
<td>12.0</td>
<td>10.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>173</td>
<td>1</td>
<td>9.0</td>
<td>7.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>173</td>
<td>2</td>
<td>9.0</td>
<td>7.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>173</td>
<td>3</td>
<td>9.0</td>
<td>7.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>186</td>
<td>1</td>
<td>12.0</td>
<td>8.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>186</td>
<td>2</td>
<td>12.0</td>
<td>8.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>199</td>
<td>1</td>
<td>12.0</td>
<td>4.3</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>199</td>
<td>2</td>
<td>12.0</td>
<td>4.3</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>199</td>
<td>3</td>
<td>12.0</td>
<td>4.3</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**SAMPLE GEOMETRIC AND SHOULDER INFORMATION FILE REPORT**

**EXHIBIT 6**

98
Texas Flexible Pavement Database
Surface File

<table>
<thead>
<tr>
<th>SID</th>
<th>Structure Layer Number</th>
<th>Aggregate Application Number</th>
<th>Admixture Type</th>
<th>Percent</th>
<th>Asphalt Application Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>AC</td>
<td>5.70</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>AC</td>
<td>5.40</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>5</td>
<td>75</td>
<td>AC-10</td>
<td>0.00</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>4</td>
<td>80</td>
<td>AC-10</td>
<td>0.00</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5</td>
<td>120</td>
<td>AC-10</td>
<td>0.00</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>AC</td>
<td>0.00</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>AC</td>
<td>4.70</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>AC</td>
<td>4.90</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>3</td>
<td>120</td>
<td>OA-230</td>
<td>0.00</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>5</td>
<td>110</td>
<td>OA-230</td>
<td>0.00</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>6</td>
<td>110</td>
<td>OA-230</td>
<td>0.00</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>8</td>
<td>80</td>
<td>OA-135</td>
<td>0.00</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>9</td>
<td>120</td>
<td>OA-135</td>
<td>0.00</td>
</tr>
<tr>
<td>42</td>
<td>2</td>
<td>10</td>
<td>95</td>
<td>AC-5</td>
<td>0.00</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>3</td>
<td>140</td>
<td>RC-2</td>
<td>0.00</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>4</td>
<td>145</td>
<td>RC-2</td>
<td>0.00</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>6</td>
<td>85</td>
<td>OA-135</td>
<td>0.00</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
<td>3</td>
<td>80</td>
<td>OA-135</td>
<td>0.00</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
<td>4</td>
<td>120</td>
<td>OA-135</td>
<td>0.00</td>
</tr>
<tr>
<td>71</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>71</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>AC</td>
<td>4.97</td>
</tr>
<tr>
<td>71</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>AC</td>
<td>4.60</td>
</tr>
<tr>
<td>84</td>
<td>1</td>
<td>4</td>
<td>80</td>
<td>AC-5</td>
<td>0.00</td>
</tr>
<tr>
<td>84</td>
<td>1</td>
<td>5</td>
<td>120</td>
<td>AC-5</td>
<td>0.00</td>
</tr>
<tr>
<td>97</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>102</td>
<td>1</td>
<td>4</td>
<td>80</td>
<td>OA-135</td>
<td>0.00</td>
</tr>
<tr>
<td>102</td>
<td>1</td>
<td>5</td>
<td>120</td>
<td>OA-135</td>
<td>0.00</td>
</tr>
<tr>
<td>102</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>102</td>
<td>2</td>
<td>7</td>
<td>75</td>
<td>AC-10</td>
<td>0.00</td>
</tr>
<tr>
<td>115</td>
<td>1</td>
<td>3</td>
<td>120</td>
<td>OA-135</td>
<td>0.00</td>
</tr>
<tr>
<td>115</td>
<td>1</td>
<td>4</td>
<td>140</td>
<td>OA-135</td>
<td>0.00</td>
</tr>
<tr>
<td>128</td>
<td>1</td>
<td>3</td>
<td>90</td>
<td>OA-135</td>
<td>0.00</td>
</tr>
<tr>
<td>128</td>
<td>1</td>
<td>4</td>
<td>180</td>
<td>OA-135</td>
<td>0.00</td>
</tr>
<tr>
<td>128</td>
<td>1</td>
<td>5</td>
<td>180</td>
<td>OA-230</td>
<td>0.00</td>
</tr>
</tbody>
</table>

SAMPLE SURFACE FILE REPORT

EXHIBIT 7
Texas Flexible Pavement Database  
Subgrade File

<table>
<thead>
<tr>
<th>SID</th>
<th>Struc Number</th>
<th>Layer Number</th>
<th>Percent Passing 200 Sieve</th>
<th>Plast. Index</th>
<th>Texas Liquid Limit</th>
<th>Triaxial Class</th>
<th>Permeability Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>88.8</td>
<td>40.4</td>
<td>64.5</td>
<td>5.3</td>
<td>0.23</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>1</td>
<td>63.5</td>
<td>27.5</td>
<td>50.1</td>
<td>5.2</td>
<td>0.36</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>1</td>
<td>85.7</td>
<td>39.1</td>
<td>61.3</td>
<td>5.6</td>
<td>0.60</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>1</td>
<td>79.3</td>
<td>29.7</td>
<td>48.0</td>
<td>5.1</td>
<td>0.06</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>1</td>
<td>81.8</td>
<td>33.8</td>
<td>53.6</td>
<td>5.3</td>
<td>0.06</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
<td>1</td>
<td>79.3</td>
<td>29.7</td>
<td>48.0</td>
<td>5.1</td>
<td>0.06</td>
</tr>
<tr>
<td>71</td>
<td>1</td>
<td>1</td>
<td>70.9</td>
<td>26.7</td>
<td>36.6</td>
<td>4.2</td>
<td>0.27</td>
</tr>
<tr>
<td>84</td>
<td>1</td>
<td>1</td>
<td>85.0</td>
<td>41.8</td>
<td>65.5</td>
<td>0.0</td>
<td>0.06</td>
</tr>
<tr>
<td>97</td>
<td>1</td>
<td>1</td>
<td>79.1</td>
<td>29.9</td>
<td>48.0</td>
<td>5.6</td>
<td>0.06</td>
</tr>
<tr>
<td>102</td>
<td>1</td>
<td>1</td>
<td>78.0</td>
<td>24.0</td>
<td>48.0</td>
<td>4.8</td>
<td>0.06</td>
</tr>
<tr>
<td>115</td>
<td>1</td>
<td>1</td>
<td>92.9</td>
<td>35.0</td>
<td>50.0</td>
<td>0.0</td>
<td>0.16</td>
</tr>
<tr>
<td>128</td>
<td>1</td>
<td>1</td>
<td>58.5</td>
<td>21.9</td>
<td>42.6</td>
<td>4.7</td>
<td>0.40</td>
</tr>
<tr>
<td>131</td>
<td>1</td>
<td>1</td>
<td>41.7</td>
<td>15.5</td>
<td>31.4</td>
<td>0.0</td>
<td>2.19</td>
</tr>
<tr>
<td>144</td>
<td>1</td>
<td>1</td>
<td>90.6</td>
<td>23.9</td>
<td>45.5</td>
<td>4.9</td>
<td>0.21</td>
</tr>
<tr>
<td>157</td>
<td>1</td>
<td>1</td>
<td>71.9</td>
<td>20.7</td>
<td>41.0</td>
<td>4.8</td>
<td>1.58</td>
</tr>
<tr>
<td>160</td>
<td>1</td>
<td>1</td>
<td>82.5</td>
<td>32.4</td>
<td>55.5</td>
<td>4.8</td>
<td>0.23</td>
</tr>
<tr>
<td>173</td>
<td>1</td>
<td>1</td>
<td>66.5</td>
<td>24.6</td>
<td>43.7</td>
<td>0.0</td>
<td>0.75</td>
</tr>
<tr>
<td>186</td>
<td>1</td>
<td>1</td>
<td>85.8</td>
<td>34.7</td>
<td>56.5</td>
<td>4.9</td>
<td>0.15</td>
</tr>
<tr>
<td>199</td>
<td>1</td>
<td>1</td>
<td>55.6</td>
<td>21.3</td>
<td>43.1</td>
<td>4.1</td>
<td>0.64</td>
</tr>
<tr>
<td>204</td>
<td>1</td>
<td>1</td>
<td>69.6</td>
<td>25.9</td>
<td>46.3</td>
<td>4.7</td>
<td>0.29</td>
</tr>
<tr>
<td>217</td>
<td>1</td>
<td>1</td>
<td>90.0</td>
<td>46.5</td>
<td>69.0</td>
<td>0.0</td>
<td>0.06</td>
</tr>
<tr>
<td>220</td>
<td>1</td>
<td>1</td>
<td>51.7</td>
<td>10.2</td>
<td>28.6</td>
<td>4.1</td>
<td>2.16</td>
</tr>
<tr>
<td>233</td>
<td>1</td>
<td>1</td>
<td>87.2</td>
<td>28.7</td>
<td>48.6</td>
<td>4.2</td>
<td>0.15</td>
</tr>
<tr>
<td>259</td>
<td>1</td>
<td>1</td>
<td>66.2</td>
<td>23.9</td>
<td>40.3</td>
<td>3.8</td>
<td>0.40</td>
</tr>
<tr>
<td>275</td>
<td>1</td>
<td>1</td>
<td>85.7</td>
<td>28.6</td>
<td>51.7</td>
<td>4.7</td>
<td>0.33</td>
</tr>
<tr>
<td>291</td>
<td>1</td>
<td>1</td>
<td>33.0</td>
<td>8.4</td>
<td>24.9</td>
<td>3.9</td>
<td>3.09</td>
</tr>
<tr>
<td>306</td>
<td>1</td>
<td>1</td>
<td>97.5</td>
<td>27.5</td>
<td>47.5</td>
<td>4.5</td>
<td>0.11</td>
</tr>
<tr>
<td>319</td>
<td>1</td>
<td>1</td>
<td>87.5</td>
<td>27.5</td>
<td>47.5</td>
<td>4.5</td>
<td>0.11</td>
</tr>
<tr>
<td>322</td>
<td>1</td>
<td>1</td>
<td>87.5</td>
<td>27.5</td>
<td>47.5</td>
<td>4.5</td>
<td>0.11</td>
</tr>
<tr>
<td>335</td>
<td>1</td>
<td>1</td>
<td>51.3</td>
<td>12.8</td>
<td>28.8</td>
<td>3.9</td>
<td>1.30</td>
</tr>
<tr>
<td>348</td>
<td>1</td>
<td>1</td>
<td>51.3</td>
<td>12.8</td>
<td>28.8</td>
<td>4.2</td>
<td>1.30</td>
</tr>
<tr>
<td>351</td>
<td>1</td>
<td>1</td>
<td>55.4</td>
<td>9.1</td>
<td>26.9</td>
<td>3.8</td>
<td>1.05</td>
</tr>
<tr>
<td>364</td>
<td>1</td>
<td>1</td>
<td>80.3</td>
<td>22.2</td>
<td>41.2</td>
<td>4.4</td>
<td>0.57</td>
</tr>
<tr>
<td>377</td>
<td>1</td>
<td>1</td>
<td>87.5</td>
<td>27.5</td>
<td>47.5</td>
<td>4.5</td>
<td>0.11</td>
</tr>
<tr>
<td>380</td>
<td>1</td>
<td>1</td>
<td>71.4</td>
<td>20.2</td>
<td>37.9</td>
<td>5.0</td>
<td>0.48</td>
</tr>
<tr>
<td>393</td>
<td>1</td>
<td>1</td>
<td>87.5</td>
<td>27.5</td>
<td>47.5</td>
<td>4.5</td>
<td>0.11</td>
</tr>
</tbody>
</table>

SAMPLE SUBGRADE FILE REPORT

EXHIBIT 8
## Texas Flexible Pavement Database
### Serviceability Index File

<table>
<thead>
<tr>
<th>Section</th>
<th>Ident. No</th>
<th>Struc Layr No</th>
<th>Actl Actl Actl Yr Mnth Day Year Count Mean</th>
<th>Stand. Deviat. Val High Low</th>
<th>Sample Serviceability Index File Report</th>
<th>Exhibit 10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>13</td>
<td>2</td>
<td>5 80 9 16 80 8 3.17500 0.38079 2.5 3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>1</td>
<td>4 77 8 5 77 10 3.50000 0.29440 3.0 3.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>1</td>
<td>4 76 7 22 76 10 3.36000 0.35650 2.7 3.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>1</td>
<td>4 75 8 24 75 10 3.42000 0.28210 2.9 3.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>1</td>
<td>4 74 8 27 74 10 3.38000 0.26600 3.0 3.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>1</td>
<td>4 73 7 19 73 10 3.35000 0.32400 2.6 3.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>1</td>
<td>5 80 9 16 80 10 2.52999 0.66508 1.4 3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>1</td>
<td>5 77 8 5 77 10 4.02000 0.12290 3.8 4.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>1</td>
<td>5 76 7 22 76 10 3.95000 0.15810 3.6 4.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>1</td>
<td>5 75 8 24 75 10 3.73000 0.20570 3.3 4.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>1</td>
<td>5 74 8 27 74 4 3.58000 0.26300 3.2 3.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>1</td>
<td>5 73 7 19 73 9 3.74000 0.20700 3.4 4.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>1</td>
<td>5 80 9 16 80 10 2.66000 0.32042 2.1 3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>1</td>
<td>5 77 8 5 77 10 3.30000 0.35590 2.7 3.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>1</td>
<td>5 76 7 22 76 10 3.45000 0.58360 2.1 4.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>1</td>
<td>5 74 10 26 74 9 3.44000 0.23000 2.9 3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>1</td>
<td>10 80 9 16 80 10 3.71000 0.47714 2.7 4.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>1</td>
<td>10 77 3 12 78 9 4.17780 0.26820 3.7 4.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>1</td>
<td>10 76 7 21 76 9 4.13330 0.44160 3.4 4.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>1</td>
<td>10 75 8 28 75 9 3.96670 0.49750 3.1 4.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>1</td>
<td>6 80 9 16 80 10 1.60000 0.30912 1.1 2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>1</td>
<td>6 77 8 5 77 10 2.34000 0.45510 1.4 3.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>1</td>
<td>6 76 7 22 76 9 1.72220 0.43900 0.5 2.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>1</td>
<td>6 75 8 28 75 10 1.86000 0.47190 0.8 2.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>1</td>
<td>6 74 10 26 74 9 2.18000 0.42900 1.2 2.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>1</td>
<td>6 73 7 20 73 10 2.18000 0.53700 1.0 2.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>68</td>
<td>1</td>
<td>4 80 9 16 80 10 1.74000 0.67856 0.9 2.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>68</td>
<td>1</td>
<td>4 77 8 5 77 8 2.25000 0.70100 1.5 3.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>68</td>
<td>1</td>
<td>4 76 7 22 76 8 1.27500 0.70650 0.5 2.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>68</td>
<td>1</td>
<td>4 75 8 28 75 7 1.77140 0.72960 1.0 2.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>68</td>
<td>1</td>
<td>4 74 10 26 74 9 2.03000 0.52700 1.4 2.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>68</td>
<td>1</td>
<td>4 73 1 10 74 10 2.16000 0.56000 1.2 3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1D</td>
<td>Str</td>
<td>Lyr</td>
<td>Date</td>
<td>SSI</td>
<td>SSI</td>
<td>SSI Reading 1</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>---------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Geophone 1 - 7</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>5</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>6</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>68</td>
<td>1</td>
<td>4</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>71</td>
<td>1</td>
<td>5</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>97</td>
<td>1</td>
<td>3</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>102</td>
<td>2</td>
<td>7</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>115</td>
<td>1</td>
<td>4</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>128</td>
<td>3</td>
<td>8</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>144</td>
<td>3</td>
<td>8</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>160</td>
<td>2</td>
<td>6</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>186</td>
<td>2</td>
<td>11</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>199</td>
<td>4</td>
<td>10</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>204</td>
<td>3</td>
<td>8</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>217</td>
<td>2</td>
<td>9</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>220</td>
<td>2</td>
<td>12</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>233</td>
<td>1</td>
<td>6</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>246</td>
<td>2</td>
<td>11</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>259</td>
<td>2</td>
<td>7</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>275</td>
<td>1</td>
<td>5</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>288</td>
<td>3</td>
<td>5</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>291</td>
<td>1</td>
<td>5</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>305</td>
<td>3</td>
<td>9</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>319</td>
<td>2</td>
<td>5</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>322</td>
<td>2</td>
<td>7</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>335</td>
<td>2</td>
<td>11</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>348</td>
<td>2</td>
<td>6</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>351</td>
<td>2</td>
<td>6</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>364</td>
<td>2</td>
<td>7</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>377</td>
<td>2</td>
<td>7</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>380</td>
<td>1</td>
<td>7</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>393</td>
<td>2</td>
<td>10</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>408</td>
<td>1</td>
<td>3</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>411</td>
<td>1</td>
<td>6</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>424</td>
<td>1</td>
<td>5</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>440</td>
<td>2</td>
<td>7</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>453</td>
<td>2</td>
<td>7</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>466</td>
<td>1</td>
<td>5</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>479</td>
<td>1</td>
<td>6</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>482</td>
<td>1</td>
<td>5</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>495</td>
<td>1</td>
<td>5</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>526</td>
<td>1</td>
<td>6</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>539</td>
<td>3</td>
<td>10</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>542</td>
<td>1</td>
<td>4</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>555</td>
<td>1</td>
<td>7</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>568</td>
<td>1</td>
<td>6</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
</tbody>
</table>

SAMPLE FALLING WEIGHT SSI REPORT FILE
EXHIBIT 11
Texas Flexible Pavement Database
Dynaflect Measurement File

<table>
<thead>
<tr>
<th>SID</th>
<th>Struc Layer No.</th>
<th>Numbr Year Month Day Station</th>
<th>Sensor Readings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4 76 8 10 1</td>
<td>0.570 0.540 0.470 0.380 0.320</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4 76 8 10 2</td>
<td>0.620 0.590 0.530 0.430 0.360</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4 76 8 10 3</td>
<td>0.246 0.231 0.189 0.138 0.102</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4 76 8 10 4</td>
<td>0.252 0.228 0.180 0.126 0.090</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4 76 8 10 5</td>
<td>0.560 0.530 0.460 0.370 0.310</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4 76 8 10 6</td>
<td>0.680 0.650 0.560 0.450 0.370</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4 76 8 10 7</td>
<td>0.450 0.430 0.380 0.300 0.240</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4 76 8 10 8</td>
<td>0.470 0.450 0.390 0.320 0.261</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4 76 8 10 9</td>
<td>0.410 0.380 0.320 0.255 0.195</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4 76 8 10 10</td>
<td>0.430 0.400 0.340 0.252 0.192</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4 76 8 10 11</td>
<td>0.360 0.340 0.290 0.222 0.168</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4 76 8 10 12</td>
<td>0.370 0.350 0.300 0.225 0.168</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4 76 8 10 13</td>
<td>0.550 0.530 0.470 0.380 0.300</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4 76 8 10 14</td>
<td>0.680 0.660 0.570 0.440 0.360</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5 76 8 10 1</td>
<td>0.630 0.410 0.210 0.102 0.053</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5 76 8 10 2</td>
<td>0.550 0.400 0.204 0.096 0.053</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5 76 8 10 3</td>
<td>0.470 0.270 0.108 0.046 0.027</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5 76 8 10 4</td>
<td>0.294 0.132 0.033 0.001 0.001</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5 76 8 10 5</td>
<td>0.310 0.126 0.038 0.001 0.001</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5 76 8 10 6</td>
<td>0.270 0.108 0.029 0.001 0.001</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5 76 8 10 7</td>
<td>0.273 0.096 0.028 0.001 0.001</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5 76 8 10 8</td>
<td>0.320 0.165 0.063 0.028 0.001</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5 76 8 10 9</td>
<td>0.650 0.440 0.279 0.183 0.138</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5 76 8 10 10</td>
<td>0.690 0.440 0.249 0.153 0.120</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5 76 8 10 11</td>
<td>0.730 0.450 0.228 0.108 0.050</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5 76 8 10 12</td>
<td>0.660 0.360 0.171 0.078 0.033</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5 76 8 10 13</td>
<td>0.420 0.276 0.126 0.061 0.030</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5 76 8 10 14</td>
<td>0.470 0.297 0.141 0.052 0.026</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>5 76 8 9 1</td>
<td>0.500 0.470 0.440 0.380 0.330</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>5 76 8 9 2</td>
<td>0.530 0.500 0.470 0.400 0.340</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>5 76 8 9 3</td>
<td>0.740 0.700 0.660 0.590 0.530</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>5 76 8 9 4</td>
<td>0.790 0.760 0.720 0.660 0.600</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>5 76 8 9 5</td>
<td>0.530 0.500 0.470 0.420 0.360</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>5 76 8 9 6</td>
<td>0.510 0.480 0.450 0.390 0.350</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>5 76 8 9 7</td>
<td>0.600 0.560 0.520 0.460 0.420</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>5 76 8 9 8</td>
<td>0.550 0.520 0.470 0.400 0.340</td>
</tr>
</tbody>
</table>

SAMPLE DYNAFLECT MEASUREMENT FILE REPORT

EXHIBIT 12

105
<table>
<thead>
<tr>
<th>Section Identification Number</th>
<th>Structure Layer Number</th>
<th>Year</th>
<th>Month</th>
<th>Skid Number</th>
<th>Skid Mean</th>
<th>Skid High</th>
<th>Skid Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1</td>
<td>4</td>
<td>71</td>
<td>4</td>
<td>38</td>
<td>42</td>
<td>32</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4</td>
<td>74</td>
<td>7</td>
<td>39</td>
<td>49</td>
<td>27</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>5</td>
<td>74</td>
<td>8</td>
<td>33</td>
<td>44</td>
<td>22</td>
</tr>
<tr>
<td>84</td>
<td>1</td>
<td>5</td>
<td>75</td>
<td>7</td>
<td>31</td>
<td>39</td>
<td>25</td>
</tr>
<tr>
<td>102</td>
<td>1</td>
<td>6</td>
<td>71</td>
<td>4</td>
<td>31</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>115</td>
<td>1</td>
<td>4</td>
<td>74</td>
<td>8</td>
<td>28</td>
<td>30</td>
<td>23</td>
</tr>
<tr>
<td>115</td>
<td>1</td>
<td>4</td>
<td>75</td>
<td>3</td>
<td>26</td>
<td>34</td>
<td>20</td>
</tr>
<tr>
<td>128</td>
<td>1</td>
<td>6</td>
<td>75</td>
<td>2</td>
<td>32</td>
<td>38</td>
<td>27</td>
</tr>
<tr>
<td>128</td>
<td>1</td>
<td>6</td>
<td>76</td>
<td>9</td>
<td>23</td>
<td>26</td>
<td>20</td>
</tr>
<tr>
<td>131</td>
<td>1</td>
<td>4</td>
<td>76</td>
<td>9</td>
<td>55</td>
<td>65</td>
<td>41</td>
</tr>
<tr>
<td>144</td>
<td>2</td>
<td>7</td>
<td>75</td>
<td>1</td>
<td>16</td>
<td>24</td>
<td>8</td>
</tr>
<tr>
<td>160</td>
<td>1</td>
<td>5</td>
<td>74</td>
<td>11</td>
<td>36</td>
<td>42</td>
<td>32</td>
</tr>
<tr>
<td>160</td>
<td>1</td>
<td>5</td>
<td>76</td>
<td>8</td>
<td>28</td>
<td>31</td>
<td>25</td>
</tr>
<tr>
<td>173</td>
<td>3</td>
<td>8</td>
<td>76</td>
<td>8</td>
<td>26</td>
<td>48</td>
<td>12</td>
</tr>
<tr>
<td>186</td>
<td>1</td>
<td>8</td>
<td>74</td>
<td>11</td>
<td>33</td>
<td>36</td>
<td>29</td>
</tr>
<tr>
<td>186</td>
<td>1</td>
<td>8</td>
<td>76</td>
<td>8</td>
<td>30</td>
<td>44</td>
<td>19</td>
</tr>
<tr>
<td>220</td>
<td>1</td>
<td>11</td>
<td>75</td>
<td>7</td>
<td>42</td>
<td>60</td>
<td>25</td>
</tr>
<tr>
<td>220</td>
<td>1</td>
<td>11</td>
<td>75</td>
<td>10</td>
<td>43</td>
<td>50</td>
<td>37</td>
</tr>
<tr>
<td>233</td>
<td>1</td>
<td>6</td>
<td>75</td>
<td>7</td>
<td>37</td>
<td>57</td>
<td>20</td>
</tr>
<tr>
<td>233</td>
<td>1</td>
<td>6</td>
<td>75</td>
<td>12</td>
<td>35</td>
<td>42</td>
<td>28</td>
</tr>
<tr>
<td>246</td>
<td>0</td>
<td>0</td>
<td>74</td>
<td>8</td>
<td>16</td>
<td>19</td>
<td>13</td>
</tr>
<tr>
<td>259</td>
<td>1</td>
<td>6</td>
<td>75</td>
<td>12</td>
<td>30</td>
<td>42</td>
<td>21</td>
</tr>
<tr>
<td>275</td>
<td>1</td>
<td>5</td>
<td>76</td>
<td>4</td>
<td>35</td>
<td>48</td>
<td>21</td>
</tr>
<tr>
<td>291</td>
<td>1</td>
<td>6</td>
<td>76</td>
<td>3</td>
<td>33</td>
<td>43</td>
<td>20</td>
</tr>
<tr>
<td>335</td>
<td>2</td>
<td>11</td>
<td>76</td>
<td>8</td>
<td>48</td>
<td>49</td>
<td>46</td>
</tr>
<tr>
<td>351</td>
<td>2</td>
<td>6</td>
<td>76</td>
<td>7</td>
<td>40</td>
<td>46</td>
<td>34</td>
</tr>
<tr>
<td>424</td>
<td>1</td>
<td>5</td>
<td>76</td>
<td>8</td>
<td>47</td>
<td>50</td>
<td>45</td>
</tr>
<tr>
<td>440</td>
<td>1</td>
<td>6</td>
<td>75</td>
<td>4</td>
<td>39</td>
<td>54</td>
<td>22</td>
</tr>
<tr>
<td>440</td>
<td>1</td>
<td>6</td>
<td>75</td>
<td>4</td>
<td>40</td>
<td>52</td>
<td>24</td>
</tr>
<tr>
<td>453</td>
<td>1</td>
<td>6</td>
<td>76</td>
<td>1</td>
<td>36</td>
<td>55</td>
<td>18</td>
</tr>
<tr>
<td>453</td>
<td>1</td>
<td>6</td>
<td>76</td>
<td>1</td>
<td>34</td>
<td>51</td>
<td>19</td>
</tr>
<tr>
<td>479</td>
<td>1</td>
<td>6</td>
<td>74</td>
<td>6</td>
<td>25</td>
<td>29</td>
<td>18</td>
</tr>
<tr>
<td>479</td>
<td>1</td>
<td>6</td>
<td>74</td>
<td>6</td>
<td>24</td>
<td>33</td>
<td>19</td>
</tr>
<tr>
<td>526</td>
<td>1</td>
<td>6</td>
<td>75</td>
<td>4</td>
<td>39</td>
<td>47</td>
<td>32</td>
</tr>
<tr>
<td>526</td>
<td>1</td>
<td>6</td>
<td>75</td>
<td>4</td>
<td>41</td>
<td>48</td>
<td>33</td>
</tr>
</tbody>
</table>

SAMPLE SKID MEASUREMENT DATA REPORT

EXHIBIT 13
<table>
<thead>
<tr>
<th>Section Identification Number</th>
<th>Average Annual Daily Traffic (ONE-WAY)</th>
<th>Annual 18KIP Equivalent Axis Loads (ONE-WAY)</th>
<th>Percent Trucks</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1933</td>
<td>386</td>
<td>54259</td>
</tr>
<tr>
<td>13</td>
<td>1934</td>
<td>486</td>
<td>63188</td>
</tr>
<tr>
<td>13</td>
<td>1935</td>
<td>586</td>
<td>71027</td>
</tr>
<tr>
<td>13</td>
<td>1936</td>
<td>686</td>
<td>78009</td>
</tr>
<tr>
<td>13</td>
<td>1937</td>
<td>702</td>
<td>79091</td>
</tr>
<tr>
<td>13</td>
<td>1938</td>
<td>719</td>
<td>80187</td>
</tr>
<tr>
<td>13</td>
<td>1939</td>
<td>736</td>
<td>81233</td>
</tr>
<tr>
<td>13</td>
<td>1940</td>
<td>792</td>
<td>84689</td>
</tr>
<tr>
<td>13</td>
<td>1941</td>
<td>849</td>
<td>87994</td>
</tr>
<tr>
<td>13</td>
<td>1942</td>
<td>906</td>
<td>91108</td>
</tr>
<tr>
<td>13</td>
<td>1943</td>
<td>912</td>
<td>91457</td>
</tr>
<tr>
<td>13</td>
<td>1944</td>
<td>919</td>
<td>91830</td>
</tr>
<tr>
<td>13</td>
<td>1945</td>
<td>926</td>
<td>92175</td>
</tr>
<tr>
<td>13</td>
<td>1946</td>
<td>986</td>
<td>95269</td>
</tr>
<tr>
<td>13</td>
<td>1947</td>
<td>1046</td>
<td>98217</td>
</tr>
<tr>
<td>13</td>
<td>1948</td>
<td>1106</td>
<td>101031</td>
</tr>
<tr>
<td>13</td>
<td>1949</td>
<td>1206</td>
<td>105453</td>
</tr>
<tr>
<td>13</td>
<td>1950</td>
<td>1306</td>
<td>109576</td>
</tr>
<tr>
<td>13</td>
<td>1951</td>
<td>1406</td>
<td>113436</td>
</tr>
<tr>
<td>13</td>
<td>1952</td>
<td>1494</td>
<td>116656</td>
</tr>
<tr>
<td>13</td>
<td>1953</td>
<td>1582</td>
<td>119692</td>
</tr>
<tr>
<td>13</td>
<td>1954</td>
<td>1670</td>
<td>122594</td>
</tr>
<tr>
<td>13</td>
<td>1955</td>
<td>1660</td>
<td>122274</td>
</tr>
<tr>
<td>13</td>
<td>1956</td>
<td>1650</td>
<td>121951</td>
</tr>
<tr>
<td>13</td>
<td>1957</td>
<td>1640</td>
<td>121627</td>
</tr>
<tr>
<td>13</td>
<td>1958</td>
<td>1536</td>
<td>118107</td>
</tr>
<tr>
<td>13</td>
<td>1959</td>
<td>1430</td>
<td>114364</td>
</tr>
<tr>
<td>13</td>
<td>1960</td>
<td>1326</td>
<td>110368</td>
</tr>
<tr>
<td>13</td>
<td>1961</td>
<td>1349</td>
<td>111286</td>
</tr>
<tr>
<td>13</td>
<td>1962</td>
<td>1372</td>
<td>112171</td>
</tr>
<tr>
<td>13</td>
<td>1963</td>
<td>1396</td>
<td>113061</td>
</tr>
<tr>
<td>13</td>
<td>1964</td>
<td>1502</td>
<td>116939</td>
</tr>
<tr>
<td>13</td>
<td>1965</td>
<td>1609</td>
<td>120593</td>
</tr>
<tr>
<td>13</td>
<td>1966</td>
<td>1716</td>
<td>124016</td>
</tr>
<tr>
<td>13</td>
<td>1967</td>
<td>1806</td>
<td>126773</td>
</tr>
</tbody>
</table>

SAMPLE TRAFFIC FILE REPORT

EXHIBIT 14

<table>
<thead>
<tr>
<th>County No</th>
<th>Thornthwaite Index Mean</th>
<th>Thornthwaite Index Std. Dev.</th>
<th>Thornthwaite Index Years</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.510</td>
<td>26.102</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>-39.322</td>
<td>7.059</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>11.692</td>
<td>23.917</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>-10.252</td>
<td>18.861</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>-16.158</td>
<td>12.329</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>-17.534</td>
<td>11.719</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>-22.308</td>
<td>16.070</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>4.458</td>
<td>23.532</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>-21.861</td>
<td>8.537</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>-11.434</td>
<td>17.491</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>-10.698</td>
<td>18.953</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>-18.964</td>
<td>10.214</td>
<td>20</td>
</tr>
<tr>
<td>13</td>
<td>-15.863</td>
<td>13.017</td>
<td>20</td>
</tr>
<tr>
<td>14</td>
<td>-12.490</td>
<td>16.739</td>
<td>20</td>
</tr>
<tr>
<td>15</td>
<td>-15.722</td>
<td>16.517</td>
<td>20</td>
</tr>
<tr>
<td>16</td>
<td>-6.667</td>
<td>24.871</td>
<td>20</td>
</tr>
<tr>
<td>17</td>
<td>-30.379</td>
<td>7.260</td>
<td>20</td>
</tr>
<tr>
<td>18</td>
<td>-10.233</td>
<td>16.502</td>
<td>20</td>
</tr>
<tr>
<td>19</td>
<td>49.059</td>
<td>29.631</td>
<td>20</td>
</tr>
<tr>
<td>20</td>
<td>38.530</td>
<td>40.910</td>
<td>20</td>
</tr>
<tr>
<td>21</td>
<td>4.594</td>
<td>23.644</td>
<td>20</td>
</tr>
<tr>
<td>22</td>
<td>-31.617</td>
<td>7.766</td>
<td>20</td>
</tr>
<tr>
<td>23</td>
<td>-17.867</td>
<td>12.314</td>
<td>20</td>
</tr>
<tr>
<td>24</td>
<td>-28.007</td>
<td>14.497</td>
<td>19</td>
</tr>
<tr>
<td>25</td>
<td>-19.138</td>
<td>12.262</td>
<td>20</td>
</tr>
<tr>
<td>26</td>
<td>2.098</td>
<td>24.167</td>
<td>20</td>
</tr>
<tr>
<td>27</td>
<td>-13.876</td>
<td>15.861</td>
<td>20</td>
</tr>
<tr>
<td>28</td>
<td>-6.157</td>
<td>19.694</td>
<td>20</td>
</tr>
<tr>
<td>29</td>
<td>1.149</td>
<td>25.127</td>
<td>20</td>
</tr>
<tr>
<td>30</td>
<td>-19.656</td>
<td>16.029</td>
<td>20</td>
</tr>
<tr>
<td>31</td>
<td>-27.658</td>
<td>11.298</td>
<td>20</td>
</tr>
<tr>
<td>32</td>
<td>47.402</td>
<td>42.416</td>
<td>20</td>
</tr>
<tr>
<td>33</td>
<td>-18.714</td>
<td>11.884</td>
<td>20</td>
</tr>
<tr>
<td>34</td>
<td>49.248</td>
<td>37.978</td>
<td>20</td>
</tr>
<tr>
<td>35</td>
<td>-22.058</td>
<td>10.633</td>
<td>20</td>
</tr>
</tbody>
</table>
**ENVIRONMENT - 20 YEAR SUMMARY (1955-1974)**

<table>
<thead>
<tr>
<th>MONTH</th>
<th>JAN</th>
<th>FEB</th>
<th>MAR</th>
<th>APR</th>
<th>MAY</th>
<th>JUN</th>
<th>JUL</th>
<th>AUG</th>
<th>SEP</th>
<th>OCT</th>
<th>NOV</th>
<th>DEC</th>
<th>ANN</th>
</tr>
</thead>
<tbody>
<tr>
<td>THORNTHWAITE INDEX</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>41.6</td>
</tr>
<tr>
<td>MEAN TEMPERATURE</td>
<td>41</td>
<td>45</td>
<td>53</td>
<td>63</td>
<td>71</td>
<td>78</td>
<td>83</td>
<td>82</td>
<td>76</td>
<td>66</td>
<td>53</td>
<td>45</td>
<td>63.1</td>
</tr>
<tr>
<td>PRECIPITATION</td>
<td>2.2</td>
<td>2.7</td>
<td>3.3</td>
<td>5.1</td>
<td>5.2</td>
<td>3.5</td>
<td>3.0</td>
<td>2.2</td>
<td>8.5</td>
<td>4.1</td>
<td>3.0</td>
<td>3.0</td>
<td>42.6</td>
</tr>
<tr>
<td>WET F-T CYCLES</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL F-T CYCLES</td>
<td>13</td>
<td>9</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>11</td>
<td>40.2</td>
</tr>
<tr>
<td>DIST TEMP CONSTANT</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>21</td>
</tr>
</tbody>
</table>

**TRAFFIC**

<table>
<thead>
<tr>
<th>Year</th>
<th>truck</th>
<th>vehicles</th>
<th>axles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1986</td>
<td>7800</td>
<td>47966822</td>
<td>14603862</td>
</tr>
</tbody>
</table>

**SERVICEABILITY INDEX**

<table>
<thead>
<tr>
<th>Year</th>
<th>VR</th>
<th>MEAN</th>
<th>STD</th>
<th>DEV</th>
<th>N</th>
<th>CV</th>
<th>LOW</th>
<th>HIGH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1986</td>
<td>7800</td>
<td>2.66</td>
<td>.320</td>
<td>10</td>
<td>12.0</td>
<td>2.10</td>
<td>3.20</td>
<td></td>
</tr>
<tr>
<td>1994-1995</td>
<td>77</td>
<td>3.30</td>
<td>.356</td>
<td>10</td>
<td>10.8</td>
<td>2.70</td>
<td>3.80</td>
<td></td>
</tr>
<tr>
<td>1995-1996</td>
<td>76</td>
<td>3.45</td>
<td>.584</td>
<td>10</td>
<td>16.9</td>
<td>2.10</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>1996-1997</td>
<td>74</td>
<td>3.44</td>
<td>.230</td>
<td>9</td>
<td>6.7</td>
<td>2.90</td>
<td>3.60</td>
<td></td>
</tr>
</tbody>
</table>

**SKID NUMBER**

<table>
<thead>
<tr>
<th>Date</th>
<th>AVG</th>
<th>LOW</th>
<th>HIGH</th>
</tr>
</thead>
<tbody>
<tr>
<td>7/74</td>
<td>33</td>
<td>30</td>
<td>36</td>
</tr>
</tbody>
</table>

**DEFLECTION (MEAN VARIABLES)**

<table>
<thead>
<tr>
<th>Date</th>
<th>D</th>
<th>W1</th>
<th>W2</th>
<th>W3</th>
<th>W4</th>
<th>W5</th>
<th>W6</th>
<th>W7</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/9/76</td>
<td>D</td>
<td>0.56</td>
<td>0.53</td>
<td>0.50</td>
<td>0.44</td>
<td>0.39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0/0/0</td>
<td>F</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SAMPLE SUMMARY REPORT**

**EXHIBIT 17**
**LOCATION**

- **SECTION ID NO:** 39
- **DISTRICT NO:** 1
- **COUNTY NO/NAME:** 117/HUNT
- **CONTROL-SECTION:** 9-13
- **HIGHWAY:** IH 30
- **MILE POINTS:** 27.800-29.800
- **LANE:** R
- **MILE POST:** 106+00 TO 109+00
- **PREVIOUS SID:** -
- **NEXT SID:** -
- **FUNCTIONAL CLASS:** 0
- **TYPE OF PAVEMENT:** PCC
  - 2.5 <= HMAC < 5.5
    - (NO SEALS)
- **INACTIVE SID:** NO
- **NUMBER OF LANES:** 2

**GEOMETRIC AND SHOULDER INFORMATION**

<table>
<thead>
<tr>
<th>STRUCTURE</th>
<th>TYPE</th>
<th>OUTSIDE</th>
<th>SHOULDER</th>
<th>SHOULDER</th>
<th>SHOULDER</th>
<th>SHOULDER</th>
<th>SHOULDER</th>
<th>WIDENING</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER</td>
<td>PAVEMENT</td>
<td>WIDTH</td>
<td>WIDTH</td>
<td>TYPE</td>
<td>TYPE</td>
<td>THICKNESS</td>
<td>THICKNESS</td>
<td>FLAG</td>
</tr>
<tr>
<td>1</td>
<td>34</td>
<td>12.0</td>
<td>17.0</td>
<td>2</td>
<td>21</td>
<td>0.00</td>
<td>0.00</td>
<td>1</td>
</tr>
</tbody>
</table>

**SAMPLE INVENTORY UPDATE REPORT**
**EXHIBIT 18**
<table>
<thead>
<tr>
<th>LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECTION ID NO:  39</td>
</tr>
<tr>
<td>DISTRICT NO:  1</td>
</tr>
<tr>
<td>COUNTY NO/NAME: 117/HUNT</td>
</tr>
<tr>
<td>CONTROL SECTION: 9-13</td>
</tr>
<tr>
<td>HIGHWAY:  IH 30</td>
</tr>
<tr>
<td>MILE POINTS:  27.800-29.800</td>
</tr>
<tr>
<td>LANE:  R</td>
</tr>
<tr>
<td>MILE POST:  106+00 TO 109+00</td>
</tr>
<tr>
<td>PREVIOUS SID:  -</td>
</tr>
<tr>
<td>NEXT SID:  -</td>
</tr>
<tr>
<td>FUNCTIONAL CLASS:  0</td>
</tr>
<tr>
<td>TYPE OF PAVEMENT:  PCC</td>
</tr>
</tbody>
</table>

2.5 <= HMAC < 5.5  
(NO SEALS)  
INACTIVE SID:  NO  
NUMBER OF LANES:  2

<table>
<thead>
<tr>
<th>GEOMETRIC AND SHOULDER INFORMATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRUCTURE TYPE</td>
</tr>
<tr>
<td>NUMBER PAVEMENT WIDTH</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>---</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LAYER THICKNESS ACROSS THE ROAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRUCTURE</td>
</tr>
<tr>
<td>NUMBER</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>---</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STRUCTURAL SECTION</th>
<th>THICK</th>
<th>WIDENING AGG</th>
<th>ADMIXTURE</th>
<th>APPL % PASSING</th>
<th>PERM,</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRUCTURE LAYER DESCRIPTION</td>
<td>CENT</td>
<td>MATERIAL TYPE</td>
<td>DATE</td>
<td>DATE</td>
<td>RATE</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>

SAMPLE INVENTORY UPDATE REPORT  
EXHIBIT 18 (Continued)
<table>
<thead>
<tr>
<th>County Number</th>
<th>County Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ANDERSON</td>
</tr>
<tr>
<td>2</td>
<td>ANDREWS</td>
</tr>
<tr>
<td>3</td>
<td>ANGELINA</td>
</tr>
<tr>
<td>4</td>
<td>ARANSAS</td>
</tr>
<tr>
<td>5</td>
<td>ARCHER</td>
</tr>
<tr>
<td>6</td>
<td>ARMSTRONG</td>
</tr>
<tr>
<td>7</td>
<td>ATASCOSA</td>
</tr>
<tr>
<td>8</td>
<td>AUSTIN</td>
</tr>
<tr>
<td>9</td>
<td>BAILEY</td>
</tr>
<tr>
<td>10</td>
<td>BANDERA</td>
</tr>
<tr>
<td>11</td>
<td>BASTROP</td>
</tr>
<tr>
<td>12</td>
<td>BAYLOR</td>
</tr>
<tr>
<td>13</td>
<td>BEE</td>
</tr>
<tr>
<td>14</td>
<td>BELL</td>
</tr>
<tr>
<td>15</td>
<td>BEXAR</td>
</tr>
<tr>
<td>16</td>
<td>BLANCO</td>
</tr>
<tr>
<td>17</td>
<td>BORDEN</td>
</tr>
<tr>
<td>18</td>
<td>BOSQUE</td>
</tr>
<tr>
<td>19</td>
<td>BOWIE</td>
</tr>
<tr>
<td>20</td>
<td>BRAZORIA</td>
</tr>
<tr>
<td>21</td>
<td>BRAZOS</td>
</tr>
<tr>
<td>22</td>
<td>BREWSTER</td>
</tr>
<tr>
<td>23</td>
<td>BRISCOE</td>
</tr>
<tr>
<td>24</td>
<td>BROOKS</td>
</tr>
<tr>
<td>25</td>
<td>BROWN</td>
</tr>
<tr>
<td>26</td>
<td>BURLESON</td>
</tr>
<tr>
<td>27</td>
<td>BURNET</td>
</tr>
<tr>
<td>28</td>
<td>CALDWELL</td>
</tr>
<tr>
<td>29</td>
<td>CALHOUN</td>
</tr>
<tr>
<td>30</td>
<td>CALHAN</td>
</tr>
<tr>
<td>31</td>
<td>CAMERON</td>
</tr>
<tr>
<td>32</td>
<td>CAMP</td>
</tr>
<tr>
<td>33</td>
<td>CARSON</td>
</tr>
<tr>
<td>34</td>
<td>CASS</td>
</tr>
<tr>
<td>35</td>
<td>CASTRO</td>
</tr>
</tbody>
</table>

SAMPLE COUNTY NAME TABLE REPORT

EXHIBIT 19
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HOT MIX - HOT LAID</td>
<td>HMAC</td>
<td>S</td>
</tr>
<tr>
<td>2</td>
<td>HOT MIX-COLD LAID</td>
<td>HMCL</td>
<td>S</td>
</tr>
<tr>
<td>4</td>
<td>COLD MIX ROCK ASPH</td>
<td>CMRA</td>
<td>S</td>
</tr>
<tr>
<td>5</td>
<td>ONE COURSE SURF TRT</td>
<td>ST</td>
<td>S</td>
</tr>
<tr>
<td>6</td>
<td>TWO COURSE SURF TRT</td>
<td>DST</td>
<td>S</td>
</tr>
<tr>
<td>7</td>
<td>THREE COURSE SURF TRT</td>
<td>TST</td>
<td>S</td>
</tr>
<tr>
<td>9</td>
<td>RUBBER ASPH CONCRETE</td>
<td></td>
<td>S</td>
</tr>
<tr>
<td>10</td>
<td>OPEN GRADE FRIC COUR</td>
<td>FC</td>
<td>S</td>
</tr>
<tr>
<td>11</td>
<td>SEAL COAT - REGULAR</td>
<td>SC</td>
<td>S</td>
</tr>
<tr>
<td>16</td>
<td>BLACK BASE</td>
<td>ASB</td>
<td>S</td>
</tr>
<tr>
<td>17</td>
<td>PORTLAND CEMENT CONC</td>
<td>PCC</td>
<td>S</td>
</tr>
<tr>
<td>18</td>
<td>BLANK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>FLEXIBLE BASE</td>
<td>FB</td>
<td>B</td>
</tr>
<tr>
<td>22</td>
<td>LIME STABILIZED</td>
<td>LSB</td>
<td>B</td>
</tr>
<tr>
<td>23</td>
<td>CEMENT STABILIZED</td>
<td>CSB</td>
<td>B</td>
</tr>
<tr>
<td>24</td>
<td>ASPHALT STAB BASE</td>
<td>ASB</td>
<td>B</td>
</tr>
<tr>
<td>25</td>
<td>ASPHALT BASE ROAD MIX</td>
<td>ARM</td>
<td>B</td>
</tr>
<tr>
<td>27</td>
<td>FABRIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>FLEXIBLE</td>
<td>FLEX</td>
<td>SB</td>
</tr>
<tr>
<td>32</td>
<td>LIME STABIL SUBGRADE</td>
<td>LSS</td>
<td>SB</td>
</tr>
<tr>
<td>33</td>
<td>CEMENT STABIL SUBG</td>
<td>CSS</td>
<td>SB</td>
</tr>
<tr>
<td>41</td>
<td>GRAVEL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>SAND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>SILT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>CLAY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>PEAT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SAMPLE MATERIAL TYPE CLASSIFICATION TABLE REPORT

EXHIBIT 20
<table>
<thead>
<tr>
<th>Pavement Code</th>
<th>Type of Base</th>
<th>Surface Thickness</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GRANULAR BASE</td>
<td>SURFACE TREATED</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>GRANULAR BASE</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(NO SEALS)</td>
</tr>
<tr>
<td>3</td>
<td>GRANULAR BASE</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(WITH SEALS)</td>
</tr>
<tr>
<td>4</td>
<td>GRANULAR BASE</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(NO SEALS)</td>
</tr>
<tr>
<td>5</td>
<td>GRANULAR BASE</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(WITH SEALS)</td>
</tr>
<tr>
<td>6</td>
<td>GRANULAR BASE</td>
<td>HMAC =&gt; 5.5</td>
<td>(NO SEALS)</td>
</tr>
<tr>
<td>7</td>
<td>GRANULAR BASE</td>
<td>HMAC =&gt; 5.5</td>
<td>(WITH SEALS)</td>
</tr>
<tr>
<td>11</td>
<td>STABILIZED (CEMENT/LIME)</td>
<td>SURFACE TREATED</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>STABILIZED (CEMENT/LIME)</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(NO SEALS)</td>
</tr>
<tr>
<td>13</td>
<td>STABILIZED (CEMENT/LIME)</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(WITH SEALS)</td>
</tr>
<tr>
<td>14</td>
<td>STABILIZED (CEMENT/LIME)</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(NO SEALS)</td>
</tr>
<tr>
<td>15</td>
<td>STABILIZED (CEMENT/LIME)</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(WITH SEALS)</td>
</tr>
<tr>
<td>16</td>
<td>STABILIZED (CEMENT/LIME)</td>
<td>HMAC =&gt; 5.5</td>
<td>(NO SEALS)</td>
</tr>
<tr>
<td>17</td>
<td>STABILIZED (CEMENT/LIME)</td>
<td>HMAC =&gt; 5.5</td>
<td>(WITH SEALS)</td>
</tr>
<tr>
<td>21</td>
<td>ASPHALT STABILIZED BASE</td>
<td>SURFACE TREATED</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>ASPHALT STABILIZED BASE</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(NO SEALS)</td>
</tr>
<tr>
<td>23</td>
<td>ASPHALT STABILIZED BASE</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(WITH SEALS)</td>
</tr>
<tr>
<td>24</td>
<td>ASPHALT STABILIZED BASE</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(NO SEALS)</td>
</tr>
<tr>
<td>25</td>
<td>ASPHALT STABILIZED BASE</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(WITH SEALS)</td>
</tr>
<tr>
<td>26</td>
<td>ASPHALT STABILIZED BASE</td>
<td>HMAC =&gt; 5.5</td>
<td>(NO SEALS)</td>
</tr>
<tr>
<td>27</td>
<td>ASPHALT STABILIZED BASE</td>
<td>HMAC =&gt; 5.5</td>
<td>(WITH SEALS)</td>
</tr>
<tr>
<td>31</td>
<td>PCC</td>
<td>SURFACE TREATED</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>PCC</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(NO SEALS)</td>
</tr>
<tr>
<td>33</td>
<td>PCC</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(WITH SEALS)</td>
</tr>
<tr>
<td>34</td>
<td>PCC</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(NO SEALS)</td>
</tr>
<tr>
<td>35</td>
<td>PCC</td>
<td>HMAC &lt; 2.5&quot;</td>
<td>(WITH SEALS)</td>
</tr>
<tr>
<td>36</td>
<td>PCC</td>
<td>HMAC =&gt; 5.5</td>
<td>(NO SEALS)</td>
</tr>
<tr>
<td>37</td>
<td>PCC</td>
<td>HMAC =&gt; 5.5</td>
<td>(WITH SEALS)</td>
</tr>
<tr>
<td>District</td>
<td>Temperature Constant</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>19</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Texas Flexible Pavement Database
Layer Description Table

<table>
<thead>
<tr>
<th>Layer Code</th>
<th>Short Code</th>
<th>Layer Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OVLY</td>
<td>Overlay</td>
</tr>
<tr>
<td>2</td>
<td>SC</td>
<td>Seal Coat</td>
</tr>
<tr>
<td>3</td>
<td>OS</td>
<td>Original Surface</td>
</tr>
<tr>
<td>4</td>
<td>HMAC</td>
<td>HMAC Layer</td>
</tr>
<tr>
<td>5</td>
<td>BSLY</td>
<td>Base Layer</td>
</tr>
<tr>
<td>6</td>
<td>SBLY</td>
<td>Subbase Layer</td>
</tr>
<tr>
<td>7</td>
<td>SBGR</td>
<td>Subgrade</td>
</tr>
<tr>
<td>8</td>
<td>INTL</td>
<td>Interlayer</td>
</tr>
<tr>
<td>9</td>
<td>PRFC</td>
<td>Porous Friction Course</td>
</tr>
<tr>
<td>10</td>
<td>ST</td>
<td>Surface Treatment</td>
</tr>
<tr>
<td>11</td>
<td>EMBK</td>
<td>Embankment (Fill)</td>
</tr>
<tr>
<td>12</td>
<td>RCSF</td>
<td>Recycle Surface</td>
</tr>
<tr>
<td>13</td>
<td>PMSF</td>
<td>Partially Milled Surface</td>
</tr>
<tr>
<td>14</td>
<td>FABR</td>
<td>Fabric</td>
</tr>
</tbody>
</table>

SAMPLE LAYER DESCRIPTION TABLE REPORT

EXHIBIT 23
Texas Flexible Pavement Database Conversion
Widening Table

<table>
<thead>
<tr>
<th>Widening Code</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No Widening</td>
<td>The center thickness can be used</td>
</tr>
<tr>
<td>1</td>
<td>Widening Present</td>
<td>The center thickness can be used for Deflection data - materials have been added to the shoulder</td>
</tr>
<tr>
<td></td>
<td>Special Widening</td>
<td>The center thickness cannot be the material type changed in the middle of the lane.</td>
</tr>
</tbody>
</table>

SAMPLE WIDENING TABLE REPORT

EXHIBIT 24
<table>
<thead>
<tr>
<th>Code</th>
<th>Code Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interstate</td>
</tr>
<tr>
<td>2</td>
<td>Other Urban Freeway and Expressway</td>
</tr>
<tr>
<td>3</td>
<td>Rural or Urban Principal Arterials</td>
</tr>
<tr>
<td>4</td>
<td>Minor Arterial Road or Street</td>
</tr>
<tr>
<td>5</td>
<td>Rural Major or Urban Collector Street</td>
</tr>
<tr>
<td>6</td>
<td>Rural Minor Collectors</td>
</tr>
<tr>
<td>7</td>
<td>Local Road or Street</td>
</tr>
</tbody>
</table>
* SUBSYSTEM: REPORTS MAIN MENU
* PROGRAM NAME: REPORTS.PRG 05/25/88
* UPDATED ON: 09/04/88
* PROJECT 2456 = TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO PRINT REPORTS FOR THE FOLLOWING:
  * - Monitoring data
  * - Environment data
  * - Inventory data
  * - Traffic data
  * - Inventory Update forms
  * - Summary report
  * - Tables

ON ERROR DO \PAVEDB\REPORTS\ERR_PRINT
SET PRINT TO &PRINT1
SET PRINT TO &PRINT2
ON ERROR DO \PAVEDB\ERROR
MREPEAT2 = .T.
DO WHILE MREPEAT2
  * set parameters and initialize variables
  STORE " " TO REPTPICK

  * set up the report menu screen and do the loop until REPTPICK is 1-7
  DO WHILE .NOT. (REPTPICK $ '1234567')
  @ 0, 0 CLEAR
  @ 4, 22 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE 2.0"
  @ 5, 33 SAY "Reports"
  @ 8, 16 SAY "1 - Summary"
  @ 9, 16 SAY "2 - Inventory Update Forms"
  @ 10, 16 SAY "3 - Inventory Data"
  @ 11, 16 SAY "4 - Monitoring Data"
  @ 12, 16 SAY "5 - Traffic Data"
  @ 13, 16 SAY "6 - Environmental Data"
  @ 14, 16 SAY "7 - Tables"
  @ 17, 36 SAY "OPTION ———>"  
  @ 17, 50 GET REPTPICK
  @ 3, 9 TO 18, 65 DOUBLE
  READ

  IF READKEY() = 12
    RETURN TO MASTER
  ENDIF
ENDDO

* according to the response received from the report menu, the appropriate
* commands are executed
IF MREPEAT2
DO CASE
* print summary report
CASE REPTPICK = "1"
   DO \PAVED\EDITUPDT\TBLFILE
      close all
   IF .NOT. FILE(\PAVED\FILES\STRUCSEC.VUE)
      CLEAR
      ? "STRUCTURAL SECTION VIEW FILE (STRUCSEC.VUE) not found. Please check . . . ."
      WAIT
      RETURN TO MASTER
   ENDIF
   SET SAFETY OFF
   CLEAR
   REPTCOLL = "$"
   DO WHILE .NOT. (REPTCOLL $ '123')
      @ 4, 22 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE 2.1"
      @ 5, 34 SAY "REPORTS"
      @ 6, 30 SAY "Summary Report"
      @ 9, 16 SAY "1 - By SID Number"
      @ 10, 16 SAY "2 - By District"
      @ 11, 16 SAY "3 - ALL SID Numbers"
      @ 15, 41 SAY "OPTION ==========
      @ 15, 55 GET REPTCOLL
      @ 3, 9 TO 17, 65 DOUBLE READ
      IF READKEY() = 12
         EXIT
      ENDIF
   ENDDO
   DO CASE
   CASE REPTCOLL = "3"
      * run program to print report for all the SID numbers
   DO \PAVED\REPORTS\SUMMALL
      CLEAR TYPE
   CASE REPTCOLL = "2"
      * run program to print report for a certain district
   DO \PAVED\REPORTS\SUMMDIST
      CLEAR TYPE
   CASE REPTCOLL = "1"
      * run program to print report for one SID number
   DO \PAVED\REPORTS\SUMMSID
      CLEAR TYPE
   ENDCASE
   CLEAR
   SET SAFETY ON
* print inventory update forms for a single SID number or for a complete district if choice 2 is chosen
CASE REPTPICK = "2"
   DO \PAVED\EDITUPDT\TBLFILE
      close all
   IF .NOT. FILE(\PAVED\FILES\STRUCSEC2.VUE)
CLEAR
? "STRUCTURAL SECTION VIEW FILE (STROSEC2.VUE) not found. Please check ..."
WAIT
RETURN TO MASTER
ENDIF
SET SAFETY OFF
CLEAR
REPTCOLL = ""
DO WHILE .NOT. (REPTCOLL $ '123')
  @ 4, 22 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE 2.2"
  @ 5, 34 SAY "REPORTS"
  @ 6, 27 SAY "Inventory Update Forms"
  @ 10, 16 SAY "1 - By District"
  @ 11, 16 SAY "2 - By SID Number"
  @ 12, 16 SAY "3 - All SID Numbers"
  @ 15, 41 SAY "OPTION ———>
  @ 15, 55 GET REPTCOLL
  @ 3, 9 TO 17, 65 DOUBLE READ
  IF READKEY() = 12
    EXIT
  ENDIF
ENDDO
DO CASE
  CASE REPTCOLL = "1"
    DO \PAVEDB\REPORTS\SUM2DIST
      CLEA TYPE
    CASE REPTCOLL = "2"
    DO \PAVEDB\REPORTS\SUM2SID
      CLEA TYPE
    CASE REPTCOLL = "3"
    DO \PAVEDB\REPORTS\SUM2FILE
      CLEA TYPE
ENDCASE
SET SAFETY ON

* If choice 3 is chosen, print one of the 8 reports for the Inventory data
CASE REPTPICK = "3"
  MREPEAT = .T.
  DO WHILE MREPEAT
    CLEAR
    SET PRINT TO &IMPORT2
    REPTCOLL = ""
    DO WHILE .NOT. (REPTCOLL $ '12345678')
      @ 3, 21 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE 2.3"
      @ 4, 34 SAY "REPORTS"
      @ 5, 30 SAY "Inventory Data"
      @ 8, 16 SAY "1 - Location"
      @ 9, 16 SAY "2 - Location Section"
      @ 10, 16 SAY "3 - Layer ID"
      @ 11, 16 SAY "4 - Geometric & Shoulder"
      @ 12, 16 SAY "5 - Surface"
      @ 13, 16 SAY "6 - Subgrade"
    ENDDO
@ 14, 16 SAY "7 - Layer Thickness Across the Road"
@ 15, 16 SAY "8 - Double Surface Treatment"
@ 17, 37 SAY "OPTION .F."
@ 17, 51 GET REPTCOLL
@ 2, 9 TO 19, 65 DOUBLE
READ
IF READKEY() = 12
   MREPEAT = .F.
   EXIT
ENDIF
ENDDO
IF MREPEAT
SET ESCAPE ON
DO CASE
CASE REPTCOLL = "1"
   CLEAR
   USE \PAVEDB\FILES\LOCATION INDEX \PAVEDB\INDEXES\LOCSID
   REPORT FORM \PAVEDB\REPORTS\SEC_LOCN TO PRINT
CASE REPTCOLL = "2"
   CLEAR
   SET PRINT TO LMPRT1
   @ 10, 5 SAY "Please set the printer to Condensed print"
   WAIT
   CLEAR
   USE \PAVEDB\FILES\LOCATION INDEX \PAVEDB\INDEXES\LOCSID
   REPORT FORM \PAVEDB\REPORTS\LOCATION TO PRINT
CASE REPTCOLL = "3"
   CLEAR
   USE \PAVEDB\FILES\LAYER INDEX \PAVEDB\INDEXES\LAYNDX
   REPORT FORM \PAVEDB\REPORTS\LAYER TO PRINT
CASE REPTCOLL = "4"
   CLEAR
   USE \PAVEDB\FILES\GEOGEO INDEX \PAVEDB\INDEXES\GEOGNDX
   REPORT FORM \PAVEDB\REPORTS\GEOGEO TO PRINT
CASE REPTCOLL = "5"
   CLEAR
   USE \PAVEDB\FILES\SURFACE INDEX \PAVEDB\INDEXES\SURFNDX
   REPORT FORM \PAVEDB\REPORTS\SURFACE TO PRINT
CASE REPTCOLL = "6"
   CLEAR
   USE \PAVEDB\FILES\SUBGRADE INDEX \PAVEDB\INDEXES\SUBGNDX
   REPORT FORM \PAVEDB\REPORTS\SUBGRADE TO PRINT
CASE REPTCOLL = "7"
   CLEAR
   USE \PAVEDB\FILES\LAYTHICK INDEX \PAVEDB\INDEXES\LAYNDX
   REPORT FORM \PAVEDB\REPORTS\LAYTHICK TO PRINT
CASE REPTCOLL = "8"
   CLEAR
   USE \PAVEDB\FILES\LAYER INDEX \PAVEDB\INDEXES\LAYNDX
   REPORT FORM \PAVEDB\REPORTS\DISTREPT FOR LAYMATCL = 6 TO PRINT
ENDCASE
SET ESCAPE OFF
CLEA TYPE
CLEAR
ENDIF
ENDDO

* if choice 4 is chosen, print one of the 5 reports for the monitoring data
CASE REPTCOLL = "4"
MREPEAT = .T.
DO WHILE MREPEAT
CLEAR
SET PRINT TO &IMPORT2
REPTCOLL = ""
DO WHILE .NOT. (REPTCOLL $ '12345')
@ 4, 22 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE 2.4"
@ 5, 34 SAY "REPORTS"
@ 6, 30 SAY "Monitoring Data"
@ 9, 16 SAY "1 - Visual"
@ 10, 16 SAY "2 - Serviceability Index"
@ 11, 16 SAY "3 - Falling Weight"
@ 12, 16 SAY "4 - Dynaflect"
@ 13, 16 SAY "5 - Skid"
@ 16, 38 SAY "OPTION ———>"
@ 16, 52 GET REPTCOLL
@ 3, 9 TO 18, 65 DOUBLE
READ
IF READKEY() = 12
    MREPEAT = .F.
EXIT
ENDIF
ENDDO
IF MREPEAT
SET ESCAPE ON
DO CASE
CASE REPTCOLL = "1"
    CLEAR
    SET PRINT TO &IMPORT1
    @ 10, 5 SAY "Please set the printer to Condensed print to print this report"
    WAIT
    CLEAR
    USE \PAVEDB\FILES\VISUAL INDEX \PAVEDB\INDEXES\VISUAL
    REPORT FORM \PAVEDB\REPORTS\VISUAL TO FILE VISUAL1.RPT
CASE REPTCOLL = "2"
    CLEAR
    USE \PAVEDB\FILES\SI INDEX \PAVEDB\INDEXES\SI
    REPORT FORM \PAVEDB\REPORTS\SI TO PRINT
CASE REPTCOLL = "3"
    CLEAR
    SET PRINT TO &IMPORT1
    @ 10, 5 SAY "Please set the printer to Condensed print to print this report"
    @ 11, 5 SAY "Note that this report is in 2 parts and that the first"
    @ 12, 5 SAY "part will be printed completely before the next part is printed."
    WAIT
CLEAR
USE \PAVEDB\FILES\FALLWGT INDEX \PAVEDB\INDEXES\FALLWGT
REPORT FORM \PAVEDB\REPORTS\FALLWGT TO PRINT
GOTO TOP
REPORT FORM \PAVEDB\REPORTS\FALLWGT2 TO PRINT
CASE REPTCOLL = "4"
CLEAR
USE \PAVEDB\FILES\DYNAPLM INDEX \PAVEDB\INDEXES\DYNAPLM
REPORT FORM \PAVEDB\REPORTS\DYNAPLM TO PRINT
CASE REPTCOLL = "5"
CLEAR
USE \PAVEDB\FILES\SKID INDEX \PAVEDB\INDEXES\SKID
REPORT FORM \PAVEDB\REPORTS\SKID TO PRINT
ENDCASE
SET ESCAPE OFF
CLEAR TYPE
CLEAR
ENDIF
ENDDO

* print traffic data if 5 is chosen
CASE REPTPICK = "5"
CLEAR
MRESPONSE = ""
@ 10,10 SAY "You are about to print the Traffic report."
@ 11,11 SAY "Want to continue (Y/N)? " GET MRESPONSE
READ IF MRESPONSE = "Y"
SET ESCAPE ON
SET PRINT TO &IMPORT2
USE \PAVEDB\FILES\TRAFFIC INDEX \PAVEDB\INDEXES\TRAFFIC
REPORT FORM \PAVEDB\REPORTS\TRAFFIC TO PRINT
SET ESCAPE OFF
ENDIF

* print out weather or environment data if choice 6 is chosen
CASE REPTPICK = "6"
MREPEAT = .T.
DO WHILE MREPEAT
CLEAR
SET PRINT TO &IMPORT2
REPTCOLL = ""
DO WHILE .NOT. (REPTCOLL $ '12')
@ 5, 22 SAY "Texas Flexible Pavement Database" 2.6"
@ 6, 34 SAY "REPORTS"
@ 7, 29 SAY "Environmental Data"
@ 10, 15 SAY "1 - Environment"
@ 11, 15 SAY "2 - Weather"
@ 13, 40 SAY "OPTION ————>
@ 13, 55 GET REPTCOLL
@ 4, 9 TO 15, 65 DOUBLE
READ IF READKEY() = 12
MREPEAT = .F.
EXIT
ENDIF
ENDDO
IF MREPEAT
SET ESCAPE ON
DO CASE
CASE REPTCOLL = "1"
CLEAR
USE \PAVEDB\FILES\ENV INDEX \PAVEDB\INDEXES\ENV
REPORT FORM \PAVEDB\REPORTS\ENV TO PRINT
CASE REPTCOLL = "2"
CLEAR
SET PRINT TO IMPORT1
@ 10, 5 SAY "Please set the printer to Condensed print to print this report"
WAIT
CLEAR
USE \PAVEDB\FILES\WEATHER INDEX \PAVEDB\INDEXES\WEATHER
REPORT FORM \PAVEDB\REPORTS\WEATHER TO PRINT
ENDCASE
SET ESCAPE OFF
CLEAR
ENDIF
ENDDO

* print out one of the tables if choice 7 is picked
CASE REPTPICK = "7"
MREPEAT = .T.
DO WHILE MREPEAT
CLEAR
REPTCOLL = ""
DO WHILE .NOT. (REPTCOLL $ '1234567')
@ 3, 21 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE 2.7"
@ 4, 34 SAY "REPORTS"
@ 5, 34 SAY "Tables"
@ 8, 16 SAY "1 - County Name"
@ 9, 16 SAY "2 - Material Type"
@ 10, 16 SAY "3 - Type of Pavemen"t
@ 11, 16 SAY "4 - District Temperature Constant"
@ 12, 16 SAY "5 - Widening Flag"
@ 13, 16 SAY "6 - Layer Description"
@ 14, 16 SAY "7 - Functional Classification"
@ 17, 42 SAY "OPTION ————"
@ 17, 56 GET REPTCOLL
@ 2, 9 TO 19, 65 DOUBLE
READ
IF READKEY() = 12
MREPEAT = .F.
EXIT
ENDIF
ENDDO
IF MREPEAT
SET ESCAPE ON
DO CASE
CASE REPTCOLL = "1"
  CLEAR
  USE \PAVEDB\FILES\CNTYTB
  REPORT FORM \PAVEDB\REPORTS\CNTYTB TO PRINT
CASE REPTCOLL = "2"
  CLEAR
  USE \PAVEDB\FILES\MATTB
  REPORT FORM \PAVEDB\REPORTS\MATTB TO PRINT
CASE REPTCOLL = "3"
  CLEAR
  USE \PAVEDB\FILES\PAVETYPE
  REPORT FORM \PAVEDB\REPORTS\PAVETYPE TO PRINT
CASE REPTCOLL = "4"
  CLEAR
  USE \PAVEDB\FILES\DISTTEMP
  REPORT FORM \PAVEDB\REPORTS\DISTTEMP TO PRINT
CASE REPTCOLL = "5"
  CLEAR
  USE \PAVEDB\FILES\WIDENFLG
  REPORT FORM \PAVEDB\REPORTS\WIDENFLG TO PRINT
CASE REPTCOLL = "6"
  CLEAR
  USE \PAVEDB\FILES\LAYERTB
  REPORT FORM \PAVEDB\REPORTS\LAYERTB TO PRINT
CASE REPTCOLL = "7"
  CLEAR
  USE \PAVEDB\FILES\FUNCLTBl
  REPORT FORM \PAVEDB\REPORTS\FUNCLTBl TO PRINT
ENDCASE
SET ESCAPE OFF
CLEAR
ENDIF
ENDDO
ENDCASE
MREPEAT2 = .F.
ENDIF
MREPEAT2 = .T.
ENDDO
RETURN
PROGRAM LISTING

* SUBSYSTEM: PRINT SUMMARY REPORT
* PROGRAM NAME: SUMMSID.PRG 01/21/88
* REVISED ON: 05/26/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO PRINT OUT A SUMMARY REPORT FOR INDIVIDUAL SID NUMBERS IN
* THE LOCATION FILE. THE FOLLOWING PROGRAMS ARE CALLED:
* - GETPAVCN.PRG
* - GETENVIR.PRG
* - GETLOCAT.PRG
* - PRNTREPT.PRG
* - GETSKID.PRG
* 
* THE FOLLOWING FILES ARE USED IN THIS PROGRAM:
* CNTYENV.DBF CNTYENV.NDX
* CNTYTBEL.DBF
* CNTYWEAT.DBF WEATMNTH.NDX
* WEATCNTY.NDX
* CNTYTBLNO.NDX
* DISTTEMP.DBF DYNALLNO.NDX
* DYNALLID.DBF FALLWTG.NDX
* FALLWTG.DBF GEOSHO.NDX
* GEOSHO.DBF LAYNDX.NDX
* LAYER.DBF LAYNDX.NDX
* LAYERTBL.DBF
* LAYTHICKTBL.DBF LAYNDX.NDX
* LOCATION.DBF LOGSID.NDX
* MATTBL.DBF
* PAVETRYPE.DBF SI.NDX
* SI.DBF SIMSKID.NDX
* SIMSKID.DBF SUBGRADE.NDX
* SUBGRADE.DBF SURFN.DX
* SURFACE.DBF TRAFFIC.NDX
* TRAFFIC.DBF VISUAL.NDX
* VISUAL.DBF
* 
* the program first gets all the data and stores it in separate files
* using memory variables. When the report is printed out, the appropriate
* files with the variables are opened and deleted when done with them.

* set all parameters
SET TALK OFF
SET ECHO OFF
CLOSE DATABASES
CLEAR
SET PRINTER TO &PRINT2
PUBLIC MSID_NO, VALID, MCOUNTY, MDISTRICT

128
STORE " " TO MQUIT
STORE .F. TO VALID
STORE 0 TO MSID_NO

* get the sid number
DO WHILE .NOT. VALID
  CLEAR
  @ 10, 5 SAY " "
  ACCEPT "Please enter Section Identification (SID) number: " TO SID_NO
  IF README() = 12
    CLEAR
    @ 19, 5 SAY "ARE YOU SURE YOU WANT TO RETURN TO PREVIOUS MENU? (Y/N)"
    GET MQUIT
  READ
    IF MQUIT = "Y" .OR. MQUIT = "y"
    EXIT
  ENDF I N
ENDIF
MSID_NO = VAL(SID_NO)

* calculates the correct Sid Number
STORE 0 TO VAR1, VAR2, VAR3, COMPARE
VAR1 = INT(MSID_NO/1000)
VAR2 = MOD(INT(MSID_NO/100),10)
VAR2 = VAR2 * 2
VAR3 = MOD(INT(MSID_NO/10),10)
VAR3 = VAR3 * 3
VAR4 = MOD(VAR1,10)+VAR2+VAR3
COMPARE = MOD(VAR4,10)

* compares calculated Sid number with Sid number entered
IF COMPARE = MOD(MSID_NO,10)
  STORE .T. TO VALID
ELSE
  @ 19, 10 SAY "Invalid SID Number"
  WAIT
  @ 18, 10 CLEAR
ENDIF
* verifies that Sid number has been entered
IF MSID_NO = 0
  STORE .F. TO VALID
  @ 19, 10 SAY "Invalid SID Number"
  WAIT
  @ 18, 10 CLEAR
ENDIF
ENDDO

* get all the data and print the report
IF VALID
  CLEAR
  DO \PAVEDB\REPORTS\GETLOCAT
  DO \PAVEDB\REPORTS\GETPAVON
  DO \PAVEDB\REPORTS\GETENVIR
  DO \PAVEDB\REPORTS\GETSKID

129
DO \PAVED\REPORTS\PRINTREPT
ENDIF
RELEASE MSID_NO, VALID, MCounty, MDistrict
RETURN
**PROGRAM LISTING**

* * SUBSYSTEM: PRINT SUMMARY REPORT
* * PROGRAM NAME: SUMM DIST.PRG 02/04/88
* * REVISED ON: 05/26/88
* * PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* * TAMU/TIT
* * AUTHOR: TREVOR X. PEREIRA
* * PURPOSE: TO PRINT OUT A SUMMARY REPORT FOR ALL SID NUMBERS IN A PARTICULAR DISTRICT. THE FOLLOWING PROGRAMS ARE CALLED:
  * - GETPAVCN.PRG
  * - GETENVIR.PRG
  * - GETLOCAT.PRG
  * - PRNI REPT.PRG
  * - GETSKID.PRG
* * THE FOLLOWING FILES ARE USED IN THIS PROGRAM:
  * CNTYENV.DBF CNTYENV.NDX
  * CNTYTB.L.DBF
  * CNTYWEAT.DBF WEATMINH.NDX
  * WEATCNTY.NDX
  * CNTYTDNO.NDX
  * DISTTEMP.DBF
  * DYNAILL.DBF DYNAILL.DNDX
  * FALLWGT.DBF FALLWGT.NDX
  * GEOGO.DBF GEOGO.NDX
  * LAYER.DBF LAYNDX.NDX
  * LAYERB.L.DBF
  * LAYTHICK.DBF LAYNDX.NDX
  * LOCATION.DBF LOCISD.NDX
  * MATTB.L.DBF
  * PAVETYPE.DBF
  * SH.DBF SH.NDX
  * SIMPSKID.DBF SIMPSKID.NDX
  * SUBGRADE.DBF SUBGRADE.NDX
  * SURFACE.DBF SURFDX.NDX
  * TRAFFIC.DBF TRAFFIC.NDX
  * VISUAL.DBF VISUAL.NDX
  * SIDSTORE.DBF - This file is used to store the SID numbers for a particular district.
* * the program first gets all the SID numbers for the particular district and stores them in a temporary file. Then data for each SID number is stored in memory variable files. When the report is printed out, the appropriate files with the variables are opened and deleted when done with them. This process continues for each SID number in the temporary file.

* set all parameters
CLOSE DATABASES
CLEAR
SET PRINTER TO &MPORT2
PUBLIC MSID_NO, VALID, MOUNTY, MDISTRICT
STORE " " TO MQUIT, MDIST
STORE .F. TO VALID
STORE 0 TO MSID_NO, MDISTT

* get the sid number
DO WHILE .NOT. VALID
  CLEAR
  CLEAR TYPEAHEAD
  @10, 5 SAY ""
  ACCEPT "Please enter District number: " TO MDISTT
  IF READKEY() = 12
    CLEAR
    @19, 5 SAY "ARE YOU SURE YOU WANT TO RETURN TO PREVIOUS MENU? (Y/N)"
  GET MQUIT
  READ
  IF MQUIT = "Y" .OR. MQUIT = "y"
    EXIT
  ENDIF
ENDIF
MDIST = VAL(MDISTT)
@10, 0 CLEAR

* Get all SID_NO's for the required district and store them to a temporary dbase file
SELECT 1
  USE \PAVEDB\FILES\LOCATION index \PAVEDB\INDEXES\locsid
SELECT 2
  IF .NOT. FILE('\PAVEDB\FILES\SIDSTORE.DBF')
    CLEAR
    ?"Temporary SID storage file (SIDSTORE.DBF) not found. Please check..."
    WAIT
    RETURN TO MASTER
  ENDIF
  USE \PAVEDB\FILES\SIDSTORE
  DELE ALL
PACK

SELECT 1
LOCATE FOR HWYDIST = MDIST
IF .NOT. FOUND()
  @12, 5 SAY "District not found. Please try again."
  WAIT
ENDIF
DO WHILE FOUND()
  MSID_NO = A->SID_NO
  SELE 2
  APPEND BLANK
  REPLACE B->SID_NO WITH MSID_NO
  SELE 1
  VALID = .T.
  CONTINUE
ENDD

ENDDO

SELE 1

USE

SELE 2

APPEND BLANK

REPLACE B->SID_NO WITH 0

USE

* Print out the reports

USE \PAVEDB\FILES\SIDSTORE

MREC = 1

IF SID_NO = 0

STORE .F. TO GOON

ELSE

STORE .T. TO GOON

ENDIF

DO WHILE GOON

MSID_NO = SID_NO

DO \PAVEDB\REPORTS\GETLOCAT

DO \PAVEDB\REPORTS\GETPAVCN

DO \PAVEDB\REPORTS\GETENVIR

DO \PAVEDB\REPORTS\GETSKID

DO \PAVEDB\REPORTS\PRNTREPT

MREC = MREC + 1

USE \PAVEDB\FILES\SIDSTORE

GOTO MREC

IF SID_NO = 0

GOON = .F.

ENDIF

CLEAR

ENDDO

SELE 3

USE

SELE 2

USE

SELE 1

USE

CLOSE DATABASES

SET PRINTER TO LPT1

RELE MSID_NO, VALID, MCOUNTY, MDISTRICT

RETURN
PROGRAM LISTING

* SUBSYSTEM: PRINT SUMMARY REPORT
* PROGRAM NAME: SUMMALL.PRG 01/21/88
* REVISED ON: 05/26/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR: TREvor X. PEREIRA
* PURPOSE: TO PRINT OUT A SUMMARY REPORT FOR ALL THE SID NUMBERS
IN THE LOCATION FILE. THE FOLLOWING PROGRAMS
* ARE CALLED: - GETPAVCN.PRG
* - GETENVIR.PRG
* - GETLOCAT.PRG
* - PRNTREPRT.PRG
* - GETSKID.PRG

* THE FOLLOWING FILES ARE USED IN THIS PROGRAM:
* CNTYENV.DBF CNTYENV.NDX
* CNTYTL.EX DBF CNTYTL.NDX
* CNTYWEAT.DBF WEATMNTH.NDX
* WEATCNTY.NDX
* CNTYTLNO.NDX
* DISTSMP.DBF
* DYNSTDN.DBF DYNSTDN.NDX
* FALLIGHT.DBF FALLIGHT.NDX
* GEOHO.DBF GEOHO.NDX
* LAYER.DBF LAYNDX.NDX
* LAYERBL.DBF
* LAYTHICK.DBF LAYNDX.NDX
* LOCATION.DBF LOCSDT.NDX
* MTLTB.L.DBF
* Pavetype.dbf
* SI.DBF SI.NDX
* SIMPSKID.DBF SIMPSKID.NDX
* SUBGRADE.DBF SUBGRADE.NDX
* SURFACE.DBF SURFNDX.NDX
* TRAFFIC.DBF TRAFFIC.NDX
* VISUAL.DBF VISUAL.NDX

* the program first gets all the data and stores it in separate files
* using memory variables. When the report is printed out, the appropriate
* files with the variables are opened and closed when done with them.

* set all parameters
SET TALK OFF
SET ECHO OFF
CLOSE DATABASES
CLEAR
SET PRINTER TO &MPRT2
PUBLIC MSID NO, MOUNITY, MDISTRICT
STORE " " TO MQUIT

134
STORE 0 TO MSID_NO
STORE 1 TO MREC
USE \PAVEDB\FILES\LOCATION
MRECNUM = RECCOUNT() + 1

DO WHILE MREC < MRECNUM
    USE \PAVEDB\FILES\LOCATION
    GOTO MREC
    MSID_NO = SID_NO
    USE
    IF READKEY() = 12
        CLEAR
        0 19, 5 SAY "ARE YOU SURE YOU WANT TO QUIT? (Y/N) "
        0 19, 43 GET MQUIT
        READ
        IF MQUIT = "Y" .OR. MQUIT = "y"
            EXIT
        ENDIF
    ENDIF
ENDIF
CLEAR

* get all the data and print the report
DO \PAVEDB\REPORTS\GETLOCAT
DO \PAVEDB\REPORTS\GETPAVCN
DO \PAVEDB\REPORTS\GETENVIR
DO \PAVEDB\REPORTS\GETSKID
DO \PAVEDB\REPORTS\PRNTREPT
MREC = MREC + 1
ENDDO

RELEASE MSID_NO, MCOUNTY, MDISTRICT
RETURN
* SUBSYSTEM: PRINT SUMMARY REPORT
* PROGRAM NAME: GETLOCAT.PRG   01/26/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TITI
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO GET DATA FROM THE FOLLOWING FILES TO PRINT THE SUMMARY REPORT:
* 1) LOCATION dBASE FILE
* 2) SERVICEABILITY dBASE FILE
* 3) TRAFFIC dBASE FILE
* 4) LAYER IDENTIFICATION dBASE FILE
* 5) SUBGRADE LAYER dBASE FILE
* 6) SURFACE LAYER dBASE FILE
* 7) GEOMETRIC & SHOULDER INFO LAYER dBASE FILE

**********************************************************************************
* GET LOCATION PART OF THE REPORT AND STORE THE DATA IN A FILE
**********************************************************************************

@ 5, 5 SAY "Getting data for the Location section of the report ...."

* assign databases to different work areas
SELECT 1
use PAVEDB\FILES\location INDEX PAVEDB\INDEXES\LOCSID

SELECT 2
USE PAVEDB\FILES\CNTYTBL INDEX PAVEDB\INDEXES\CTYTBNO

* locate for the requested sid number
SELECT 1
seek msid no
if .not. found()
   okay = .F.
   clear
   @ 19, 10 say "SID number not found. "
   WAIT
   @ 18, 10 CLEAR
   RETURN
else
   okay = .T.
endif
STORE 0 TO MCOUNTY
STORE 0 TO MDISTRICT

* if the sid number is found, get all the required data
if Okay
   mlhwydist = LTRIM(STR(hwydist,2))
   NDISTRICT = HWYDIST
   MCOUNTY = CNTYNUM
   mlcntytm = LTRIM(STR(cntytm,3))
   SELECT 2
SEEK A->CNTYNUM
IF FOUND()
   MLCNTYNUM = LITRIM(CNTYNAME)
ENDIF
SELECT 1
mcontrl = LITRIM(STR(control,4))
msectn = LITRIM(STR(section,2))
MLCONSEC = MCONTRL + "-" + MSECIN
mprefix = hwyprefix
mhwnum = LITRIM(STR(hwynum,4))
msuffix = hwysuffix
MLHWY = MPREFIX + " " + LITRIM(MHWNUM) + " " + MSUFFIX
MBEGP = LITRIM(STR(BEGMPST,3))
MBEGD = RIGHT((STUFF(STR(BMPSIDTIS,2),1,(2-LEN(LITRIM(STR(BMPSIDTIS,2))))),"0")),2)
MPOSTB = MBEGP+BOSISSIGN+MBEGD
MEND =
RIGHT((STUFF(STR(BMPSIDTIS,2),1,(2-LEN(LITRIM(STR(BMPSIDTIS,2))))),"0")),2)
MENDP = LITRIM(STR(ENDMPST,3))
MPOSTE = MENDP+EOSISSIGN+MENDD
MLPOST = MPOSTB+" TO "+MPOSTE
mlaneid = laneid
IF PREVISD = 0
   MLPREVSD = "+"
ELSE
   mlprevsd = LITRIM(STR(prevsid,4))
ENDIF
IF NEXTSID = 0
   MLNEXTSD = "+"
ELSE
   mlnextsd = LITRIM(STR(nextsid,4))
ENDIF
MLFUNCLS = LITRIM(STR(FUNCLS,2))
endif

* get geometric and shoulder information layer data
SELECT 1
use \PAVEDB\FILES\geosho index \PAVEDB\INDEXES\geondx

SELECT 2
USE \PAVEDB\FILES\PAVETYPE

SELECT 1
seek str(msid_no,4)
if found()
   DO CASE
      CASE WIDENFLG = 0
         MLWIDEN = "NO WIDENING"
      CASE WIDENFLG = 1
         MLWIDEN = "WIDENING"
      CASE WIDENFLG = 2
         MLWIDEN = "UNUSUAL WIDENING"
   END
ENDCASE
MPAVEM = PAVETYP
SELECT 2
LOCATE FOR PAVECODE = MPAVEM
IF FOUND()
    MLBASETYPE = LITRIM(BASETYPE)
    MLBASETHK = LITRIM(BASETHK)
    MLBASESEAL = LITRIM(BASESEAL)
ELSE
    MLBASETYPE = " "
    MLBASETHK = " "
    MLBASESEAL = " "
ENDIF
ELSE
    MLWIDEN = " "
    MLBASETYPE = " "
    MLBASETHK = " "
    MLBASESEAL = " "
ENDIF

* save all location variables to a file
SAVE ALL LIKE ML* TO GETLOCAT
RELEASE ALL LIKE ML*
sel 1
use
sel 2
use

*****************************
* GET SERVICEABILITY FILE DATA AND STORE TO A FILE
*****************************

@ 6, 5 SAY "Getting data for Serviceability Index Section ...."
X = 1
use \PAVEDB\FILES\SI index \PAVEDB\INDEXES\SI
DO WHILE X < 7
    S = STR(X,1,0)
    msyear&S = " "
    mmean&S = " "
    mssdrd&S = " "
    mscont&S = " "
    msoe&S = " "
    mslow&S = " "
    mshigh&S = " 
    X = X + 1
ENDDO

seek str(msid_no,4)

* if the sid number is found, get the required data and store it in variables
if found()
    * move the file pointer to the last record of the sid_no in order to get
    * the most current year first
    DO WHILE .NOT. EOF() .AND. SID_NO = MSID_NO
SKIP
ENDDO
SKIP -1
X = 1
do while .NOT. EOF() .AND. SID_NO = MSID_NO
   IF X > 6
      EXIT
   ENDIF
   S = STR(X,1,0)
   msysyearS = STR(actyear,2)
   msmeanS = STR(simean,4,2)
   msstddS = STR(sisd,4,3)
   mscontS = STR(sicount,2)
   mscoefS = STR(((sisd/simean)*100),4,1)
   mslowvS = STR(silowval,4,2)
   mshighS = STR(sihival,4,2)
   X = X + 1
   skip -1
   IF EOF()
      SKIP 35
   ENDIF
ENDDO
ENDIF

USE

* save all variables to file
save all like ms* to getservc
RELEASE ALL LIKE MS*

***************
* GET TRAFFIC FILE DATA AND STORE IT IN VARIABLES
***************

@ 7, 5 SAY "Getting data for the Traffic Section ...."
USE \PAVED\FILES\TRAFFIC INDEX \PAVED\INDEXES\TRAFFIC SEEK STR(MSID NO,4)
* if the sid number is found, get the required data
IF FOUND()
   MIBYEAR = STR(YEAR,4)
   MVEHADD = AADT1WAY * 365.25
   MVEHICLE = MVEHADD
   M18KEAL = A18KEAL
   SKIP
   DO WHILE .NOT. EOF() .AND. TRAFFIC->SID NO = MSID NO
      MVEHADD = AADT1WAY * 365.25
      MVEHICLE = MVEHICLE + MVEHADD
      M18KEAL = M18KEAL + A18KEAL
   SKIP
ENDDO
MT18KEAL = STR(M18KEAL,9)
MTVEHIC = STR(MVEHICLE,10)
SKIP -1
MIAADT = STR(AADT1WAY,6)
MIPIRK = STR(PCTIRK,4,1)
MIEYEAR = STR(YEAR,4)
ELSE
STORE " " TO MIBYEAR, MIEYEAR, MIPIRK
STORE " " TO MIAADT
STORE " " TO MT19KEAL
STORE " " TO MIVEHIC
ENDIF
* save the variables to file
SAVE ALL LIKE MT* TO GETTRAFF
RELEASE ALL LIKE MT*
CLOSE ALL
use

************************
* GET STRUCTURAL SECTION OF THE REPORT AND SAVE THE DATA IN VARIABLES
************************

* variable mcount is to find out how many layers there are for the SID #.
* This will help determine the layer the subgrade goes on. This is needed
* if layer # 1 is the top most layer and not the subgrade.

@ 8, 5 SAY "Getting data for the Structural Section ...."
STORE " " TO MISTL, MIILQILM, MIITXIRL
MOUNT = 1

set path to \pavedb\files,\pavedb\indexes
set view to \pavedb\files\strucsec
seek str(msid_no,4)

* if the sid number is found, get the required data
if found()
do while .NOT. EOF().AND. sid_no = msid_no
    skip
endo
do while .NOT. EOF().AND. sid_no = msid_no
    IF X < 10
        S = STR(X,1)
    ELSE
        S = STR(X,2)
    ENDIF
    mtstruc&S = str(structnum,2)
    mtlay&S = STR(laynum,2)
    mlaydes&S = laydesc
    MTDATE&S = str(jobcompmo,2)+"/"+str(jobcompyr,2)
    mlaymat&S = laymatcl
    IF CENTHRK = 0
        MTCENTHRK&S = " 
    ELSE

\texttt{mtoenthkS} = \texttt{str(centthk,5,2)}
\textbf{ENDIF}
\textbf{IF} \texttt{AGAPPLRT} = 0
\textbf{MTAGGRAT\&S} = " "
\textbf{ELSE}
\texttt{mtaggrat\&S} = \texttt{str(agapplrt,3)}
\textbf{ENDIF}
\texttt{mtadm\&xtp\&S} = adm\&xtyp
\textbf{IF} \texttt{ADMXPER} = 0
\texttt{MTADM\&XPR\&S} = " "
\textbf{ELSE}
\texttt{mtadm\&xpr\&S} = \texttt{str(admxper,5,2)}
\textbf{ENDIF}
\textbf{IF} \texttt{ASAPPLRT} = 0
\texttt{MTAPPLRT\&S} = " "
\textbf{ELSE}
\texttt{mtapplrt\&S} = \texttt{str(asapplrt,4,2)}
\textbf{ENDIF}
\texttt{SAVE ALL LIKE MT* TO GETSTC\&S}
\texttt{RELEASE ALL LIKE MT*}
\texttt{X} = \texttt{X + 1}
\texttt{MCOUNT} = \texttt{MCOUNT} + 1
\texttt{skip -1}
\texttt{IF BOF()}
\quad \texttt{SKIP 35}
\textbf{ENDIF}
\textbf{enddo}
\texttt{X} = \texttt{MCOUNT}
\texttt{MCOUNT} = \texttt{MCOUNT} - 1
* initialize all variables
\textbf{DO WHILE} \texttt{X} < 14
\quad \textbf{IF} \texttt{X} < 10
\quad \texttt{S} = \texttt{STR(X,1)}
\quad \textbf{ELSE}
\quad \quad \texttt{S} = \texttt{STR(X,2)}
\quad \textbf{ENDIF}
\texttt{STORE " " TO MT\&STUC\&S, MT\&AYES, MI\&JOEMO\&S, MI\&JOBYR\&S, MTC\&ENTHK\&S, MT\&AGGRAT\&S, MT\&ADM\&XTP\&S, MT\&ADM\&XPR\&S}
\texttt{STORE " " TO MT\&APPLRT\&S, MTA\&DATE\&S, MI\&AYDES\&S, MI\&AYMAT\&S}
\texttt{SAVE ALL LIKE MT* TO GETSTC\&S}
\texttt{RELEASE ALL LIKE MT*}
\quad \texttt{X} = \texttt{X + 1}
\textbf{ENDDO}

* get the material type and the layer names from the tables
\textbf{SELECT} 6
\textbf{USE} \texttt{\PAVEDB\FILES\\MAT\&TBL}
\textbf{SELECT} 7
\textbf{USE} \texttt{\PAVEDB\FILES\LAY\&ERTBL}
\textbf{SELECT} 6
\texttt{MCOUNT} = \texttt{MCOUNT}
\texttt{X} = 1
\textbf{DO WHILE} \texttt{X} < 14
\quad \textbf{IF} \texttt{X} < 10

141
S = STR(X,1)
ELSE
    S = STR(X,2)
ENDIF
SELECT 6
TYPE1 = MLAYMAT&S
TYPE2 = "TYPE1"
TYPE3 = TYPE(TYPE2)
IF TYPE3 = "N"
    LOCATE FOR MATCODE = MLAYMAT&S
    IF FOUND()
        MZMATTYP&S = F->MATDESC
        MZLAYDA&S = F->LAYRDES
    ELSE
        MZMATTYP&S = " "
        MZLAYDA&S = " "
    ENDIF
ELSE
    MZMATTYP&S = " "
    MZLAYDA&S = " "
ENDIF
SELECT 7
TYPE1 = MLAYDES&S
TYPE2 = "TYPE1"
TYPE3 = TYPE(TYPE2)
IF TYPE3 = "N"
    LOCATE FOR CODE = MLAYDES&S
    IF FOUND()
        MZLAYDB&S = G->CODE_DESC
    ELSE
        MZLAYDB&S = " "
    ENDIF
ELSE
    MZLAYDB&S = " "
ENDIF
X = X + 1
ENDDO
ELSE
    X = 1
DO WHILE X < 14
    IF X < 10
        S = STR(X,1)
    ELSE
        S = STR(X,2)
    ENDIF
STORE " " TO MISTRUC&S, MLAY&S, MICEH&K&S, MIAGGRAT&S, MTADMXP&S, MTADMXPR&S
STORE " " TO MTAPPLRT&S, MINTER&S, MZLAYDA&S, MZLAYDB&S, MZMATTYP&S
SAVE ALL LIKE MT* TO GEISTIC&S
RELEASE ALL LIKE MT*
    X = X + 1
ENDDO
STORE 0 TO MZCOUNT
endif
MCOUNT = MCOUNT - 1
close DATABASES
set path to

* get the subgrade file data
use \\PAVEDB\\FILES\\subgrade index \\PAVEDB\\INDEXES\\subgndx
seek str(msid_no,4)
if found()
    MZplast = str(plastix,4,1)
    MZliqlim = str(liqlim,4,1)
    MZtxtrl = str(txtraxl,3,1)
else
    MZplast = " "
    MZliqlim = " "
    MZtxtrl = " "
endif

close all

* save all the variables to file
save all like MZ* to getstruc
RELEASE ALL LIKE MZ*
return
PROGRAM LISTING

* * SUBSYSTEM: PRINT SUMMARY REPORT
* PROGRAM NAME: GETENVIR.FRG 01/26/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TII
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO GET DATA FOR THE ENVIRONMENT SECTION OF THE REPORT FROM
* THE FOLLOWING FILES: 1) WEATHER dBASE FILE
* 2) ENVIRONMENT dBASE FILE
*

@ 10, 5 SAY "Getting Environment Summary data ...."

* get data from the environment file
USE \PAVEDB\FILES\ENV INDEX \PAVEDB\INDEXES\ENV
SEEK MCOCOUNTY
IF FOUND()
   METHORN = STR(THORNMN,4,1)
ELSE
   METHORN = "   "
ENDIF

* get data from the weather file
USE \PAVEDB\FILES\WEATHER INDEX \PAVEDB\INDEXES\WEATHER
X = 1
* initialize variables
DO WHILE X < 13
   IF X < 10
      S = STR(X,1,0)
   ELSE
      S = STR(X,2,0)
   ENDIF
   MEMEAN&S = "   "
   MEPREC&S = "   "
   MEWFTFC&S = "   "
   METFIC&S = "   "
   X = X + 1
ENDDO
MEMTAVER = "   "
MEPRECI = "   "
MEWFTFC = "   "
METFIC = "   "

* get file data
SEEK str(MCOUNTY,3)
IF FOUND()
   X = 1
   STORE 0 TO MMTAVER, MEPREC, MEWFTFC, METFIC
   DO WHILE .NOT. EOF() .AND. A->CNTNUM = MCOUNTY
      IF X > 12

144
EXIT
ENDIF
IF X < 10
   S = STR(X,1,0)
ELSE
   S = STR(X,2,0)
ENDIF
MEMAN6&S = STR(ATIPEMN,3,0)
MEPRECS = STR(PRECMN,3,1)
MEWFTC&S = STR(WFTCMN,3,0)
MEITFC&S = STR(TFTCMN,3,0)
MMIAVER = MMIAVER + ATIPEMN
MPRECIP = MPRECIP + PRECMN
MWETITFC = MWETITFC + WFTCMN
MIOTITFC = MIOTITFC + TFTCMN
X = X + 1
SKIP
ENDDO
MITEMP = MMIAVER/12
MEMIAVER = STR(MITEMP,4,1)
MEPRECI = STR(MPRECI,4,1)
MEWETITFC = STR(MWETITFC,3,0)
MEIOTITFC = STR(MIOTITFC,4,1)
ENDIF

USE \PAVEDB\FILES\DISTITEMP
LOCATE FOR DISTITEMP->DISTRICT = MDISTRICT
IF FOUND()
   TEMPPCN = STR(TEMPPONS,2)
ELSE
   TEMPPCN = " "
ENDIF

* save all variables to file
SAVE ALL LIKE ME* TO GETENVIR
CLOSE ALL

RETURN
* SUBSYSTEM:         PRINT SUMMARY REPORT
* PROGRAM NAME:     GETPAVCON.PRG     01/26/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR:           TREVOR X. PEREIRA
* PURPOSE:          TO GET DATA FROM THE VISUAL dBASE FILE TO PRINT OUT THE
*                    SUMMARY REPORT.
*
X = 1
* initialize all variables
DO WHILE X < 7
  IF X < 10
    S = STR(X,1,0)
  ELSE
    S = STR(X,2,0)
  ENDDIF
  mpactyr&$ = " "
  mpprs&$ = " "
  MPRUTIN&$ = " "
  MPRUTTA&$ = " "
  MPBLCN&$ = " "
  MPBLCK&$ = " "
  MPALIGN&$ = " "
  MPALLIG&$ = " "
  MPLONGN&$ = " "
  MPLONGA&$ = " "
  MPTRANN&$ = " "
  MPTRANA&$ = " "
  mpcrack&$ = " "
  MPPATCN&$ = " "
  MPPATCA&$ = " "
  mpfailml&$ = " "
  X = X + 1
ENDDO

? 9, 5 SAY "Getting Pavement Condition Survey data ...."
use \PAVEDB\FILES\visual index \PAVEDB\INDEXES\visual
seek str(msid_no,4)
if found()
  * go to the last record of the sid number in order to get the most current
  * sid number first
  DO WHILE .NOT. EOF() .AND. MSID_NO = SID_NO
    SKIP
  ENDDO
  SKIP -1
X = 1
* get the required data and store it in variables
do while .NOT. EOF() .AND. msid_no = sid_no
  IF X < 10
    S = STR(X,1,0)
  ELSE
S = STR(X,2,0)
ENDIF
mpactyr&S = str(actyear,2)
mpprs&S = str(prs,3)
DO CASE
CASE RUTTIV<>0
    MPRUTIN&S = STR(RUTTIV,1)
    MPRUTTA&S = "SE"
CASE RUTTSL<>0
    MPRUTIN&S = STR(RUTTSL,1)
    MPRUTTA&S = "SL"
CASE RUTIMD<>0
    MPRUTIN&S = STR(RUTIMD,1)
    MPRUTTA&S = "MO"
OTHERWISE
    MPRUTIN&S = " "
    MPRUTTA&S = " "
ENDCASE
DO CASE
CASE BLKCRKMD<>0
    mpblkcm&S = STR(blkcrkMD,1)
    MPBLCKA&S = "MD"
CASE BLKCRKSL<>0
    mpblkcm&S = STR(blkcrkSL,1)
    MPBLCKA&S = "SL"
CASE BLKCRKSV<>0
    mpblkcm&S = STR(blkcrksv,1)
    MPBLCKA&S = "SE"
OTHERWISE
    MPBLCKN&S = " "
    MPBLCKA&S = " "
ENDCASE
DO CASE
CASE ALLGCRMD<>0
    mpALGN&S = STR(ALLGCRMD,1)
    MPALGSA&S = "MO"
CASE ALLGCRSL<>0
    mpALGN&S = STR(ALLGCRSL,1)
    MPALGSA&S = "SL"
CASE ALLGCRSV<>0
    mpALGN&S = STR(ALLGCRsv,1)
    MPALGSA&S = "SE"
OTHERWISE
    MPALGN&S = " "
    MPALGSA&S = " "
ENDCASE
DO CASE
CASE LONGCRMD<>0
    mpLONG&S = STR(LONGCRMD,1)
    MPLONGA&S = "MO"
CASE LONGCRSL<>0
    mpLONG&S = STR(LONGCRSL,1)
    MPLONGA&S = "SL"
CASE LONGCRSV<>0

mpLONGN$ = STR(LONGCRsv,1)
MPLONGA$ = "SE"

OTHERWISE
  MPLONGN$ = " "
  MPlONGA$ = " \\
ENDCASE

DO CASE
  CASE TRANCRMD<>0
    mpTRANN$ = STR(TRANCRMD,1)
    MPtrANA$ = "MD"
  CASE TRANCRSL<>0
    mpTRANN$ = STR(TRANCRSL,1)
    MPtrANA$ = "SL"
  CASE TRANCRSV<>0
    mpTRANN$ = STR(TRANCRSV,1)
    MPtrANA$ = "SE"
  OTHERWISE
    MPTRANN$ = " "
    MPtrANA$ = " \\
ENDCASE

DO CASE
  CASE SEALCRCD = 0
    mpCrackS$ = " "
  CASE SEALCRCD = 1
    mpCrackS$ = " S"
  CASE SEALCRCD = 2
    MpcrackS$ = "PS"
  CASE SEALCRCD = 3
    MpcrackS$ = "NS"
ENDCASE

DO CASE
  CASE PATCHGD<>0
    mpPATCHN$ = STR(PATCHGD,1)
    MPPATCA$ = "G"
  CASE PATCHFR<>0
    mpPATCHN$ = STR(PATCHFR,1)
    MPPATCA$ = "F"
  CASE PATCHFR<>0
    mpPATCHN$ = STR(PATCHFR,1)
    MPPATCA$ = "F"
  OTHERWISE
    MPPATCN$ = " \\
    MPPATCA$ = " \\
ENDCASE

mpfailml$ = str(failmile,1)
X = X + 1
skip -1
IF EOF()
  SKIP 35
ENDIF
enddo
endif

use
* save the variables to file
save all like mp* to getpavcn
return
*
* SUBSYSTEM: PRINT SUMMARY REPORT
* PROGRAM NAME: GETSKID.PRG 01/26/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO GET DATA FROM THE FOLLOWING FILES:
*        1) SKID dBASE FILE
*        2) DYNAPLCT dBASE FILE
*        3) FALLING WEIGHT dBASE FILE
*
***************
* GET SKID DATA
***************

@ 11, 5 SAY "Getting the Skid data ...."
X = 1
DO WHILE X < 7
   S = STR(X,1)
   MKDATE&S = " "
   MKMEAN&S = " "
   MKHIGH&S = " "
   MKLOW&S = " "
   X = X + 1
ENDDO

USE \PAVEDB\FILES\SKID INDEX \PAVEDB\INDEXES\SKID

SEEK str(MSID_NO,4)
IF FOUND()
   DO WHILE .NOT. EOF() .AND. SID_NO = MSID_NO
      SKIP
   ENDDO
   SKIP -1
   X = 1
   DO WHILE .NOT. EOF() .AND. SID_NO = MSID_NO
      IF X > 6
         EXIT
      ENDF
      S = STR(X,1)
      MYEAR = STR(YEAR,2)
      MMONIH = STR(MONIH,2)
      MKDATE&S = MMONIH +"/" + MYEAR
      MKMEAN&S = STR(SKIDNUMM,2)
      MKHIGH&S = STR(SKIDNUMH,2)
      MKLOW&S = STR(SKIDNUML,2)
      X = X + 1
      SKIP -1
      IF EOF()
         SKIP 18
EXIT
ENDIF
ENDDO
ENDIF
SAVE ALL LIKE MK* TO GETSKID

**************
* GET DYNAFLECT DATA
**************

@ 12, 5 SAY "Getting the Dynaflect data ...."
USE \PAVEDB\FILES\DYNAFLID INDEX \PAVEDB\INDEXES\DYNAFLID
SEEK STR(MSID_NO,4)
IF FOUND()
   MDAY = LITRIM(STR(DAY,2))
   MMONTH = LITRIM(STR(MONTH,2))
   MYEAR = STR(YEAR,2)
   MDDATE = MMONTH + "/" + MDAY + "/" + MYEAR
   MW1 = 0
   MW2 = 0
   MW3 = 0
   MW4 = 0
   MW5 = 0
   DO WHILE .NOT. EOF() .AND. SID_NO = MSID_NO
      MW1 = MW1 + SENS1RD
      MW2 = MW2 + SENS2RD
      MW3 = MW3 + SENS3RD
      MW4 = MW4 + SENS4RD
      MW5 = MW5 + SENS5RD
      SKIP
   ENDDO
   MW1 = MW1/14
   MW2 = MW2/14
   MW3 = MW3/14
   MW4 = MW4/14
   MW5 = MW5/14
   MDD = "D"
   MDW1 = STR(MW1,5,2)
   MDW2 = STR(MW2,5,2)
   MDW3 = STR(MW3,5,2)
   MDW4 = STR(MW4,5,2)
   MDW5 = STR(MW5,5,2)
ELSE
   MDW1 = " "
   MDW2 = " "
   MDW3 = " "
   MDW4 = " "
   MDW5 = " "
   MDD = " "
   MDDATE = " "
ENDIF
SAVE ALL LIKE MD* TO GETDYNA
***************
* GET FALLING WEIGHT DATA
***************
@ 13, 5 SAY "Getting Falling Weight data ...."
USE \PAVEDB\FILES\FALLWIGHT INDEX \PAVEDB\INDEXES\FALLWIGHT
SEEK STR(MSID_NO,4)
IF FOUND()
    MDAY = LTRIM(STR(DAY,2))
    MMONTH = LTRIM(STR(MONTH,2))
    MYEAR = STR(YEAR,2)
    MFDATE = MMONTH + "/" + MDAY + "/" + MYEAR
    MW1 = SSIGP11+SSIGP21+SSIGP31+SSIGP41+SSIGP51
    MW2 = SSIGP12+SSIGP22+SSIGP32+SSIGP42+SSIGP52
    MW3 = SSIGP13+SSIGP23+SSIGP33+SSIGP43+SSIGP53
    MW4 = SSIGP14+SSIGP24+SSIGP34+SSIGP44+SSIGP54
    MW5 = SSIGP15+SSIGP25+SSIGP35+SSIGP45+SSIGP55
    MW6 = SSIGP16+SSIGP26+SSIGP36+SSIGP46+SSIGP56
    MW7 = SSIGP17+SSIGP27+SSIGP37+SSIGP47+SSIGP57
    MW1 = MW1/5
    MW2 = MW2/5
    MW3 = MW3/5
    MW4 = MW4/5
    MW5 = MW5/5
    MW6 = MW6/5
    MW7 = MW7/5
    MFW1 = STR(MW1,5,2)
    MFW2 = STR(MW2,5,2)
    MFW3 = STR(MW3,5,2)
    MFW4 = STR(MW4,5,2)
    MFW5 = STR(MW5,5,2)
    MFW6 = STR(MW6,5,2)
    MFW7 = STR(MW7,5,2)
    MFD = "F"
ELSE
    MFW1 = "   "
    MFW2 = "   "
    MFW3 = "   "
    MFW4 = "   "
    MFW5 = "   "
    MFW6 = "   "
    MFW7 = "   "
    MFD = "   "
ENDIF
SAVE ALL LIKE MF* TO GETFALL
CLOSE DATABASES
RETURN
STORE "N" TO MANSWER
CLEAR
ON ERROR DO \PAVEDB\REPORTS\ERR_PRINT
    SET PRINT TO @IMPORT.
ON ERROR DO \PAVEDB\ERROR
@ 10,10 SAY "Printing report for SID " + STR(MSID_NO,4)
mdate = dtoc(date())
SET DEVICE TO PRNT
@ 1,115 say "Date: " + mdate
@ 2,0  say " "
-----------------------------------------------
@ 3,0  say "LOCATION                     | ENVIRONMENT -
20 YEAR SUMMARY  (1955-1974)  |

************
* PRINT LOCATION SECTION
************
RESTORE FROM GETLOCAT ADDITIVE
@ 4,0  SAY " " | SECTION ID NO:  "+LIRIM(STR(MSID_NO,4))
@ 4,47  SAY " " |
@ 4,131  SAY " " |
@ 5,0  SAY " " | DISTRICT NO:  "+MLHWYDST
RELEASE MLHWYDST
@ 5,47  SAY " " |
@ 5,72  SAY "JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC  ANN  |
@ 6,0  SAY " " | COUNTY NO/NAME:  "+MLCNTYNM:"+"+MLCNTYNAME

************
* PRINT ENVIRONMENT SECTION
************
RESTORE FROM GETENVIR ADDITIVE
@ 6,47  SAY " " | THORNHWAITE INDEX:  " - "  " - "  " - "  " - "  " - "  " - "  " - "  " - "  " - "  " - "
@ 6,122  SAY METHORN
@ 6,131  SAY " " |
@ 7,0  SAY " " | CONTROL-SECTION:  "+MLCONISEC
@ 7,47  SAY " " | MEAN TEMPERATURE:  "+MEAN1+"  "+MEAN2+"  "+MEAN3+"  "+MEAN4+"  "+MEAN5+"  "+MEAN6+"  "+MEAN7
@ 7,100  SAY MEAN8+"  "+MEAN9+"  "+MEAN10+"  "+MEAN11+"  "+MEAN12+"
"+MEMTAVER
@ 7, 131 SAY "" "" HIGHWAY: "MLEHWY
@ 8, 131 SAY "" "" PRECIPITATION: "MLHWY
@ 8, 27 SAY "" "" Precipitation: "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" 

RELEASE ALL LIKE ME*
@ 11, 131 SAY "" "" NEXT SID: "+MLNEXTSD
@ 12, 0 SAY "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" 

***************
* PRINT TRAFFIC SECTION
***************
RESTORE FROM GET_TRAFFIC_ADDITIVE
@ 18, 0 SAY ""
@ 18, 47 SAY "| |
@ 18, 49 SAY "| "+MTEYEAR+" ADT:" 
@ 18, 78 SAY MPAADT 
@ 18, 85 SAY "| YR MEAN STD DEV N CV LOW HIGH | |
@ 19, 0 SAY "|
@ 19, 47 SAY "| "+MTEYEAR
@ 19, 57 SAY "PERCENT TRUCKS:" 
@ 19, 80 SAY MITPRK

*************
* PRINT SERVICEABILITY SECTION
*************
RESTORE FROM GETSERVC ADDITIVE 
@ 19, 85 SAY "| "+MTEYEAR+" "+MSMEAN1+" "+MSSTDD1+" "+MSCONT1+"
"+MSCOEF1+" "+MSLOWV1+" "+MSHIGH1+" | |
@ 20, 0 SAY "|
@ 20, 47 SAY "| |
@ 20, 49 SAY "|
@ 20, 52 SAY MITYEAR+"="+MTEYEAR
@ 20, 63 SAY "VEHICLES: "+MTVEHIC 
@ 20, 85 SAY "| "+MSMEAN2+" "+MSSTDD2+" "+MSCONT2+"
"+MSCOEF2+" "+MSLOWV2+" "+MSHIGH2+" | |
RELEASE ALL LIKE MS????2, MS????1

*************
* PRINT PAVEMENT CONDITION SURVEY SECTION
*************
RESTORE FROM GETPAVCON ADDITIVE 
@ 21, 0 SAY "|
@ 21, 52 SAY MITYEAR+"="+MTEYEAR
@ 21, 63 SAY "18K AXLES: "+MT18KEAL
RELEASE ALL LIKE MT*
@ 21, 85 SAY "| "+MSMEAN3+" "+MSMEAN4+" "+MSSTDD3+" "+MSCONT3+"
"+MSCOEF3+" "+MSLOWV3+" "+MSHIGH3+" | |
@ 22, 0 SAY "|
@ 22, 49 SAY "|
@ 22, 85 SAY "| "+MSMEAN4+" "+MSMEAN5+" "+MSSTDD4+" "+MSCONT4+"
"+MSCOEF4+" "+MSLOWV4+" "+MSHIGH4+" | |
@ 23, 0 SAY "PAVEMENT CONDITION SURVEY"
@ 23, 47 SAY "|
@ 23, 85 SAY "| "+MSMEAN5+" "+MSMEAN6+" "+MSSTDD5+" "+MSCONT5+"
"+MSCOEF5+" "+MSLOWV5+" "+MSHIGH5+" | |
@ 24, 0 SAY "PVMT RATING "+MPACTYR1+" "+MPACTYR2+" "+MPACTYR3+"
"+MPACTYR4+" "+MPACTYR5+" "+MPACTYR6+" | |
@ 24, 85 SAY "| "+MSMEAN6+" "+MSMEAN7+" "+MSSTDD6+" "+MSCONT6+"
"+MSCOEF6+" "+MSLOWV6+" "+MSHIGH6+" | |
@ 25, 0 SAY "| FRS"
@ 25, 15 SAY MPFRS1+"="+MPFRS2+"="+MPFRS3+"="+MPFRS4+"="+MPFRS5+"
"MPFRS6
@ 25, 47 SAY "| |
@ 25, 87 SAY "| |
RELEASE ALL LIKE MS*
@ 26, 0 SAY "| RUTT"
@ 26, 16 SAY MPFUTTN1+MPFUTTA1+"="+MPFUTTN2+MPFUTTA2+"
"+MPRUTN3+MPRUTA3+" "+MPRUTN4+MPRUTA4+" "+MPRUTN5+MPRUTA5+"
@ 26, 41 SAY MPRUTING+MPRUTAG+"
@ 27, 36 SAY MPRUTING5+MPRUTAG5+"
@ 27, 47 SAY "                        "
@ 28, 36 SAY MPRUTAG6+MPRUTAGA6+"

RELEASE ALL LIKE MPRRS?, MPRUTIN?, MPBLCKN?, MPACTYR?
@ 28, 47 SAY " | SKID NUMBER"
@ 28, 76 SAY " | DEFLECTION (MEAN VARIABLES)"
@ 28, 131 SAY " "
@ 29, 36 SAY MPRTAG2+MPRUTAN2+"
@ 29, 52 SAY "DATE AVG LOW HIGH "
@ 29, 131 SAY " "
@ 30, 36 SAY MPRTAG6+MPRUTAN6+"

RELEASE ALL LIKE MPALLG*, MPLONG*, MPRTRAN*

******************************
* PRINT SKID DATA
******************************
RESTORE FROM GETSKID ADDITIVE
@ 30, 51 SAY MDATEN1
@ 30, 60 SAY MKEAN1+" "+MLOW1+" "+MHI1
@ 30, 76 SAY " | DATE D W1 W2 W3 W4 W5 W6 W7 "
@ 31, 52 SAY "CRACKS "+MCRACK+" "+MCRACK2+" "+MCRACK3+"
"+MCRACK4+" "+MCRACK5+" "+MCRACK6
@ 31, 47 SAY " |"
@ 31, 51 SAY MDATEN2
@ 31, 60 SAY MKEAN2+" "+MLOW2+" "+MHI2
@ 31, 76 SAY " |"

******************************
* PUT IN DEFLECTION DATA
******************************
RESTORE FROM GETDYNADDITIVE
@ 31, 79 SAY MDATE
@ 31, 88 SAY MDAT+" "+MDW1+" "+MDW2+" "+MDW3+" "+MDW4+" "+MDW5
RELEASE ALL LIKE MD*
@ 31, 131 SAY " |
@ 32, 0 SAY " | PATCHING "+MPATC1+"+MPATC2+" "+MPATC3+"+MPATC4+"
"+MPATC5+"+MPATC6+"
@ 32, 36 SAY MPATC7+MPATC8+"
@ 32, 51 SAY MDATEN3
@ 32, 60 SAY MKEAN3+" "+MLOW3+" "+MHI3
@ 32, 76 SAY " |
RESTORE FROM GETFALL ADDITIVE
@ 32, 79 SAY MDATE
@ 32, 88  SAY MFD+" "+MFW1+" "+MFW2+" "+MFW3+" "+MFW4+" "+MFW5+" "+MFW6+"
"+MFW7
RELEASE ALL LIKE MF*
@ 32, 131 SAY " |
@ 33, 0 SAY " " | FAIL/MI "+MPFAIIML1+" "+MPFAIIML2+"
"+MPFAIIML3+" "+MPFAIIML4+" "+MPFAIIML5+" "+MPFAIIML6+" | |
RELEASE ALL LIKE MP*
@ 33, 51  SAY MKDATE4
@ 33, 60  SAY MKMEAN4+" "+MKLOW4+" "+MKHIGH4
@ 33, 76  SAY " | |
@ 33, 131 SAY " |
@ 34, 0 SAY " |
@ 34, 47 SAY " | |
@ 34, 51  SAY MKDATE5
@ 34, 60  SAY MKMEAN5+" "+MKLOW5+" "+MKHIGH5
@ 34, 76  SAY " | |
@ 34, 131 SAY " |
@ 35, 0 SAY " |
@ 35, 47 SAY " | |
@ 35, 51  SAY MKDATE6
@ 35, 60  SAY MKMEAN6+" "+MKLOW6+" "+MKHIGH6
RELEASE ALL LIKE MK*
@ 35, 76  SAY " | |
@ 35, 131 SAY " |
@ 36, 0 SAY " |
@ 36, 47 SAY " | |
@ 36, 76  SAY " | |
@ 36, 131 SAY " |
@ 37, 0 SAY " | |
@ 37, 78  SAY " |
@ 39, 0 SAY " |

@ 40, 0 SAY " | STRUCTURAL SECTION |
@ 40, 60  SAY " " | AGG. —ADMIIXTURE— APPL. THICK |
@ 40, 131 SAY " |
@ 41, 0 SAY " | LAYER STRUCTURE DESCRIPTION DATE MATERIAL TYPE |
RATE TYPE POINT RATE CENT TTC LL PI |
@ 41, 131 SAY " |

************
* PRINT STRUCTURAL SECTION
************
RESTORE FROM GEISTRUC ADDITIVE
X = 1
MROW = 42
DO WHILE X < 14
   IF X < 10
      S = SIR(X,1)
   ELSE
      S = SIR(X,2)
   ENDIF
   RESTORE FROM GEISTC&S ADDI
   @ MROW, 0 SAY " | "+MTLAY&S
@ MROW, 12 SAY MISTRUC&S
@ MROW, 21 SAY MZLAYDA&S
@ MROW, 24 SAY MZLAYDB&S
@ MROW, 31 SAY MIDATE&S
@ MROW, 38 SAY MZMAILTP&S
@ MROW, 60 SAY MTAGGRAT&S
@ MROW, 65 SAY MTADMXTP&S
@ MROW, 78 SAY MTADMXR&S
@ MROW, 85 SAY MTAPPLRT&S
@ MROW, 91 SAY MTCENIH&S
IF MZCOUNT = X
    @ MROW, 101 SAY MZTXTRL
    @ MROW, 108 SAY MZLIQLIM
    @ MROW, 114 SAY MZPLAST
ENDIF
RELEASE ALL LIKE MT*
MFILE = "GETISTIC" + S + ".MEM"
DELETE FILE &MFILE
@ MROW, 131 SAY "|
X = X + 1
MROW = MROW + 1
ENDDO
RELEASE ALL LIKE MZ*

@ 55, 0 SAY "|
@ 55, 131 SAY "|
@ 56, 0 SAY
"|--------------------|" 

* AT END DELETE ALL MEM FILES

CLEAR
SET PRINT OFF
SET DEVICE TO SCREEN
DELETE FILE GETENVIR.MEM
DELETE FILE GETLOCAT.MEM
DELETE FILE GETSKID.MEM
DELETE FILE GETPAVCN.MEM
DELETE FILE GETSERVC.MEM
DELETE FILE GETTRAFF.MEM
DELETE FILE GETSTRUC.MEM
DELETE FILE GETDYNA.MEM
DELETE FILE GETFALL.MEM

@ 15, 20 SAY "DONE ......"
RETURN
PROGRAM LISTING

*  SUBSYSTEM: PRINT SUMMARY REPORT
*  PROGRAM NAME: SUM2DIST.PRG   02/04/88
*  PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
*  TAMU/TTI
*  AUTHOR: TREVOR X. PEREIRA
*  PURPOSE: TO PRINT OUT THE INVENTORY UPDATE REPORT. THE FOLLOWING
*  PROGRAMS ARE CALLED:
*     - GETLOC2.PRG
*     - PRIREPT2.PRG
*     - PRIREPT3.PRG
*
* the program first gets all the SID numbers for the particular district and
* stores them in a temporary file. Then data for each SID number is stored
* in memory variable files. When the report is printed out, the appropriate
* files with the variables are opened and deleted when done with them. This
* process continues for each SID number in the temporary file.

* set all parameters
CLOSE DATABASES
CLEAR
SET PRINTER TO &MPORT2
PUBLIC MSID NO, VALID, MCOUNTY, MDISTRICT
DEVICEON = "SET DEVICE TO PRINT"
DEVICEOF = "SET DEVICE TO SCREEN"
STORE " " TO MQUIT, MDISTT
STORE .F. TO VALID
STORE 0 TO MSID_NO, MDISTT

* get the sid number
DO WHILE .NOT. VALID
    CLEAR
    CLEAR TYPEAHEAD
    @ 10, 5 SAY " "
    ACCEPT "Please enter District number: " TO MDISTT
    IF READKEY() = 12
        CLEAR
        RETURN
    ENDIF
    MDIST = VAL(MDISTT)
    @ 10, 0 CLEAR

* Get all SID NO's for the required district and store them to a temporary
* dBASE file
SELECT 1
    USE \PAVEDB\FILES\LOCATION index \PAVEDB\INDEXES\locsid
SELECT 2
    IF .NOT. FILE("\PAVEDB\FILES\SIDSTOR2.DBF")
        CLEAR
        ?"Temporary SID storage file (SIDSTOR2.DBF) not found. Please
check . . ."
    WAIT
    RETURN TO MASTER
ENDIF
USE \PAVEDB\FILES\SIDSTOR2
DELETE ALL
PACK

SELECT 1
LOCATE FOR HWYDIST = MDIST
IF .NOT. FOUND()
   @ 12, 5 SAY "Not found. Please try again."
   WAIT
ENDIF
DO WHILE FOUND()
    MSID_NO = A->SID_NO
    SELECT 2
    APPEND BLANK
    REPLACE B->SID_NO WITH MSID_NO
    SELECT 1
    VALID = .T.
    CONTINUE
ENDDO
SELECT 1
USE
SELECT 2
APPEND BLANK
  REPLACE B->SID_NO WITH 0
USE

* Print out the reports
USE \PAVEDB\FILES\SIDSTOR2
MREC = 1
IF SID_NO = 0
   STORE .F. TO GOON
ELSE
   STORE .T. TO GOON
ENDIF
DO WHILE GOON
   MSID_NO = SID_NO
   DO \PAVEDB\REPORTS\GETLOCT2
   DO \PAVEDB\REPORTS\PRIREPT2
   DO \PAVEDB\REPORTS\PRIREPT3
   MREC = MREC + 1
   USE \PAVEDB\FILES\SIDSTOR2
   GOTO MREC
   IF SID_NO = 0
      GOON = .F.
   ENDIF
   CLEAR
ENDDO
SELECT 1
USE
SELE 2
USE
SELE 3
USE
CLOSE DATABASES
SET PRINTER TO LPT1
RELE MSID NO, VALID, MCOUNTY, MDISTRICT
RETURN
PROGRAM LISTING

* SUBSYSTEM: PRINT INVENTORY UPDATE REPORT
* PROGRAM NAME: SUMM2SID.PRG  01/21/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR:  TREVOR X. PEREIRA
* PURPOSE: TO PRINT OUT THE INVENTORY UPDATE REPORT. THE FOLLOWING
*    PROGRAMS ARE CALLED:
*          - GETLOCT2.PRG
*          - PRIREPT2.PRG
*          - PRIREPT3.PRG

* the program first gets all the data and stores it in separate files
* using memory variables. When the report is printed out, the appropriate
* files with the variables are opened and closed when done with them.

* set all parameters
SET TALK OFF
SET ECHO OFF
CLOSE DATABASES
CLEAR
DEVICEON = "SET DEVICE TO PRINT"
DEVICEOFF = "SET DEVICE TO SCREEN"
SET PRINTER TO &IMPORT2
PUBLIC MSID_NO, VALID, MCOUNTY, MDISTRICT
STORE " " TO MQUIT
STORE .F. TO VALID
STORE 0 TO MSID_NO

* get the sid number
DO WHILE .NOT. VALID
    CLEAR
    @ 10, 5 SAY " "
    ACCEPT "Please enter Section Identification (SID) number: " TO SID_NO
    IF READKEY() = 12
        CLEAR
        RETURN
    ENDIF
@ 0, 0 CLEAR
MSID_NO = VAL(SID_NO)

* check the sid number
STORE 0 TO VAR1, VAR2, VAR3, COMPARE

* calculates the correct Sid Number
VAR1 = INT(MSID_NO/1000)
VAR2 = MOD(INT(MSID_NO/100),10)
VAR2 = VAR2 * 2
VAR3 = MOD(INT(MSID_NO/10),10)
VAR3 = VAR3 * 3
VAR4 = MOD(VAR1,10)+VAR2+VAR3
COMPARE = MOD(VAR4,10)

* compares calculated Sid number with Sid number entered
IF COMPARE = MOD(MSID NO,10)
   STORE .T. TO VALID
ELSE
   @ 19, 10 SAY "Invalid SID Number"
   WAIT
   @ 18, 10 CLEAR
ENDIF

* verifies that Sid number has been entered
IF MSID NO = 0
   STORE .F. TO VALID
   @ 19, 10 SAY "Invalid SID Number"
   WAIT
   @ 18, 10 CLEAR
ENDIF
ENDDO

* get all the data and print the report
IF VALID
   CLEAR
   DO \PAVEDB\REPORTS\GETLOCT2
   DO \PAVEDB\REPORTS\PRTREPT2
   DO \PAVEDB\REPORTS\PRTREPT3
ENDIF

RELEASE MSID NO, VALID, MCOUNTY, MDISTRICT
CLOSE DATABASES
RETURN
MSID NO = SID NO
DO \PAVEDB\REPORTS\GETLOC2
DO \PAVEDB\REPORTS\PRIREPT2
DO \PAVEDB\REPORTS\PRIREPT3
MREC = MREC + 1
USE \PAVEDB\FILES\LOCATN2
GOTO MREC
IF SID NO = 0
  COON = .F.
ENDIF
CLEAR
ENDDO
SELE 1
  USE
SELE 2
  USE
SELE 3
  USE
CLOSE ALTE
RELEASE MSID NO, MOUNTY, MDISTRICT
CLOSE DATABASES
RETURN
* SUBSYSTEM: PRINT INVENTORY UPDATE REPORT
* PROGRAM NAME: SUM2FILE.PRG  06/14/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TII
* AUTHOR: TREvor X. PEREIRA
* PURPOSE: TO PRINT OUT THE INVENTORY UPDATE REPORT FOR ALL THE SID
* NUMBERS. THE FOLLOWING PROGRAMS ARE CALLED:
* - GETLOCT2.PRG
* - PRIREPT2.PRG
* - PRIREPT3.PRG
* 
* the program first gets all the data and stores it in separate files
* using memory variables. When the report is printed out, the appropriate
* files with the variables are opened and closed when done with them.
* THE REPORT FOR EACH SID IS PRINTED OUT TO A FILE.
* 
* set all parameters
SET TALK OFF
SET ECHO OFF
CLOSE DATABASES
CLEAR
SET PRINTER TO &IMPORT2
DEVICEON = "$SET DEVICE TO PRINT"
DEVICEOFF = "$SET DEVICE TO SCREEN"
PUBLIC MSID_NO, MCOUNTY, MDISTRICT
STORE " " TO MQUIT
STORE 0 TO MSID_NO

* get the sid number
IF FILE("\PAVED\FILES\LCATIN2.DBF")
    DELE FILE \PAVED\FILES\LCATIN2.DBF
ENDIF
COPY FILE \PAVED\FILES\LOCATION.DBF TO \PAVED\FILES\LCATIN2.DBF
USE \PAVED\FILES\LCATIN2
APPEND BLANK
REPLACE SID NO WITH 0
GOTO TOP
MREC = 1
IF SID NO = 0
    STORE .F. TO GOON
ELSE
    STORE .T. TO GOON
ENDIF
CLEAR
DO WHILE GOON
    IF READKEY() = 12
        CLEAR
        RETURN
    ENDIF
* SUBSYSTEM: PRINT INVENTORY UPDATE REPORT
* PROGRAM NAME: GETLOC2.PRG 02/04/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TIT
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO GET DATA FROM THE FOLLOWING FILES TO PRINT THE SUMMARY
* REPORT: 1) LOCATION DBASE FILE
* 2) LAYER IDENTIFICATION DBASE FILE
* 3) SUBGRADE LAYER DBASE FILE
* 4) SURFACE LAYER DBASE FILE
* 5) GEOMETRIC & SHOULDER INFO LAYER DBASE FILE
* 6) LAYER THICKNESS ACROSS THE ROAD

***********
* GET LOCATION PART OF THE REPORT AND STORE THE DATA IN A FILE
***********

@ 2,14 SAY "PROCESSING SID NUMBER " + SIR(MSID NO,4)
@ 5, 5 SAY "Getting data for the Location section of the report ...." 

close all

* assign databases to different work areas
SELECT 1
use \PAVEDEB\FILES\LOCATION INDEX \PAVEDEB\INDEXES\LOCSID

SELECT 2
USE \PAVEDEB\FILES\CNTYTB1 INDEX \PAVEDEB\INDEXES\CITYBLNO

* locate for the requested sid number
SELECT 1
seek msid no
if .not. found()
   okay = .F.
clear
   @ 19, 10 say "SID number not found. Please try again ..."
   WAIT
   @ 18, 10 CLEAR
   RETURN
else
   okay = .T.
endif
STORE 0 TO MCOUNTY
STORE 0 TO MDISTRICT

* if the sid number is found, get all the required data
if okay
   mlhwydst = LTRIM(SIR(hwydist,2))
   MDISTRICT = HWDIST
MCOUNTY = CNTYNUM
mlcntynm = LITRIM(STR(cntynum,3))
SELECT 2
SEEK A->CNTYNUM
IF FOUND()
   MLCNTYNM = LITRIM(CNTYNAME)
ENDIF
SELECT 1
mcontrol = LITRIM(STR(control,4))
msectn = LITRIM(STR(section,2))
MLCONSEC = MCONTRL + "-" + MSECIN
MNUMLANE = LITRIM(STR(NUMLANES,2))
mprefix = hwprefix
mhwnum = LITRIM(STR(hwnum,4))
msuffix = hwsuffix
MLHWY = MPREFIX + " " + LITRIM(MHWYNUM) + " " + MSUFFIX
MBEGIN = LITRIM(STR(BEGMPST,3))
MEND =
RIGHT((STUFF(STR(BMPSTDIS,2),1,(2-LEN(LITRIM(STR(BMPSTDIS,2))))),"0")),2)
MPOSTB = MBEGIN+EDISSIGN+MEND
MENDD =
RIGHT((STUFF(STR(BMPSTDIS,2),1,(2-LEN(LITRIM(STR(BMPSTDIS,2))))),"0")),2)
MENDP = LITRIM(STR(ENDMPST,3))
MPOSTE = MENDP+EDISSIGN+MEND
MLPOST = MPOSTB+"" TO ""+MPOSTE
IF ACTIVFLAG
   MLINACTIV = "NO"
ELSE
   MLINACTIV = "YES"
ENDIF
mlbegmpt = LITRIM(STR(begmpt,6,3))
mlendmpt = LITRIM(STR(endmpt,6,3))
mlaneid = laneid
IF PREVSID = 0
   MLPREVSD = "-"
ELSE
   MLPREVSD = LITRIM(STR(prevsid,4))
ENDIF
IF NEXTSID = 0
   MLNEXTSD = "-"
ELSE
   MLNEXTSD = LITRIM(STR(nextsid,4))
ENDIF
MLFUCNLS = LITRIM(STR(FUCNLS,2))
ENDIF
SELECT 1
USE
SELECT 2
USE

***************
* get PAVEMENT INFORMATION FROM geometric and shoulder LAYER FILE
***************
SELECT 1
use \PAVEDB\FILES\geosho index \PAVEDB\INDEXES\geonxy

SELECT 2
USE \PAVEDB\FILES\PAVETYPE

SELECT 1
LOCATE FOR SID_NO = msid_no .AND. STRUCNUM = 1
if found()
   DO CASE
      CASE WIDENFLG = 0
         MLWIDEN = "NO WIDENING"
      CASE WIDENFLG = 1
         MLWIDEN = "WIDENING"
      CASE WIDENFLG = 2
         MLWIDEN = "UNUSUAL WIDENING"
   ENDCASE
   MPAVEM = PAVETYP
SELECT 2
LOCATE FOR PAVECODE = MPAVEM
IF FOUND()
   MLBASETYP = LTRIM(BASETYPE)
   MLBASETHK = LTRIM(BASETHK)
   MLBASESEAL = LTRIM(BASESEAL)
ELSE
   MLBASETYP = ""
   MLBASETHK = ""
   MLBASESEAL = ""
ENDIF
ELSE
   MLWIDEN = ""
   MLBASETYP = ""
   MLBASETHK = ""
   MLBASESEAL = ""
ENDIF
* save all location variables to a file
SAVE ALL LIKE ML* TO GETLOCT2
RELEASE ALL
SELE 1
USE
SELE 2
USE

***************
* GET LAYER THICKNESS ACROSS THE ROAD INFORMATION
***************
USE \PAVEDB\FILES\LAYTHICK INDEX \PAVEDB\INDEXES\LAYINDEX
SEEK STR(MSID_NO,4)
IF FOUND()
   DO WHILE .NOT. EOF() .AND. SID_NO = MSID_NO
      SKIP
   ENDOO
   SKIP -1
DO CASE
    CASE FC3DIS <> 0
        MDIHICK = STR(FC3DIS,4,1)
    CASE FC2DIS <> 0
        MDIHICK = STR(FC2DIS,4,1)
    CASE FC1DIS <> 0
        MDIHICK = STR(FC1DIS,4,1)
    OTHERWISE
        MDIHICK = " "
ENDCASE
ELSE
    MDIHICK = " "
ENDIF
SAVE ALL LIKE MD* TO GETDISIN
RELEASE ALL LIKE MD*

*****************************************************************************
* GET OTHER GEOMETRIC AND SHOULDER INFORMATION
*****************************************************************************
USE \PAVEDB\FILES\geosho index \PAVEDB\INDEXES\geondx
SEEK STR(msid_no,4)
X = 0
DO WHILE X < 16
    IF X < 10
        S = STR(X,1)
    ELSE
        S = STR(X,2)
    ENDIF
    MSGLANEWD&S = " "
    MSGOUTSHD&S = " "
    MSGSHDSUR&S = " "
    MSGSHDSE&S = " "
    MSGSHDST&S = " "
    MGNUMLAN&S = " "
    MSGSTRUC&S = " "
    MGPAVE&S = " "
    MSGWIDEN&S = " "
    X = X + 1
ENDDO

IF found()
    X = 0
    DO WHILE .NOT. EOF() .AND. SID_NO = MSID_NO
        IF X > 15
            EXIT
        ENDIF
        IF X < 10
            S = STR(X,1)
        ELSE
            S = STR(X,2)
        ENDIF
        MSGLANEWD&S = STR(LANEWID,4,1)
MGOUSHD&S = STR(OUTHOWD,4,1)
MGOUSHDSUR&S = STR(SHOSFTYP,1)
MGOUSHDBSE&S = STR(SHOBSTYP,2)
MGOUSHDSFT&S = STR(SHOFTHIC,5,2)
MGOUSHDST&S = STR(SHOBSTHIC,5,2)
MGOUSRC&S = STR(STRUCNUM,2)
MGOUPAV&S = STR(PAVETYP,2)
MGOUIDEN&S = STR(WIDENFLG,1)
X = X + 1
SKIP
ENDDO
ENDIF
SAVE ALL LIKE MG* TO GEIGEOSH
RELEASE ALL LIKE MG*
USE

**********************
* GET STRUCTURAL SECTION OF THE REPORT AND SAVE THE DATA IN VARIABLES
**********************

* variable mcount is to find out how many layers there are for the SID #.
* This will help determine the layer the subgrade goes on. This is needed
* if layer # 1 is the top most layer and not the subgrade.

@ 8, 5 SAY "Getting data for the Structural Section ...."
STORE " " TO MTPLAST, MTILQLIM, MTXTRL
MOUNT = 1

set path to \pavedb\files, \pavedb\indexes
set view to \pavedb\files\strcsec2
seek str(msid_no,4)

* if the sid number is found, get the required data
if found()
do while .NOT. EOF() .AND. sid_no = msid_no
   skip
endo
skip -1
X = 1
do while .NOT. EOF() .AND. sid_no = msid_no
   IF X < 10
      S = STR(X,1)
   ELSE
      S = STR(X,2)
   ENDIF
   MTSTRUC&S = str(structnum,2)
   MTLAY&S = str(laynum,2)
   MLAYDES&S = laydesc
   MIDATE&S = str(jobcmno,2)+"/"+ str(jobcmpyr,2)
   MLAYMAT&S = laymatcl
   IF WIDENLMO = 0 .OR. WIDENLYR = 0
      MIWIDAT&S = " "
   ELSE
      MIWIDEYR = STR(WIDENLYR,2)
MWIDEMO = LTRIM(STR(WIDENIMO,2))
MWIDAT&S = MWIDEMO+"/"+MWIDEYR
ENDIF
IF CENTTHK = 0
  MICENTHK&S = " "
ELSE
  mcentthk&S = str(centthk,5,2)
ENDIF
IF AGAPPLRT = 0
  MTAGGRAT&S = " "
ELSE
  mtaggrat&S = str(agapplrt,3)
ENDIF
mtadmtp&S = admtyp
IF ADMXPER = 0
  MTADMXPR&S = " "
ELSE
  mtdadmxp&S = str(admxper,5,2)
ENDIF
IF ASAPPLRT = 0
  MITAPPLRT&S = " "
ELSE
  mtapplrt&S = str(asapplrt,4,2)
ENDIF
SAVE ALL LIKE MT* TO GETSTC&S
RELEASE ALL LIKE MT*
X = X + 1
MOUNT = MOUNT + 1
skip -1
IF BOF()
  SKIP 35
ENDIF
enddo
SELE 1
USE
SELE 2
USE
SELE 3
USE
X = MOUNT
MOUNT = MOUNT - 1
* initialize rest of variables
DO WHILE X < 14
  IF X < 10
    S = STR(X,1)
  ELSE
    S = STR(X,2)
  ENDIF
STORE " " TO MISTRUC&S, MITAY&S, MIJÖBER&S, MIJÖBYR&S, MICENTHK&S,
MTAGGRAT&S, MTADMXPR&S
STORE " " TO MTAPPLRT&S, MITDATE&S, MIAYDES&S, MIAYMAT&S, MIWIDAT&S
SAVE ALL LIKE MT* TO GEISTC&S
RELEASE ALL LIKE MT*
X = X + 1
171
ENDO

* get the material type and the layer names from the tables
SELECT 6
  USE \\PAVEDB\\FILES\\MATLTLBL
SELECT 7
  USE \\PAVEDB\\FILES\\LAYERTBL
SELECT 6
MZCOUNT = MCOUNT
X = 1
DO WHILE X < 14
  IF X < 10
    S = STR(X,1)
  ELSE
    S = STR(X,2)
  ENDIF
SELECT 6
TYPE1 = MLAYMAT&S
TYPE2 = "TYPE1"
TYPE3 = TYPE(TYPE2)
IF TYPE3 = "N"
  LOCATE FOR CODE = MLAYMAT&S
  IF FOUND()
    MZMATYPE&S = F->MATDESC
    MZLAYDA&S = F->LAYRDES
  ELSE
    MZMATYPE&S = " "
    MZLAYDA&S = " "
 ENDIF
ELSE
  MZMATYPE&S = " "
  MZLAYDA&S = " "
ENDIF
SELECT 7
TYPE1 = MLAYDES&S
TYPE2 = "TYPE1"
TYPE3 = TYPE(TYPE2)
IF TYPE3 = "N"
  LOCATE FOR CODE = MLAYDES&S
  IF FOUND()
    MZLAYDB&S = G->CODE_DESC
  ELSE
    MZLAYDB&S = " "
  ENDIF
ELSE
  MZLAYDB&S = " "
ENDIF
X = X + 1
ENDDO
ELSE
  * initialize all variables
  X = 1
DO WHILE X < 14
  IF X < 10
    S = STR(X,1)
  ELSE
S = STR(X,1)
ELSE
   S = STR(X,2)
ENDIF
STORE " " TO MISTRUC&S, MTLAY&S, MTCENHKS&S, MTAGGRAT&S, MTADMXPR&S, MZMATTYP&S
STORE " " TO MTAPPLRT&S, MTDATE&S, MLAYDES&S, MLAYMAT&S, MWIDAT&S, MZLAYDA&S, MZLAYDB&S
SAVE ALL LIKE MT* TO GETISC&S
RELEASE ALL LIKE MT*
X = X + 1
ENDDO
STORE 0 TO MZCOUNT
endif
MCOUNT = MCOUNT - 1
close DATABASES
set path to

*******************************
* get the subgrade file data
*******************************
use \PAVEDB\FILES\subgrade index \PAVEDB\INDEXES\subgnidx
seek str(msid_no,4)
if found()
   MZplast = str(plastix,4,1)
   MZliqlim = str(liqlim,4,1)
   MZtxtrl = str(txtriax1,3,1)
   MZPRPASS = STR(PFSV200,4,1)
   MZPERM = STR(PERMIX,5,2)
ELSE
   MZplast = " "
   MZliqlim = " "
   MZtxtrl = " "
   MZPRPASS = " "
   MZPERM = " "
endif
close all

* save all the variables to file
save all likeMZ* to getstruc
RELEASE ALL LIKE MZ*
return
PROGRAM LISTING

* * SUBSYSTEM: PRINT SUMMARY REPORT
* PROGRAM NAME: PRIREPORT2.PRG 02/04/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO PRINT OUT THE INVENTORY UPDATE REPORT USING THE DATA
STOURED
* IN MEMORY VARIABLE FILES.
*
CLEAR
@ 10, 5 SAY "Printing report One (Inventory Update Sheet) for "+str(msid_no)
ON ERROR DO \PAVEDB\REPORTS\ERR_PRINT
SET PRINT TO &REPORT2
ON ERROR DO \PAVEDB\ERROR
&DEVICEON
@ 0, 0 Say ""[&llo"
mdate = dtoc(date())
SET ALIEN ON
@ 0, 145 say "Date: " + mdate
@ 1, 0 say "---------------------------------------------------------------------

@ 2, 0 say "LOCATION
| GEOMETRIC AND SHOULDER INFORMATION
|
*************
* PRINT LOCATION SECTION
*************
RESTORE FROM GETLC2 ADDITIVE
@ 3, 0 SAY ""| SECTION ID NO: "+LTRIM(STR(MSID NO,4))
@ 3, 57 SAY ""| TYPE OUTSIDE SHOULDER
SHOULDER SHOULDER SHOULDER"
@ 3, 161 SAY """
@ 4, 0 SAY ""| DISTRICT NO: "+MLHWYIDT
@ 4, 57 SAY ""| STRUCTURE OF LANE SHOULDER SURFACE BASE
SURFACE | BASE WIDENING"
@ 4, 161 SAY """
RELEASE MLHWYIDT
@ 5, 0 SAY ""| COUNTY NO/NAME: "+MLCITYNM+"/"+MLCNTYNAM
@ 5, 57 SAY ""| NUMBER PAVEMENT WIDTH WIDTH TYPE TYPE
THICKNESS THICKNESS FLAG"
@ 5, 161 SAY """
@ 6, 0 SAY ""| CONTROL-SECTION: "+MLCONISEC
@ 6, 57 SAY ""| "-----------------" "-----------------" "-----------------"

@ 6, 161 SAY """
@ 7, 0 SAY ""| HIGHWAY: "+MLHWY
RESTORE FROM GETGEOISH ADDITIVE
@ 7, 57 SAY ""| "+MGSTRUCO+" "+MGPAVEO+" "+MGLANEWD+"
"+MGOUTSHDO+" "+MGSHELDSO+" "+MGSHELSEO+"

174
@ 7, 161 SAY "" ""MGSHDSTO+" ""MGSHDBSTO+" ""MGWIDENO
@ 8, 0 SAY "" MILE POINTS: "" ""MGSTMPN+" ""MGENDMPN
@ 8, 57 SAY "" ""MGSTUC1+" ""MGPAVE1+" ""MGLANED1+
""MGOUTSD1+" ""MGSHDSUR1+" ""MGSHDBSE1+"
""MGSHDST1+" ""MGSHDBST1+" ""MGWIDEN1
@ 8, 161 SAY "" ""***************
* PRINT GEOMETRIC & SHOULDER SECTION
***************
@ 9, 0 SAY "" ""LANE: "" ""MGLANEDT
@ 9, 57 SAY "" ""MGSTUC2+" ""MGPAVE2+" ""MGLANED2+
""MGOUTSHTD2+" ""MGSHDSUR2+" ""MGSHDBSE2+"
""MGSHDST2+" ""MGSHDBST2+" ""MGWIDEN2
@ 9, 161 SAY "" ""MILE POST: "" ""MGPOST
@ 10, 0 SAY "" ""MGSTUC3+" ""MGPAVE3+" ""MGLANED3+
""MGOUTSHTD3+" ""MGSHDSUR3+" ""MGSHDBSE3+"
""MGSHDSPT3+" ""MGSHDBST3+" ""MGWIDEN3
@ 10, 161 SAY "" ""PREVIOUS SID: "" ""MGPREVSD
@ 11, 0 SAY "" ""MGSTUC4+" ""MGPAVE4+" ""MGLANED4+
""MGOUTSHTD4+" ""MGSHDSUR4+" ""MGSHDBSE4+"
""MGSHDST4+" ""MGSHDBST4+" ""MGWIDEN4
@ 11, 161 SAY "" ""NEXT SID: "" ""MGNEXTSD
@ 12, 0 SAY "" ""MGSTUC5+" ""MGPAVE5+" ""MGLANED5+
""MGOUTSHTD5+" ""MGSHDSUR5+" ""MGSHDBSE5+"
""MGSHDST5+" ""MGSHDBST5+" ""MGWIDEN5
@ 12, 161 SAY "" ""FUNCTIONAL CLASS: "" ""MGFUNCLS
@ 13, 0 SAY "" ""MGSTUC6+" ""MGPAVE6+" ""MGLANED6+
""MGOUTSHTD6+" ""MGSHDSUR6+" ""MGSHDBSE6+"
""MGSHDST6+" ""MGSHDBST6+" ""MGWIDEN6
@ 13, 161 SAY "" ""TYPE OF PAVEMENT: "" ""MGBASETP
@ 14, 0 SAY "" ""MGSTUC7+" ""MGPAVE7+" ""MGLANED7+
""MGOUTSHTD7+" ""MGSHDSUR7+" ""MGSHDBSE7+"
""MGSHDSPT7+" ""MGSHDBST7+" ""MGWIDEN7
@ 14, 161 SAY "" ""MGBASETHK
@ 15, 0 SAY "" ""MGSTUC8+" ""MGPAVE8+" ""MGLANED8+
""MGOUTSHTD8+" ""MGSHDSUR8+" ""MGSHDBSE8+"
""MGSHDSPT8+" ""MGSHDBST8+" ""MGWIDEN8
@ 15, 161 SAY "" ""MGBASESEAL
@ 16, 0 SAY "" ""MGSTUC9+" ""MGPAVE9+" ""MGLANED9+
""MGOUTSHTD9+" ""MGSHDSUR9+" ""MGSHDBSE9+"
""MGSHDSPT9+" ""MGSHDBST9+" ""MGWIDEN9
@ 16, 161 SAY "" ""MLWIDEN
"MGSTRUCT10" "MGPAVE10" "MGLANEW10"
"MGOUTSHD10" "MGSHDSUR10" "MGSHDBSE10"
"MGSHDSFT10" "MGSHDBST10" "MGWIDEN10"
"MGSTRUCT11" "MGPAVE11" "MGLANEW11"
"MGOUTSHD11" "MGSHDSUR11" "MGSHDBSE11"
"MGSHDSFT11" "MGSHDBST11" "MGWIDEN11"
"MGOUTSHD12" "MGSHDSUR12" "MGSHDBSE12"
"MGSHDSFT12" "MGSHDBST12" "MGWIDEN12"
"MGSTRUCT13" "MGPAVE13" "MGLANEW13"
"MGOUTSHD13" "MGSHDSUR13" "MGSHDBSE13"
"MGSHDSFT13" "MGSHDBST13" "MGWIDEN13"
"MGSTRUCT14" "MGPAVE14" "MGLANEW14"
"MGOUTSHD14" "MGSHDSUR14" "MGSHDBSE14"
"MGSHDSFT14" "MGSHDBST14" "MGWIDEN14"
"MGSTRUCT15" "MGPAVE15" "MGLANEW15"
"MGOUTSHD15" "MGSHDSUR15" "MGSHDBSE15"
"MGSHDSFT15" "MGSHDBST15" "MGWIDEN15"

@ 21, 0 SAY "RELEASE ALL LIKE ML*

@ 21, 57 SAY "MGSTRUCT14" "MGPAVE14" "MGLANEW14"
"MGOUTSHD14" "MGSHDSUR14" "MGSHDBSE14"
"MGSHDSFT14" "MGSHDBST14" "MGWIDEN14"

@ 22, 0 SAY "RELEASE ALL LIKE MG*

@ 23, 0 SAY

@ 24, 0 SAY

@ 25, 0 SAY "STRUCTURAL SECTION  THICK
  WIDENING AGG.  ADMIXTURE  APPL % PASSING  PERM."

@ 25, 161 SAY "*

@ 26, 0 SAY "STRUCTURE LAYER DESCRIPTION  CENT  MATERIAL TYPE
  DATE  DATE  RATE  TYPE  PCNT  RATE  200 SIEVE  TTC  LL  PI
  INDEX"

@ 26, 161 SAY "*

@ 27, 0 SAY "*

************
* PRINT STRUCTURAL SECTION
************
RESTORE FROM GETSTRUC ADDITIVE
X = 1
MRROW = 28
DO WHILE X < 14
  IF X < 10
    S = STR(X,1)
  ELSE
    S = STR(X,2)
  ENDIF
RESTORE FROM GETSTC&S ADDI
@ MRROW, 0  SAY "|
@ MRROW, 6  SAY MTRSTRUC&S
@ MRROW, 14  SAY MLAY&S
@ MRROW, 21  SAY MZLAYDA&S
@ MRROW, 24  SAY MZLAYDB&S
@ MRROW, 31  SAY MTЦентр&S
@ MRROW, 38  SAY MZМТип&S
@ MRROW, 60  SAY MIDATE&S
@ MRROW, 70  SAY MIIIDAT&S
@ MRROW, 76  SAY MTаGRAT&S
@ MRROW, 94  SAY MTAADMXPR&S
@ MRROW, 101  SAY MTAAPLRT&S
IF MZCOUNT = X
    @ MRROW, 108  SAY MZPRPASS
    @ MRROW, 116  SAY MZTXTRL
    @ MRROW, 121  SAY MZLIQLIM
    @ MRROW, 127  SAY MZPLAST
    @ MRROW, 132  SAY MZPERM
ENDIF
RELE ALL LIKE MT*
MFILE = "GETSTC" + S + ".MEM"
DELE FILE &MFILE
@ MRROW, 161  SAY "|
X = X + 1
MRROW = MRROW + 1
ENDDO
RELE ALL LIKE MZ*
X = 1
MRROW = 41
DO WHILE X < 4
  IF X < 10
    S = STR(X,1)
  ELSE
    S = STR(X,2)
  ENDIF
@ MRROW, 0  SAY "|
@ MRROW, 161  SAY "|
X = X + 1
MRROW = MRROW + 1
ENDDO
RELEASE ALL LIKE MT*
@ 44, 0  SAY
"|--------------------------|
--------------------------
* AT END DELETE ALL MEM FILES
SET ALITE OFF
SET PRINT OFF
&DEVICEOFF
DELETE FILE GETGEOSH.MEM
DELETE FILE GETSTRUC.MEM
RETURN
PROGRAM LISTINGS

* SUBSYSTEM: PRINT SUMMARY REPORT
* PROGRAM NAME: PRTRPT3.PRG 02/04/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMI/TTI
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO PRINT OUT THE SECOND PAGE OF THE INVENTORY UPDATE REPORT
*

@ 15, 10 SAY "Printing report Two (Blank) ...."
ON ERROR DO \PAVE\DB\REPORTS\ERR_PRINT
SET PRINT TO &MPRT2
ON ERROR DO \PAVE\DB\ERROR
&DEVICEN
@ 0, 0 Say "[&110"
SET ALTE ON
mdate = dtoc(date())
@ 0, 145 say "Date: " + mdate
@ 1, 0 say "

@ 2, 0 say "LOCATION"
| GEOMETRIC AND SHOULDER INFORMATION |

**************
* PRINT LOCATION SECTION
**************
RESTORE FROM GETLOC2 ADDITIVE
@ 3, 0 say "| SECTION ID NO: " + LTRIM(STR(MSID_NO,4))
@ 3, 57 say "| TYPE OUTSIDE SHOULDER SHOULDER SHOULDER"
@ 3, 161 say "|
@ 4, 0 say "| DISTRICT NO: " + MLHWYDST
@ 4, 57 say "| STRUCTURE OF LANE SHOULDER SURFACE BASE
       SURFACE BASE WIDENING"
@ 4, 161 say "|
RELEASE MLHWYDST
@ 5, 0 say "| COUNTY NO/NAME: " + MLCNTYNM+"/"+MLCNTYNAM
@ 5, 57 say "| NUMBER PAVEMENT WIDTH WIDTH TYPE TYPE
       THICKNESS THICKNESS FLAG"
@ 5, 161 say "|
@ 6, 0 say "| CONTROL-SECTION: " + MLCNTSEC
@ 6, 57 say "| -------------- -------------- -------------- --------------
      -------------- -------------- -------------- --------------"
@ 6, 161 say "|
@ 7, 0 say "| HIGHWAY: " + MLHWY
@ 7, 57 say "| -------------- -------------- --------------
      -------------- --------------
@ 7, 161 say "|
@ 8, 0 say "| MILE POINTS: " + MLBEGIN+"-"+MLENDMPN

179
@ 8, 57 SAY "| |"
@ 8, 161 SAY "n"

***************
* PRINT GEOMETRIC & SHOULDER SECTION
***************
@ 9, 0 SAY "| LANE: +MLANEID"
@ 9, 57 SAY "| +MLPREVS
@ 9, 161 SAY "n"
@ 10, 0 SAY "| MILE POST: +MLPOST"
@ 10, 57 SAY "n"
@ 10, 161 SAY "n"
@ 11, 0 SAY "| PREVIOUS SID: +MLPREVS"
@ 11, 57 SAY "| +MLFUNCLS"
@ 11, 161 SAY "n"
@ 12, 0 SAY "| NEXT SID: +MLNEXTS"
@ 12, 57 SAY "n"
@ 12, 161 SAY "n"
@ 13, 0 SAY "| FUNCTIONAL CLASS: +MLFUNCLS"
@ 13, 57 SAY "| +MLBASETP"
@ 13, 161 SAY "n"
@ 14, 0 SAY "| TYPE OF PAVEMENT: +MLBASETP"
@ 14, 57 SAY "| +MLBASETHK"
@ 15, 0 SAY "| +MLBASETHK"
@ 15, 57 SAY "n"

@ 16, 0 SAY "| +MLBASESEAL"
@ 16, 22 SAY "| RESTORE FROM GETDISTIN ADDITIVE"
@ 16, 57 SAY "| LAYER THICKNESS ACROSS THE ROAD"
@ 16, 132 SAY "CURRENT SURFACE WIDTH " + MDTHICK
@ 16, 161 SAY "n"
@ 17, 0 SAY "| "
@ 17, 22 SAY MDWIDTH
@ 17, 57 SAY "| THICKNESS FROM CENTER"
@ 17, 161 SAY "n"
@ 18, 0 SAY "| INACTIVE SID: +MLINACTIV"
@ 18, 57 SAY "| STRUCTURE LAYER ---------------------------------- AT

-----------------------------
@ 18, 161 SAY "n"
@ 19, 0 SAY "| NUMBER OF LANES: +MINULAN"
@ 19, 57 SAY "| NUMBER NUMBER 3rd POS 2nd POS 1st POS CENTER"
@ 20, 0 SAY "| 3rd POS 2nd POS 1st POS"
@ 19, 161 SAY "n"
@ 20, 0 SAY "| "
@ 20, 57 SAY "| "
@ 20, 161 SAY "|
RELEASE ALL LIKE ML*
@ 21, 0 SAY "|
@ 21, 59 SAY "|
@ 22, 161 SAY "|
@ 22, 0 SAY "|
@ 22, 57 SAY " |
@ 23, 161 SAY "|
@ 23, 0 SAY "|
@ 23, 59 SAY "|
@ 24, 161 SAY "|
@ 24, 0 SAY "|
@ 24, 57 SAY " |
@ 25, 161 SAY "|
@ 25, 0 SAY "|
@ 25, 59 SAY "|
@ 26, 161 SAY "|
@ 26, 0 SAY "|
@ 26, 57 SAY " |
@ 27, 161 SAY "|
@ 27, 0 SAY "|
@ 27, 59 SAY "|
@ 28, 161 SAY "|
@ 28, 0 SAY "|---------------------------------------------------------------------|
@ 28, 161 SAY "|
@ 29, 0 SAY "|---------------------------------------------------------------------|
@ 30, 0 SAY "|STRUCTURAL SECTION          THICK
WIDENING AGG.    ---ADMIXTURE--- APPL % PASSING
PERM."
@ 30, 161 SAY "|
@ 31, 0 SAY "| STRUCTURE LAYER DESCRIPTION CENT MATERIAL TYPE DATE
DATE RATE TYPE PCNT RATE 200 SIEVE TTC LL PI
INDEX"
@ 31, 161 SAY "|
@ 32, 0 SAY "|-----------------------------------------|-----------------------|
@ 32, 161 SAY "|

*********
* PRINT STRUCTURAL SECTION
*********
X = 1
MROW = 33
DO WHILE X < 12
    IF X < 10
        S = STR(X,1)

181
ELSE
    S = STR(X, 2)
ENDIF

@ MROW, 0 SAY "|"
@ MROW, 3 SAY "_____"
@ MROW, 13 SAY "_______"
@ MROW, 21 SAY "_______"
@ MROW, 31 SAY "_______"
@ MROW, 40 SAY "__________________"    && STRUCTURE NUMBER
@ MROW, 53 SAY "_______"    && LAYER NUMBER
@ MROW, 61 SAY "_______"    && DESCRIPTION
@ MROW, 69 SAY "_______"    && CENTER THICKNESS
@ MROW, 77 SAY "_______"    && MATERIAL TYPE
@ MROW, 88 SAY "_______"    && DATE
@ MROW, 95 SAY "_______"    && WIDEN DATE
@ MROW, 103 SAY "_______"    && AGGREGATE RATE
@ MROW, 112 SAY "_______"    && ADMIXTURE TYPE
@ MROW, 119 SAY "_______"    && ADMIXTURE PERCENT
@ MROW, 127 SAY "_______"    && APPLICATION RATE
@ MROW, 135 SAY "_______"    && % PASSING SIEVE
@ MROW, 161 SAY "|"
X = X + 2
MROW = MROW + 1
@ MROW, 0 SAY "|"
@ MROW, 161 SAY "|"
MROW = MROW + 1
ENDDO

@ 44, 0 SAY "|____________________________________|
|____________________________________|

* AT END DELETE ALL MEM FILES
SET ALIPE OFF
CLEAR
SET PRINT OFF
&DEVICEOFF
DELETE FILE GETILOC2.MEM
DELETE FILE GETDISTIN.MEM

@ 15, 20 SAY "DONE ....."
RETURN
CHAPTER 4

EDIT AND UPDATE SUBSYSTEM
GENERAL NARRATIVE

The Edit & Update Menu System allows the user to edit or update the data in the Master files and the 7 tables. This portion is divided into 4 sections. The sections are:

- Edit and Update Driver
- Monitoring Data Update
- Inventory Data Update
- Traffic Date Update.

In addition to driving the edit and update process, EDITUPDT.PRG also updates the table files. The user must be an experienced dBASE programmer because the master tables are modified directly (without edit checks) using the dBASE BROWSE command. The table files can be found in the subdirectory \PAVEDB\FILES while the program EDITUPDT.PRG is in \PAVEDB\EDITUPDT.

The PES data is copied from the Original PES Tape to a disk file by a stand alone batch file (i.e. the batch file is not included in the Flexible Pavement Menu System). The batch file (READPES.BAT) is documented in Section 2 of this Chapter. This data is then converted to 6 comma delimited data files by the program PESMOD.PAS. These delimited files are further converted to dBASE files by PESUP.PRG and are used to update the monitoring files (Skid, Visual, Serviceability Index and Falling Weight). The PES programs can be found in the subdirectory \PAVEDB\EDITUPDT\PES.

Updating the Inventory files is divided into two parts - adding inventory data and changing inventory data. The data is added manually by the user through the program ENCRLAYR.PRG while the data is modified through CHNGLAYR.PRG. The Inventory files modified are Location, Layer Identification, Layer Thickness Across The Road, Geometric and Shoulder Information, Surface and Subgrade. The Add and Change programs can be found in the subdirectory \PAVEDB\EDITUPDT.

The traffic data is copied from the Roadway Inventory (RIFILE) Tape File to a disk file by a stand alone batch program (READTRAF.BAT). The batch file is documented in section 4 of this Chapter. This data is then converted to a comma delimited format to be added to a dBASE File. This dBASE file is then used to update the Traffic File. Other than copying the tape file to disk, TRAFUPD.PRG controls the traffic update process. The traffic programs can be found in the subdirectory \PAVEDB\EDITUPDT\TRAFFIC.

Each Sub-section has the following information:

- Program Narrative
- Program Flow Diagram
- Program Specification
- Menu Screens
- Sample Reports
- Program Listings

185
Section 1: Edit and Update Driver
Edit and Update Driver
Program Narrative

The Edit & Update program (EDITUPDT.PRG) produces the main menu screen for the Edit & Update Subsystem. It also produces the menu screen to edit or change the tables. The screens generated are displayed in this section.

The Edit & Update program also calls the programs PESMOD.PAS and PESUP.PRG to add PES data to the monitoring files. To add data to the Traffic file, the program TRAFUPD.PRG is called which obtains traffic data from the Roadway Inventory File. EDITUPDT.PRG also calls the programs to update the data in the Inventory Files. To add inventory data, ENTRLAYR.PRG is called and to change inventory data, CHNGLAYR.PRG is called. To update the tables, EDITUPDT.PRG uses the browse command to let the user change or add data.
EDIT AND UPDATE
SCREENS

The following screens are produced by EDITUPDT.PRG except for the Main Menu.

TEXAS FLEXIBLE PAVEMENT DATABASE
MAIN MENU

1 - Inquiry
2 - Reports
3 - Edit & Update
4 - Applications
5 - Backup
6 - Installation
7 - Reindex Master Files
Q - QUIT

Above is the Main Menu of the Texas Flexible Pavement Database System. When the Edit & Update option is chosen, then screen 3.0 is displayed.
EDIT AND UPDATE
MENU SCREENS (cont.)

TEXAS FLEXIBLE PAVEMENT DATABASE 3.0
Edit & Update

1 - Pavement Condition Data
2 - Inventory Data
3 - Traffic Data
4 - Tables

OPTION ————

Choice 1  Pavement Condition Data runs the PESMOD.PAS and PESUP.PRG programs which display their own screens and are shown in Section 2 of this chapter.

Choice 2  Inventory Data runs the ENTRLAYR.PRG program if ADD is desired or CHNGLAYR.PRG if CHANGE is desired. These programs have their own screens which are displayed in Section 3 of this chapter.

Choice 3  Traffic Data runs the TRAFUPD.PRG program which has its own screens and are shown in Section 4 of this chapter.

Choice 4  Tables displays screen 3.5 and executes the dBASE BROWSE command for the requested option.

TEXAS FLEXIBLE PAVEMENT DATABASE 3.5
EDIT & UPDATE
Tables

1 - County Name
2 - Material Type
3 - Type of Pavement
4 - District Temperature Constant
5 - Widening Flag
6 - Layer Description
7 - Functional Classification

OPTION ————

WARNING ! ! ! - These files should only be modified by an experienced dBASE user. To abandon process, press the Esc key. Otherwise, enter option.
PROGRAM SPECIFICATION

Program Name: EDITUPDT.PRG

Purpose: To update the data in the master files and tables.

Procedures/Edits:

Refer to the appropriate sub-section for each program.

1) Convert PES data from tape to comma delimited ASCII files using the program PESMOD.PAS.
2) Add PES data to the Monitoring Files using the program PESUP.PRG.
3) Add data to the Inventory Files using the program ENTRIAVR.PRG.
4) Change data in the Inventory Files using the program CHNGIAVR.PRG.
5) Update the Traffic File with data from the Roadway Inventory File using the program TRAFUPD.PRG.
6) Add or Change the data in the Table files using the program EDITUPDT.PRG.

Input/Output Files:

PES Data - refer to the PESMOD.PAS sub-section for PES Modification files and the PESUP.PRG sub-section for PES Update files.

Inventory Data - refer to the CHNGIAVR.PRG sub-section for change files and the ENTRIAVR.PRG sub-section for add files.

Traffic Data - Refer to the TRAFUPD.PRG sub-section for Traffic files.

Tables -
- County Table (CNYTBL.DBF)
- Material Type Table (MATLTTBL.DBF)
- Pavement Type Table (PAVETYPE.DBF)
- District Temperature (DISTTEMP.DBF)
- Widening Table (WIDENFLG.DBF)
- Layer Description (LAYERTBL.DBF)
- Functional Classification (FUNCLTBL.DBF)
Programs Called (See Program Flow Diagram):

PES
- PESMOD.PAS
- NOPESSID.PRG
- NOLOCN.PRG
- PESUP.PRG
- PES_MRM.PRG
- PES_SKID.PRG
- PES_SST.PRG
- PES_VISL.PRG

INVENTORY
- FILECHEK.PRG

Add
- ENTRILAYR.PRG
- ADDLAYR.PRG
- INV_BKUP.PRG
- INV_UPDT.PRG
- LOCNCHEK.PRG
- LAYRCHEK.PRG
- LAYTCHER.PRG
- GEOCHEK.PRG
- SURFCHC.EK.PRG
- SUBGCHC.PRG

Change
- CHNGLAYR.PRG
- CHNGBKUP.PRG
- CHEKLOCN.PRG
- CHEKLAYR.PRG
- CHEKLAYT.PRG
- CHEKGEOS.PRG
- CHEKSURF.PRG
- CHEKSUBG.PRG
- COPYLAYR.PRG

TRAFFIC
- TRAFUPD.PRG
- LOGITRAF.PRG
- NEWITRAF.PRG
- SITITLOG.PRG
- STITLOG.PAS

TABLES
- TABLFILE.PRG
PROGRAM LISTING

* 
* SUBSYSTEM: EDIT & UPDATE MAIN MENU 
* PROGRAM NAME: EDITUPDTPRG 06/03/88 
* MODIFIED ON: 10/27/88 
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION 
* TAMU/TTI 
* AUTHOR: TREVOR X. PEREIRA 
* PURPOSE: TO EDIT AND UPDATE DATA FOR THE FOLLOWING: 
* 
* - Monitoring data (PES) 
* - Inventory data 
* - Traffic data 
* - Create Graph file 
* - Create Model file 
* - Tables 
* 

* set parameters and initialize variables 
STORE 0 TO MSID_NO 
CLOSE DATABASES 
* SET PROCEDURE TO \PAVEDB\EDITUPDTP\EDITCOLL 
MREPEAT2 = .T. 
SET STAT OFF 
SET ECHO OFF 
SET TALK OFF 
SET CONFIRM ON 

DO WHILE MREPEAT2 
STORE " " TO EDITPICK 
* set up the EDIT & UPDATE menu screen and do the loop until EDITPICK is 1-4 
DO WHILE .NOT. (EDITPICK $ '1234!') 
@ 0, 0 CLEAR 
@ 4, 20 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE 3.0" 
@ 5, 29 SAY "Edit & Update" 
@ 10, 16 SAY "1 - Pavement Condition Data" 
@ 11, 16 SAY "2 - Inventory Data" 
@ 12, 16 SAY "3 - Traffic Data" 
@ 13, 16 SAY "4 - Tables" 
@ 16, 41 SAY "OPTION >>>=> " GET EDITPICK 
@ 3, 9 TO 18, 65 DOUBLE 
READ 
IF READKEY() = 12 
CLEAR 
CLOSE DATABASE 
SET PROC TO 
MREPEAT2 = .F. 
EXIT 
ENDIF 
ENDDO 
IF MREPEAT2
* according to the response received from the EDIT & UPDATE menu, the appropriate
* commands are executed
DO CASE
* if choice 1 is chosen, run the PES program to add data to the
* monitoring files
CASE EDITPICK = "1"
  CLEAR
  MRUN = ""
  @ 5, 5 SAY "This program is going to update the monitoring database"
  @ 6, 5 SAY " Files. It will take the approximately 20 HOURS to run."
  @ 8, 5 SAY "To run this program, you need to have the PES Data file"
  @ 9, 5 SAY " that is obtained from tape in the subdirectory"
  @ 10, 5 SAY " \PAVEDB\EDITUPDT\PES"
  @ 12, 5 SAY "The data file should be called PES87.DAT for year 1987."
  @ 13, 5 SAY " If the file is for year 1988, the data file should be"
  @ 14, 5 SAY " called PES88.DAT"
  @ 17, 5 SAY "Do you want to continue (Y/N) ? " GET MRUN
  READ
  IF MRUN = "Y"
    RUN \PAVEDB\EDITUPDT\PES\PESMOD
    DO \PAVEDB\EDITUPDT\PES\PESUP
    CLEA TYPE
  ENDIF
  RELEASE MRUN
CASE EDITPICK = "2"
  CLEAR
  DO \PAVEDB\EDITUPDT\FILECHEK
  GETREPLY = ""
  DO WHILE .NOT. (GETREPLY $ 'AC')
    @ 10, 15 SAY "Do you want to ADD data to the inventory files or"
    @ 11, 15 SAY " CHANGE the data in the inventory files."
    @ 14, 15 SAY " Enter 'A' to ADD, 'C' to CHANGE or Esc to exit"
    @ 16, 15 SAY " OPTION ==>>> " GET GETREPLY PICTURE "!"
    READ
    IF READKEY() = 12
      MCONTINUE = .F.
      EDITCOLL = ""
      GETREPLY = ""
      EXIT
    ENDIF
  ENDDO
  IF GETREPLY = "C"
    DO \PAVEDB\EDITUPDT\CHNGLAYR
  ENDIF
  IF GETREPLY = "A"
    DO \PAVEDB\EDITUPDT\ENTRLAYR
  ENDIF
  CLEAR
  SET PROCEDURE TO
* if choice 3 is chosen, run the pascal traffic program
CASE EDITPICK = "$3"
MYEAR = 0
* run the program to update traffic file
@ 15, 30 SAY "DATA YEAR (19XX) ===> " GET MYEAR PICT "$9999"
READ
* Add the data to the traffic file
IF FILE("\PAVEDB\TLOG.DAT")
   DO \PAVEDB\EDITUPDT\TRAFFIC\UPDATE
ELSE
   CLEAR
   @ 10,10 SAY " "
   ? " Cannot find TLOG.DAT file. Please check and place the"
   ? " file in subdirectory \PAVEDB "
   ? " "
   WAIT
ENDIF
CLEA TYPE
CLEAR

* if choice 4 is chosen, display the tables menu
CASE EDITPICK = "$4"
MOCONTINUE = .T.
EDITCOLL = ""
DO \PAVEDB\EDITUPDT\TABLFILE
DO WHILE MOCONTINUE
   CLEAR
   EDITCOLL = ""
   DO WHILE .NOT. (EDITCOLL $ '1234567')
   @ 3, 21 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE 3.5"
   @ 4, 31 SAY "EDIT & UPDATE"
   @ 5, 34 SAY "Tables"
   @ 8, 16 SAY "1 - County Name"
   @ 9, 16 SAY "2 - Material Type"
   @ 10, 16 SAY "3 - Type of Pavement"
   @ 11, 16 SAY "4 - District Temperature Constant"
   @ 12, 16 SAY "5 - Widening Flag"
   @ 13, 16 SAY "6 - Layer Description"
   @ 14, 16 SAY "7 - Functional Classification"
   @ 17, 42 SAY "OPTION ===> " GET EDITCOLL
   @ 2, 9 TO 18, 65 DOUBLE
* Give warning
   @ 19, 10 SAY "WARNING !!! - These files should only be modified
by"
   @ 20, 10 SAY " an experienced dBASE user. To abandon
process,"
   @ 21, 10 SAY " press the Esc key. Otherwise, enter option."
   CLEA TYPE
   READ

IF READKEY() = 12
   MOCONTINUE = .F.
CLEAR
EXIT
ENDIF
ENDDO
IF MCONTINUE
DO CASE
CASE EDITCOL whla 1
  USE \PAVEDB\FILES\CNITBL INDEX \PAVEDB\INDEXES\CTYTB1NO
  BROWSE
CASE EDITCOL = "2"
  USE \PAVEDB\FILES\MATITBL
  BROWSE
CASE EDITCOL = "3"
  USE \PAVEDB\FILES\PAVETYPE
  BROWSE
CASE EDITCOL = "4"
  USE \PAVEDB\FILES\DISSERT
  BROWSE
CASE EDITCOL = "5"
  USE \PAVEDB\FILES\WIDENFLG
  BROWSE
CASE EDITCOL = "6"
  USE \PAVEDB\FILES\LAYERTBL
  BROWSE
CASE EDITCOL = "7"
  USE \PAVEDB\FILES\FUNCLTBL
  BROWSE
ENDCASE
ENDIF
ENDDO
ENDDO
ENDDO

SET PROCEDURE TO
CLOSE DATABASES
CLEAR
RETURN
* * SUBSYSTEM: EDIT & UPDATE
* PROGRAM NAME: TABFILE.PRG 07/03/88
* CALLED FROM: EDITUPD.PRG
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TII
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: CHECKS TO SEE IF THE TABLE FILES ARE PRESENT
*
IF .NOT. FILE('PAVE\FILES\CNYTBL.DBF')
   ? "COUNTY TABLE FILE (CNYTBL.DBF) not found. Please Check . . ."
   WAIT
   RETURN TO MASTER
ENDIF
IF .NOT. FILE('PAVE\INDEXES\CNYTBLNO.NDX')
   ? "COUNTY TABLE INDEX (CNYTBLNO.NDX) not found. Please Check . . ."
   WAIT
   RETURN TO MASTER
ENDIF
IF .NOT. FILE('PAVE\FILES\DISTTEMP.DBF')
   ? "DISTRICT TEMPERATURE FILE (DISTTEMP.DBF) not found. Please Check . . ."
   WAIT
   RETURN TO MASTER
ENDIF
IF .NOT. FILE('PAVE\FILES\MATLTBL.DBF')
   ? "MATERIAL TABLE FILE (MATLTBL.DBF) not found. Please Check . . ."
   WAIT
   RETURN TO MASTER
ENDIF
IF .NOT. FILE('PAVE\FILES\PAVETYPE.DBF')
   ? "PAVEMENT TYPE FILE (PAVETYPE.DBF) not found. Please Check . . ."
   WAIT
   RETURN TO MASTER
ENDIF
IF .NOT. FILE('PAVE\FILES\WIDENFLG.DBF')
   ? "WIDENING FLAG FILE (WIDENFLG.DBF) not found. Please Check . . ."
   WAIT
   RETURN TO MASTER
ENDIF
IF .NOT. FILE('PAVE\FILES\LAYERTBL.DBF')
   ? "LAYER TABLE FILE (LAYERTBL.DBF) not found. Please Check . . ."
   WAIT
   RETURN TO MASTER
ENDIF
IF .NOT. FILE('PAVE\FILES\FUNCLTBL.DBF')
   ? "FUNCTIONAL CLASSIFICATION TABLE FILE (FUNCLTBL.DBF) not found. "
   ? "Please Check . . ."
WAIT
RETURN TO MASTER
ENDIF
RETURN
Section 2: Monitoring Data Update
PROGRAM NARRATIVE

This program is for use with a 6250 BPI Cipher GRC Cache Tape tape drive systems manufactured by Overland Data Incorporated (ODI). It utilize the ODI DEPOT2 program to copy a tape file to a disk file. The DEPOT2 program is documented in Chapter 4 of the ODI manual. PESMOD.PAS uses the disk file as an input file. The diskfile (PES.DAT) is stored in the subdirectory \PAVEDB.
PROGRAM SPECIFICATION

Program Name: READPES.BAT

Purpose: To copy the PES data from a tape to a disk file using the ODI DEPOT2 program.

Input File: Original PES Tape File obtained from the Texas State Department of Highways and Public Transportation. The user can request only those records flagged for the Texas Flexible Pavement Database; thereby, decreasing the amount of disk space needed and decreasing processing time.

Output File: PES.DAT - Original PES Disk File

Programs Called:

   DEPOT2
   READPES.CMP - command file for DEPOT2
rem readpes.bat issues the depot2 command which calls
rem the cmd file to read the original pes tape obtained from
rem SDHPT
DEPOT2 /c READPES.CMD
; READPES.CMD, a command file for the ODI program DEPOT2.
; Use as: DEPOT2 /c READPES.CMD
; Instructions on making out this command file are in Chapter 4 of the ODI manual pages 4-26 to 4-28.
; Change D:\PAVEDB\PES.DAT to the name of your output disk file,
; if needed.
; Change "/r nn" record length if needed.
; Change "/s nnnn" blocksize if needed.
; "/m 1" skips past the tape label.
; "/tvm" translates from EBCDIC, Verbose explanations, reads till End-
; Of-File mark at the end of the data.
/n D:\PAVEDB\PES.DAT /r 1389 /s 1389 /m 1 /tvm
PES Modification Program Narrative

The Pavement Evaluation System (PES) record contains monitoring information for several lanes within a segment. The PES Data Modification program, which is written in TURBO Pascal splits an original PES record into separate monitoring files which contain a separate record for each visual evaluation lane. The original data comes from the PES file which is copied from a tape using READPES.BAT. The output of this program is six comma-delimited files containing different information (general, visual, maysmeter, skid, structural strength index and score information). The key fields are the same for all the output files. The output data is used to update the LOCATION, VISUAL, SI, SKID AND FALLWGT dBASE files in the Texas Flexible Pavement System. The general program flow is shown in Figure 11. The programs and comma delimited files are stored in the subdirectory \PAVEDB\EDITUPDT\PES.
PES Modification Program

Program Specification

Program Name: PESMOD.PAS (written in Turbo PASCAL)

Purpose: To split the original PES record into separate records for each Visual Evaluation Lane.

Input File: Original PES File (PES.DAT). The PES Tape File is copied to the PES.DAT disk file using a stand alone routine.

Procedures/Edit: 1. See the Procedure page.
2. See the Original PES File Record Layout and the Modified PES File Record Layouts.

Output Files: Modified PES Files - comma delimited files.
- general information files (PESGEN.DLM)
- visual information file (PESVIS.DLM)
- maysmeter information file (PESMMR.DLM)
- skid information file (PESSKD.DLM)
- structural strength index information file (PESSSI.DLM)
- score information file (PESSCR.DLM)

Report: Missing Data - See the Missing Data Report Layout.

Note: Following the Procedure page, the record layouts are provided in the order they are mentioned above with the Original PES File record layout being first.
Program Specification

PROCEDURE

The original PES record contains general information, visual rating, skidometer, skid, SSI, and pavement scoring information for all roadways within a segment. The PESMOD.PAS program reads the PES file and creates six files each of which contain a separate record for each roadway within the segment. A separate file is created for general information, skidometer information, visual rating information, skid information, SSI information, and for pavement scores, exclusively. Identifying information for the roadway (also called laneset) is the same in all six files.

The program creates a separate record in each output file for each lane found under original Pes record Item 18 - Visual Evaluation. The lanes for the visual evaluation are in columns 80, 112, 144, 176, 208, and 240 of the original PES record. For the lane set (i.e. R-V, L-P, A-C, etc.), which includes the visual lane being processed, the program selects the appropriate MRM, Skid, SSI, and Score data. For the applicable lane set, the program identifies the high and low MRM reading, calculates the average MRM reading and the standard deviation for the MRM, and stores the total number of MRM readings (maximum of 15). Also it provides the high, the low, the mean, the standard deviation, and the count of observations for the skid data for the appropriate lane set. The program also determines if there is visual evaluation data without corresponding MRM, SSI, or Score data and vice versa. Records which have missing data are reported. Only records with an 'A' in the Mandatory Sample Selection Flag field (Item 25A) and an 'E', 'F', 'G', 'H', 'I', 'N', 'O', 'P', 'Q' or 'R' in the Other Sample Selection Flag field (Item 25B) processed.
<table>
<thead>
<tr>
<th>ITEM NO.</th>
<th>FROM THRU</th>
<th>SIZE</th>
<th>DEC. FIELD</th>
<th>ITEM NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01-02</td>
<td>2</td>
<td>N</td>
<td>DISTRICT NO.</td>
<td>+ VALUES 01-21 &amp; 23-25.</td>
</tr>
<tr>
<td>2</td>
<td>03-04</td>
<td>2</td>
<td>N</td>
<td>MAINTENANCE SECTION NO.</td>
<td>+ ALIAS FOREMAN NUMBER.</td>
</tr>
<tr>
<td>3</td>
<td>05-07</td>
<td>3</td>
<td>N</td>
<td>COUNTY NO.</td>
<td>+ VALUES 001-254.</td>
</tr>
<tr>
<td>.B</td>
<td>10-13</td>
<td>4</td>
<td>N</td>
<td>NUMBER</td>
<td>+ EXCEPT. OSR=SHOORS, NASA1=SHNASA.</td>
</tr>
<tr>
<td>.C</td>
<td>14</td>
<td>1</td>
<td>A</td>
<td>SUFIX</td>
<td>+ VALUES S, L, A, N, E, W, BLANK. FOR SYS=PR, BLANK AND A, B, D, Z.</td>
</tr>
<tr>
<td>5.A</td>
<td>15-17</td>
<td>3</td>
<td>N</td>
<td>SEGMENT BEGIN - MILEPOST NO.</td>
<td>+ FOR NON-IH, EVEN NOS. (0, 2, 4, ...) ONLY. FOR IH, ODD &amp; EVEN.</td>
</tr>
<tr>
<td>.B</td>
<td>18</td>
<td>1</td>
<td>A</td>
<td>DISPLACEMENT SIGN</td>
<td>+ VALUES + AND -. ALWAYS + WHEN DISPLACEMENT VALUE = 0.0.</td>
</tr>
<tr>
<td>.C</td>
<td>19-20</td>
<td>2</td>
<td>N</td>
<td>DISPLACEMENT VALUE</td>
<td>+ ALIAS DISTANCE=FROM-POST. VALUES 0.0-3.0.</td>
</tr>
<tr>
<td>6.A</td>
<td>21-23</td>
<td>3</td>
<td>N</td>
<td>SEGMENT END - MILEPOST NO.</td>
<td>+ SEE ITEM 5 DESCRIPTION.</td>
</tr>
<tr>
<td>.B</td>
<td>24</td>
<td>1</td>
<td>A</td>
<td>DISPLACEMENT SIGN</td>
<td>+ ESPECIALLY ON IH, SEGMENTATION CAN REMOVE POSTS FROM VIEW. (IH SEG. FROM POST 110+0.0 TO 112+0.2 COVERS-UP POST 111).</td>
</tr>
<tr>
<td>.C</td>
<td>25-26</td>
<td>2</td>
<td>N</td>
<td>DISTANCE FROM BEGINNING OF ROUTE IN THIS COUNTY TO THE SEGMENT BEGIN OR END (ITEMS 5 OR 6) POSITION RESPECTIVELY, CENTERLINE MILEAGE FROM SEG. BEGIN TO END. VALUES 0.1-3.0.</td>
<td></td>
</tr>
<tr>
<td>7.A.1</td>
<td>27-29</td>
<td>3</td>
<td>A/N</td>
<td>INTERMEDIATE POST - 1ST POST NO.</td>
<td>+ DATE THIS RECORD (SEGMENT) LAST CHANGED. FORMAT = YYYYMMDD.</td>
</tr>
<tr>
<td>.2</td>
<td>30-32</td>
<td>3</td>
<td>A/N</td>
<td>2ND POST NO.</td>
<td>+ FOR PES RELEASE 1.0, ALWAYS 000.</td>
</tr>
<tr>
<td>.B.1</td>
<td>33-34</td>
<td>2</td>
<td>A/N</td>
<td>DISTANCE TO 1ST</td>
<td>+ ALL LANES. HIGHEST NO. IN SEGMENT.</td>
</tr>
<tr>
<td>.2</td>
<td>35-36</td>
<td>2</td>
<td>A/N</td>
<td>DISTANCE TO 2ND</td>
<td>+ ALL LANES ONE DIRECTION IN THOUSANDS. HIGHEST NO. IN SEG.</td>
</tr>
<tr>
<td>.B</td>
<td>41-44</td>
<td>1</td>
<td>N</td>
<td>END VALUE</td>
<td>+ MAIN LANES ONLY, HIGHEST NO. WITHIN SEGMENT.</td>
</tr>
<tr>
<td>9</td>
<td>45-46</td>
<td>2</td>
<td>N</td>
<td>SEGMENT LENGTH</td>
<td>+ FROM VISUAL EVALUATION, HIGHEST LEGAL SPEED. DEFAULT = 55.</td>
</tr>
<tr>
<td>10</td>
<td>47-52</td>
<td>6</td>
<td>N</td>
<td>LAST UPDATE DATE</td>
<td>+ USED IN SEGMENT LIMIT DETERMINATION ONLY. (SEE FTN 2).</td>
</tr>
<tr>
<td>11</td>
<td>53-61</td>
<td>9</td>
<td>2</td>
<td>SEGMENT MAINTENANCE COST</td>
<td>+ (SEE FTN 3). BLANK WHEN NO EVALUATION DATA PRESENT.</td>
</tr>
<tr>
<td>12</td>
<td>62-67</td>
<td>6</td>
<td>N</td>
<td>AVERAGE DAILY TRAFFIC</td>
<td>+ FORMAT = YYYYMMDD.</td>
</tr>
<tr>
<td>13</td>
<td>68-72</td>
<td>5</td>
<td>N</td>
<td>18-KIP EQUIVALENCY (.000)</td>
<td>+ (SEE PES VIS. EVAL. FORM 1624, 1625 OR 1626)</td>
</tr>
<tr>
<td>14</td>
<td>73</td>
<td>1</td>
<td>N</td>
<td>FUNCTIONAL CLASSIFICATION</td>
<td>+ VALUES 01-10. (SEE FTN 4).</td>
</tr>
<tr>
<td>15</td>
<td>74-75</td>
<td>2</td>
<td>N</td>
<td>NUMBER OF LANES</td>
<td>+ (FLEX=000,100,010,001/CRC=000 TO 200).</td>
</tr>
<tr>
<td>16</td>
<td>76-77</td>
<td>2</td>
<td>N</td>
<td>SPEED LIMIT</td>
<td>+ (FLEX=000,100,010,001/JCP=010 TO 100).</td>
</tr>
<tr>
<td>17</td>
<td>78-79</td>
<td>2</td>
<td>N</td>
<td>RIS SURFACE TYPE</td>
<td>+ (FLEX=000,100,010,001/JCP=000 TO 999).</td>
</tr>
<tr>
<td>18.A.1</td>
<td>80</td>
<td>1</td>
<td>A</td>
<td>VIS EVAL.(1ST) - LANE</td>
<td>+ (FLEX=000,100,010,001/JCP=000 TO 999).</td>
</tr>
<tr>
<td>.2</td>
<td>81-86</td>
<td>6</td>
<td>N</td>
<td>DATE EVALUATED</td>
<td>+ (FLEX=000,100,010,001/CRC=000 TO 200).</td>
</tr>
<tr>
<td>.3</td>
<td>87-98</td>
<td>2</td>
<td>N</td>
<td>COMMENT</td>
<td>+ (FLEX=000,100,010,001/JCP=010 TO 100).</td>
</tr>
<tr>
<td>.4</td>
<td>99-102</td>
<td>2</td>
<td>N</td>
<td>PAVEMENT TYPE</td>
<td>+ (FLEX=000,100,010,001/JCP=000 TO 999).</td>
</tr>
<tr>
<td>.5</td>
<td>103-105</td>
<td>3</td>
<td>N</td>
<td>FAIL-ACP-B</td>
<td>+ (FLEX=000,100,010,001,CRC=000 TO 200/JCP=000 TO 999).</td>
</tr>
<tr>
<td>.6</td>
<td>106-108</td>
<td>3</td>
<td>N</td>
<td>RUT-SPALL-FAILINT</td>
<td>+ (FLEX=000,100,010,001,CRC=000 TO 200/JCP=000 TO 999).</td>
</tr>
<tr>
<td>.7</td>
<td>109-111</td>
<td>3</td>
<td>N</td>
<td>ALLG-CRCKSPC-LNGCRCK</td>
<td>+ (FLEX=000,100,010,001,CRC=000 TO 200/JCP=000 TO 999).</td>
</tr>
<tr>
<td>.B</td>
<td>112-143</td>
<td>32</td>
<td>A/N</td>
<td>(2ND) - ALL ITEMS</td>
<td>+ (FLEX=000,100,010,001,CRC=000 TO 200/JCP=000 TO 999).</td>
</tr>
<tr>
<td>.C</td>
<td>144-175</td>
<td>32</td>
<td>A/N</td>
<td>(3RD) - ALL ITEMS</td>
<td>+ (FLEX=000,100,010,001,CRC=000 TO 200/JCP=000 TO 999).</td>
</tr>
<tr>
<td>.D</td>
<td>176-207</td>
<td>32</td>
<td>A/N</td>
<td>(4TH) - ALL ITEMS</td>
<td>+ (FLEX=000,100,010,001,CRC=000 TO 200/JCP=000 TO 999).</td>
</tr>
<tr>
<td>.E</td>
<td>208-239</td>
<td>32</td>
<td>A/N</td>
<td>(5TH) - ALL ITEMS</td>
<td>+ (FLEX=000,100,010,001,CRC=000 TO 200/JCP=000 TO 999).</td>
</tr>
<tr>
<td>.F</td>
<td>240-271</td>
<td>32</td>
<td>A/N</td>
<td>(6TH) - ALL ITEMS</td>
<td>+ (FLEX=000,100,010,001,CRC=000 TO 200/JCP=000 TO 999).</td>
</tr>
<tr>
<td>ITEM NO.</td>
<td>FROM</td>
<td>THRU</td>
<td>SIZE</td>
<td>POS.</td>
<td>TYPE</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>19.A</td>
<td>272-277</td>
<td>6</td>
<td>N</td>
<td>MRM GENERAL - LAST UPDATE DATE</td>
<td>+ MAIN LANE ONLY. FORMAT = YVMDD.</td>
</tr>
<tr>
<td>.B</td>
<td>278-279</td>
<td>2</td>
<td>1</td>
<td>- AVERAGE SI VALUE</td>
<td>+ MAIN LANE ONLY. 0.0 WHEN NONE TO AVERAGE. VALUES 0.1-5.0.</td>
</tr>
<tr>
<td>.B</td>
<td>283-324</td>
<td>42</td>
<td>2</td>
<td>- SI VALUE</td>
<td>+ MAIN LANE WITH POSTS. 0.0 WHEN NO SI. VALUES 0.1-5.0.</td>
</tr>
<tr>
<td>21.A</td>
<td>460-465</td>
<td>6</td>
<td>6</td>
<td>SKID GENERAL - LAST UPDATE DATE</td>
<td>+ MAIN LANE ONLY. FORMAT = YVMDD. (SKID INFO IS OPTIONAL)</td>
</tr>
<tr>
<td>.B</td>
<td>466-467</td>
<td>2</td>
<td>2</td>
<td>- AVERAGE SN VALUE</td>
<td>+ MAIN LANE ONLY. ZERO WHEN NONE TO AVERAGE.</td>
</tr>
<tr>
<td>22.A.1.A</td>
<td>468</td>
<td>469-470</td>
<td>2</td>
<td>SKID - LANES R-V - CELL 1 - LANE</td>
<td>+ MAIN LANE WITH POSTS. BLANK WHEN NO SN. (SEE FTNOTE 5).</td>
</tr>
<tr>
<td>.B</td>
<td>471-512</td>
<td>42</td>
<td>- ASN</td>
<td>- CELLS 2-15 - ALL</td>
<td>+ MAIN LANE WITH POSTS. ZERO WHEN NO SN.</td>
</tr>
<tr>
<td>23</td>
<td>648-650</td>
<td>3</td>
<td>3</td>
<td>SURFACE WIDTH</td>
<td>+ MAIN LANE ONLY. MEASURED IN FEET.</td>
</tr>
<tr>
<td>24</td>
<td>651</td>
<td>2</td>
<td>1</td>
<td>HIGHWAY DESIGN TYPE</td>
<td>+ TOTAL 1-9. HIGHEST VALUE IN SEG. USED. (SEE FTNOTE 6).</td>
</tr>
<tr>
<td>25.A</td>
<td>652</td>
<td>1</td>
<td>1</td>
<td>SAMPLE SELECTION FLAG - MANDATORY</td>
<td>+ WHEN SET (NONBLANK), INDICATES SEGMENT WHICH MUST BE TESTED.</td>
</tr>
<tr>
<td>.B</td>
<td>653</td>
<td>1</td>
<td>1</td>
<td>- OTHER</td>
<td>+ WHEN SET (NONBLANK), INDICATES SOME SPECIAL-PURPOSE SEGMENT.</td>
</tr>
<tr>
<td>.B</td>
<td>660-662</td>
<td>3</td>
<td>3</td>
<td>- Tem</td>
<td>+ ZERO WHEN NO SSI DATA.</td>
</tr>
<tr>
<td>.C</td>
<td>663-665</td>
<td>3</td>
<td>3</td>
<td>- TEMP</td>
<td>+ ZERO WHEN NO SSI DATA.</td>
</tr>
<tr>
<td>.B</td>
<td>666-677</td>
<td>12</td>
<td>12</td>
<td>- Lanes L-P - Cells 1-3 - All</td>
<td>+ MAIN LANE AGAINST POSTS. (SEE 26.A.1 FORMAT).</td>
</tr>
<tr>
<td>27.A.1.A</td>
<td>702</td>
<td>B</td>
<td>703-706</td>
<td>SSI - LANES R-V - CELL 1 - LANE</td>
<td>+ MAIN LNS WITH POSTS. BLANK WHEN NO SSI DATA. (SEE FTNOTE 8).</td>
</tr>
<tr>
<td>.B</td>
<td>707-710</td>
<td>4</td>
<td>4</td>
<td>- GPHN1</td>
<td>+ MAIN LANE WITH POSTS. ZERO WHEN NO SSI DATA.</td>
</tr>
<tr>
<td>.C</td>
<td>711-714</td>
<td>4</td>
<td>4</td>
<td>- GPHN2</td>
<td>+ MAIN LANE WITH POSTS. ZERO WHEN NO SSI DATA.</td>
</tr>
<tr>
<td>.D</td>
<td>715-718</td>
<td>4</td>
<td>4</td>
<td>- GPHN3</td>
<td>+ MAIN LANE WITH POSTS. ZERO WHEN NO SSI DATA.</td>
</tr>
<tr>
<td>.E</td>
<td>719-722</td>
<td>4</td>
<td>4</td>
<td>- GPHN4</td>
<td>+ MAIN LANE WITH POSTS. ZERO WHEN NO SSI DATA.</td>
</tr>
<tr>
<td>.F</td>
<td>723-726</td>
<td>4</td>
<td>4</td>
<td>- GPHN5</td>
<td>+ MAIN LANE WITH POSTS. ZERO WHEN NO SSI DATA.</td>
</tr>
<tr>
<td>.G</td>
<td>727-730</td>
<td>4</td>
<td>4</td>
<td>- GPHN6</td>
<td>+ MAIN LANE WITH POSTS. ZERO WHEN NO SSI DATA.</td>
</tr>
<tr>
<td>.H</td>
<td>731-846</td>
<td>116</td>
<td>4</td>
<td>- GPHN7</td>
<td>+ MAIN LANE WITH POSTS. ZERO WHEN NO SSI DATA.</td>
</tr>
<tr>
<td>.B</td>
<td>847-991</td>
<td>145</td>
<td>4</td>
<td>- CELLS L-P - CELLS 1-5 - ALL</td>
<td>+ MAIN LNS W/ POSTS. OTHER CELLS. (4 OCCURS OF 27.A.1 FORMAT).</td>
</tr>
<tr>
<td>.C</td>
<td>992-1136</td>
<td>145</td>
<td>4</td>
<td>- CELLS A-C - CELLS 1-5 - ALL</td>
<td>+ MAIN LANE AGAINST POSTS. (SEE 27.A.1 FORMAT).</td>
</tr>
<tr>
<td>.D</td>
<td>1137-1281</td>
<td>145</td>
<td>4</td>
<td>- CELLS X-Z - CELLS 1-5 - ALL</td>
<td>+ FRONTAGE ROAD LANE AGAINST POSTS. (SEE 27.A.1 FORMAT).</td>
</tr>
<tr>
<td>28.A.1</td>
<td>1282-1284</td>
<td>3</td>
<td>3</td>
<td>SCORES - ROWAY R-V - IDENTIFIER</td>
<td>+ MAIN ROWAY WITH POSTS. BLANK WHEN NO SCORE. (SEE FTNOTE 7).</td>
</tr>
<tr>
<td>.B</td>
<td>1285-1287</td>
<td>3</td>
<td>3</td>
<td>- MRM UTILITY</td>
<td>+ MAIN ROWAY WITH POSTS. 0.00 WHEN UNABLE TO CALCULATE.</td>
</tr>
<tr>
<td>.C</td>
<td>1288-1290</td>
<td>3</td>
<td>3</td>
<td>- SKID UTILITY</td>
<td>+ FRONTAGE ROADWAY LANE POSTS. FOR RES PES RELEASE 1.0, NOT AVAILABLE.</td>
</tr>
<tr>
<td>.D</td>
<td>1291-1293</td>
<td>3</td>
<td>3</td>
<td>- MAINT. COST</td>
<td>+ MAIN ROWAY WITH POSTS. FOR PES RELEASE 1.0, NOT AVAILABLE.</td>
</tr>
<tr>
<td>.B</td>
<td>1294-1296</td>
<td>3</td>
<td>3</td>
<td>- UNWEIGHTED VTS.</td>
<td>+ MAIN ROWAY WITH POSTS. 0.00 WHEN NO CALCULATION.</td>
</tr>
<tr>
<td>.C</td>
<td>1297-1299</td>
<td>3</td>
<td>3</td>
<td>- ADJUSTED VTS</td>
<td>+ MAIN ROWAY WITH POSTS. 0.00 WHEN NO CALCULATION.</td>
</tr>
<tr>
<td>.D</td>
<td>1300-1302</td>
<td>3</td>
<td>3</td>
<td>- WEIGHTED VTS</td>
<td>+ MAIN ROWAY WITH POSTS. 0.00 WHEN NO CALCULATION.</td>
</tr>
<tr>
<td>.B</td>
<td>1303-1305</td>
<td>3</td>
<td>3</td>
<td>- PVT. SCORE</td>
<td>+ MAIN ROWAY WITH POSTS. 0.00 WHEN NO CALCULATION.</td>
</tr>
</tbody>
</table>
GENERAL INFO.

- SIZE INDICATES FULL LENGTH OF ITEM DESCRIBED. NUMERICS ARE EXTERNAL DECIMAL.
- DECIMAL POSITIONS (DEC. POS.) ARE INCLUDED IN SIZE. DECIMAL POINT IS IMPLIED.
- FIELD TYPE A IS ALPHANUMERIC (BLANK, +,-,A,B,...,Z). FIELD TYPE N IS EXTERNAL DECIMAL NUMERIC (0-9).
- SOLE USER CONTROL OVER FILE DUMP IS THRU MANIPULATION OF REPORT HEADER (MANDATORY AND OTHER SAMPLE SELECTION FLAGS). OTHERWISE, ENTIRE DISTRICT OR ENTIRE STATE IS DUMPED.
- FOR PES RELEASE 1.0, NO UTILITY SCORES ARE COMPUTED UNLESS ALL INFO. REQUIRED FOR CALCULATION OF PAVEMENT SCORE IS AVAILABLE.
- ALL KEY INFO. (DISTRICT, MILEPOSTS, ...) AND ALL RIS INFO. (ADT, FUNC. CLASS, ...) ARE PRESENTED AS CAPTURED AT CONVERSION TIME, NORMALLY AROUND THE BEGINNING OF EACH FISCAL YEAR.
- FOR PES RELEASE 1.0, SKID INFORMATION IS WHOLLY OPTIONAL. INCLUSION / EXCLUSION WILL NOT ALTER CALCULATED UTILITIES AND SCORES. NOTE THAT SKID UTILITY IS NOT PRESENTLY AVAILABLE.
- QUESTIONS SHOULD BE DIRECTED TO FILE D-18, TRAFFIC SAFETY AND MAINTENANCE OPERATIONS DIVISION.

FOOTNOTE 1 - A CHANGE IN FUNCTIONAL CLASS IS ONE CAUSE OF SEGMENTATION. DECODIFICATION IS FROM 0-10 PLATE 25 (REV. 5/1/84).
1. INTERSTATE
2. OTHER URBAN FREEWAY AND EXPRESSWAY
3. RURAL PRINCIPAL ARTERIAL
   - OR
4. MINOR ARTERIAL ROAD OR STREET
5. RURAL MAJOR COLLECTOR
   - OR
6. URBAN COLLECTOR STREET
7. LOCAL ROAD OR STREET

FOOTNOTE 2 - A CHANGE IN SURFACE TYPE GROUP IS ANOTHER CAUSE OF SEGMENTATION. CODE 61 AND 62 HAVE THEIR OWN GROUPS WHILE ALL OTHER CODES FORM A THIRD GROUP. WITHIN THAT 3RD GROUP, THE HIGHEST NON-71 AND -81 CODE IS USED. THE READER SHOULD UNDERSTAND THAT THIS DATA ITEM IS USED DURING INITIAL SEGMENTATION (CONVERSION) ONLY AND IS INCORPORATED IN THE PES RECORD FRAMEWORK SOLELY FOR DOCUMENTATION PURPOSES. PAVEMENT TYPE, FOUND UNDER VISUAL EVALUATION HEADING, IS THE VITAL PAVEMENT ITEM USED IN UTILITY SCORE CALCULATIONS. SURFACE TYPE AGAIN IS DRAWN FROM 0-10 PLATE 25.
51. BITUMINOUS SURFACE - TREATED (UNDER 1 IN.)
52. MIXED BIT. SURFACE (BASE & SURF. UNDER 7 IN.)
53. MIXED BIT. SURFACE (BASE & SURF. 7 IN. OR MORE)
54. BIT. PENETRATION (BASE & SURF. 7 IN. OR MORE)
55. BIT. CONCRETE OR SHOE ASPHALT (1 IN. OR MORE)
56. BIT. PENETRATION (BASE & SURF. UNDER 7 IN.)
57. BRICK ROAD
58. BLOCK ROAD

FOOTNOTE 3 - LANE DESIGNATIONS FOLLOW. DESCRIPTION IS FROM OUTSIDE LANE TOWARD CENTERLINE OR MEDIAN. THUS, A TWO-LANE DESIGN TYPE 1 FACILITY WILL SHOW LANES R & L ONLY. A DESIGN TYPE 9 FACILITY WILL HAVE MANY LANES (R,S,...,L,M,...,X,A,...) BUT CANNOT HAVE LANE T, FOR INSTANCE, WITHOUT AN ADJACENT LANE S (TO THE RIGHT) WHICH IS, IN TURN, ADJACENT TO LANE R.

<table>
<thead>
<tr>
<th>FRONTAGE RDS.</th>
<th>MAIN LANES</th>
<th>MAIN LANES</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ + + + + + + +</td>
<td>+ + + + + + + +</td>
<td>+ + + + + + + +</td>
</tr>
<tr>
<td>X + Y + Z</td>
<td>L + M + N + O + P</td>
<td>V + U + T + S + R</td>
</tr>
<tr>
<td>+ + + + + + + +</td>
<td>+ + + + + + + +</td>
<td>+ + + + + + + +</td>
</tr>
</tbody>
</table>

FOOTNOTE 4 - VISUALLY EVALUATED (OBTAINED) PAVEMENT TYPE DECODIFICATION HAS BEEN PULLED FROM FORM 1626.
1. CONTINUOUSLY REINFORCED CONCRETE
2. JOINTED REINFORCED CONCRETE
3. JOINTED PLAIN CONCRETE
4. THICK ASPHALTIC CONCRETE (OVER 5.5 IN.)
5. INTERMEDIATE ASPHALTIC CONCRETE (2.5-5.5 IN.)
6. THIN SURFACED FLEXIBLE BASE (UNDER 2.5 IN.)
7. COMPOSITE ASPHALT SURFACED CONCRETE
8. OVERLaid AND/OR WIDENED OLD CONCRETE
9. OVERLaid AND/OR WIDENED OLD FLEXIBLE
10. THIN SURFACED FLEX. BASE (SURF. TRT. - SEAL COAT)

FOOTNOTE 5 - 15 CELLS FOR EACH OF 4 SETS OF LANES (EACH LANE SET REPRESENTS A POTENTIAL ROADWAY) ARE RESERVED. LANE SETS WHICH, DUE TO HIGHWAY DESIGN, DO NOT EXIST, ARE VOID OF DATA. EACH CELL CONSTITUTES 0.2 MILES OF A SEGMENT (THUS, MAXIMUM SEGMENT LENGTH OF 3.0 MILES) WITH THE EXCEPTION THAT THE LAST CELL OF AN ODD MILEAGE SEGMENT (IE, LENGTH = 1.9 MILES) WILL BE UNDERSTOOD TO CONSTITUTE ONLY 0.1 MILE. CELLS OUTSIDE THE RANGE NECESSARY TO EXHAUST THE SEGMENT LENGTH ARE VOID OF DATA, REGARDLESS OF DIRECTION OF TRAVEL (WITH OR AGAINST POSTS). CELLS WITHIN A 15-MEMBER UNIT TIE TOGETHER END-TO-END FROM LOW NUMBERED CELL TO HIGH NUMBERED CELL (1 TO 15) AND COVER THE MILEAGE TRAVERSED FROM SEGMENT BEGIN POST & DISPLACEMENT TO SEGMENT END POST & DISPLACEMENT. THUS, CELL 1 FOR EACH OF THE 4 LANE SETS CONSTITUTES THE FIRST 0.2 MILES OF A SEGMENT (ROW-LINE TO ROW-LINE).
FOOTNOTE 6 - HIGHWAY DESIGN CODES ARE PULLED FROM D-10 PLATE 25. NUMBER OF ROADWAYS HAS BEEN PARENTHETICALLY APPENDED.
1. TWO-WAY TRAFFIC (1 ROADWAY)
2. ONE-WAY TRAFFIC (1 ROADWAY)
3. BOULEVARD (2 ROADWAYS)
4. EXPRESSWAY - NO SERVICE ROADS (2 ROADWAYS)
5. EXPRESSWAY - ONE SERVICE ROAD (3 ROADWAYS)
6. EXPRESSWAY - TWO SERVICE ROADS (4 ROADWAYS)
7. FREEWAY - NO SERVICE ROADS (2 ROADWAYS)
8. FREEWAY - ONE SERVICE ROAD (3 ROADWAYS)
9. FREEWAY - TWO SERVICE ROADS (4 ROADWAYS)

FOOTNOTE 7 - AS MENTIONED IN FOOTNOTE 5, EACH LANE SET REPRESENTS A POTENTIAL ROADWAY. HOWEVER, THE FOLLOWING HANDLING RULES SHOULD BE KEPT IN MIND. UTILITY AND PAVEMENT SCORES MAY / MAY NOT BE CALCULATED FOR EXISTING ROADWAYS DEPENDING UPON THE AVAILABILITY OF REQUIRED DATA ITEMS. FOR PES RELEASE 1.0, EITHER ALL SCORES WILL BE CALCULATED OR NONE WILL BE FOUND.
1. FOR HIGHWAY DESIGN TYPES 1 & 2, ONLY 1 ROADWAY EXISTS. LANE SETS R-V AND L-P DATA ARE MERGED WITH RESULTING SCORE OUTPUT (IF COMPUTATIONS POSSIBLE) AS A ROADWAY IDENTIFIED BY R-L AND OCCUPIED BY THE ROADWAY R-V AREA.
2. FOR DESIGN TYPES 3, 4, AND 7, THE ROADWAY R-V AND L-P AREAS RECEIVE SCORE CALCULATIONS.
4. FOR DESIGN TYPES 6 AND 9, ALL ROADWAY AREAS ARE POTENTIAL RECEIVERS OF SCORE CALCULATIONS.

FOOTNOTE 8 -- 5 CELLS FOR EACH OF 4 SETS OF LINES (EACH LANE SET REPRESENTS A POTENTIAL ROADWAY) ARE RESERVED. LANE SETS WHICH, DUE TO HIGHWAY DESIGN, DO NOT EXIST, ARE VOID OF DATA. EACH CELL CONSTITUTES ONE-FIFTH OF A SEGMENT AND GEOPHONE READINGS FOR EACH OF THE 5 CELLS ARE MANDATORY.
### General Information File Record Layout

<table>
<thead>
<tr>
<th>Field</th>
<th>Size/ Type</th>
<th>Note</th>
<th>Item No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HWYDIST</td>
<td>2N</td>
<td></td>
<td>1</td>
<td>Highway District Number</td>
</tr>
<tr>
<td></td>
<td>2N</td>
<td></td>
<td>2</td>
<td>Maintenance Section Number</td>
</tr>
<tr>
<td>CntyNum</td>
<td>3N</td>
<td></td>
<td>3</td>
<td>County Number</td>
</tr>
<tr>
<td>HWYPREFIX</td>
<td>2C</td>
<td></td>
<td>4.A</td>
<td>Highway System</td>
</tr>
<tr>
<td>HWYNUM</td>
<td>4N(C)</td>
<td>2</td>
<td>4.B</td>
<td>Highway Number</td>
</tr>
<tr>
<td>HWYSUFFIX</td>
<td>1C</td>
<td></td>
<td>4.C</td>
<td>Highway Suffix</td>
</tr>
<tr>
<td>BEGMPST</td>
<td>3N</td>
<td></td>
<td>5.A</td>
<td>Segment Begin - Milepost No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.B</td>
<td>- Displacement Sign</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMPSTDIS</td>
<td>3.1N</td>
<td>1</td>
<td>5.C</td>
<td>(9,9) - Displacement Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laneset</td>
<td>3C</td>
<td>11</td>
<td>6.A</td>
<td>Segment End - Milepost No.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6.B</td>
<td>- Displacement Sign</td>
</tr>
<tr>
<td>EMPSTDIS</td>
<td>3.1N</td>
<td>1</td>
<td>6.C</td>
<td>(9,9) - Displacement Value</td>
</tr>
<tr>
<td>Intermediate Post - 1st Post</td>
<td>3C/N</td>
<td>2</td>
<td>7.A.1</td>
<td>Segment Length</td>
</tr>
<tr>
<td>- 2nd Post</td>
<td>3C/N</td>
<td>2</td>
<td>7.A.2</td>
<td>Last Update Date (YMMDD)</td>
</tr>
<tr>
<td>- Distance</td>
<td>3.1N</td>
<td>2</td>
<td>7.B.1</td>
<td>Segment Maintenance Cost</td>
</tr>
<tr>
<td>- Distance</td>
<td></td>
<td></td>
<td></td>
<td>Average Daily Traffic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18-Kip Equivalency</td>
</tr>
<tr>
<td>Route-Thru-County - Begin Value</td>
<td>5.1N</td>
<td>2</td>
<td>8.A</td>
<td>Segment Length</td>
</tr>
<tr>
<td>- End Value</td>
<td></td>
<td></td>
<td>8.B</td>
<td>Last Update Date (YMMDD)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>Segment Maintenance Cost</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>Average Daily Traffic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td>18-Kip Equivalency</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>Functional Classification</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13</td>
<td>Number of lanes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>14</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

* Key field
### General Information File Record Layout

<table>
<thead>
<tr>
<th>Size/Field</th>
<th>Type</th>
<th>Note</th>
<th>Item No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2N</td>
<td>16</td>
<td></td>
<td>16</td>
<td>Speed Limit</td>
</tr>
<tr>
<td>2N</td>
<td>17</td>
<td></td>
<td></td>
<td>RIS Surface Type</td>
</tr>
<tr>
<td>3N</td>
<td>23</td>
<td></td>
<td></td>
<td>Surface Width</td>
</tr>
<tr>
<td>1N</td>
<td>24</td>
<td></td>
<td></td>
<td>Highway Design Type</td>
</tr>
<tr>
<td>1C</td>
<td>6</td>
<td>25.A</td>
<td></td>
<td>Sample Selection Flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Mandatory</td>
</tr>
<tr>
<td>1C</td>
<td>6</td>
<td>25.B</td>
<td></td>
<td>- Other</td>
</tr>
</tbody>
</table>

* Key Field
### Visual Information File Record Layout

<table>
<thead>
<tr>
<th>Field</th>
<th>Size/Type</th>
<th>Note</th>
<th>PES Item No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>* HWYDIST</td>
<td>2N</td>
<td>1</td>
<td>1</td>
<td>Highway District Number</td>
</tr>
<tr>
<td></td>
<td>2N</td>
<td>2</td>
<td>2</td>
<td>Maintenance Section Number</td>
</tr>
<tr>
<td>* CNTYNUM</td>
<td>3N</td>
<td>3</td>
<td>3</td>
<td>County Number</td>
</tr>
<tr>
<td>* HWYPREFIX</td>
<td>2C</td>
<td>4.A</td>
<td>4</td>
<td>Highway System</td>
</tr>
<tr>
<td>* HWYNUM</td>
<td>4N(C)</td>
<td>4.B</td>
<td>4</td>
<td>Highway Number</td>
</tr>
<tr>
<td>* HWYSUFFIX</td>
<td>1C</td>
<td>4.C</td>
<td>4</td>
<td>Highway Suffix</td>
</tr>
<tr>
<td>* BEGMPST</td>
<td>3N</td>
<td>5.A</td>
<td>5</td>
<td>Segment Begin - Milepost No</td>
</tr>
<tr>
<td>* EDISSIGN</td>
<td>1C</td>
<td>5.B</td>
<td>5</td>
<td>Displacement Sign</td>
</tr>
<tr>
<td>* BMPSTDIS</td>
<td>3.1N</td>
<td>1</td>
<td>5.C</td>
<td>(9,9) - Displacement Value</td>
</tr>
<tr>
<td>*</td>
<td>3C</td>
<td>11</td>
<td></td>
<td>Laneset</td>
</tr>
<tr>
<td></td>
<td>1C</td>
<td>3</td>
<td>18.A-F.1</td>
<td>Visual Evaluation - Lane</td>
</tr>
<tr>
<td></td>
<td>6N</td>
<td>3</td>
<td>18.A-F.2</td>
<td>(YMMDD) - Date Evaluated</td>
</tr>
<tr>
<td></td>
<td>2N</td>
<td>3</td>
<td>18.A-F.3</td>
<td>Comment</td>
</tr>
<tr>
<td></td>
<td>2N</td>
<td>3</td>
<td>18.A-F.4</td>
<td>Pavement Type</td>
</tr>
<tr>
<td>FAILMILE</td>
<td>3N</td>
<td>3</td>
<td>18.A-F.5</td>
<td>Failures</td>
</tr>
<tr>
<td>TRANCRMD</td>
<td>3N</td>
<td>3</td>
<td>18.A-F.6</td>
<td>Transverse Cracks</td>
</tr>
<tr>
<td>LONGCRMD</td>
<td>3N</td>
<td>3</td>
<td>18.A-F.7</td>
<td>Longitudinal Cracks</td>
</tr>
<tr>
<td></td>
<td>3N</td>
<td>3</td>
<td>18.A-F.8</td>
<td>Rutting</td>
</tr>
<tr>
<td>ALLGCRMD</td>
<td>3N</td>
<td>3</td>
<td>18.A-F.9</td>
<td>Alligator Cracks</td>
</tr>
<tr>
<td>BLKCRKMD</td>
<td>3N</td>
<td>3</td>
<td>18.A-F.10</td>
<td>Block Cracks</td>
</tr>
<tr>
<td>PATCHFR</td>
<td>3N</td>
<td>3</td>
<td>18.A-F.11</td>
<td>Maintenance Patching</td>
</tr>
</tbody>
</table>

* Key field
<table>
<thead>
<tr>
<th>Field</th>
<th>Size/ Type</th>
<th>Note</th>
<th>PES Item No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>* HWYDIST</td>
<td>2N</td>
<td>1</td>
<td>1</td>
<td>Highway District Number</td>
</tr>
<tr>
<td></td>
<td>2N</td>
<td>2</td>
<td>2</td>
<td>Maintenance Section Number</td>
</tr>
<tr>
<td>* CNTYNUM</td>
<td>3N</td>
<td>3</td>
<td>3</td>
<td>County Number</td>
</tr>
<tr>
<td>* HWYNUM</td>
<td>4N(C)</td>
<td>2</td>
<td>4.C</td>
<td>Highway Number</td>
</tr>
<tr>
<td>* HWYSUFFIX</td>
<td>1C</td>
<td>5.A</td>
<td>5.B</td>
<td>Highway Suffix</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Segment Begin - Milepost No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Displacement Sign</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(9.9) - Displacement Value</td>
</tr>
<tr>
<td>* BMPSTDIS</td>
<td>3.1N</td>
<td>1</td>
<td>5.C</td>
<td>Laneset</td>
</tr>
<tr>
<td></td>
<td>3C</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6N</td>
<td>19.A</td>
<td></td>
<td>MRM General - Last Update (YMMDD) Date</td>
</tr>
<tr>
<td></td>
<td>3.1N</td>
<td>19.B</td>
<td></td>
<td>- Average SI Value</td>
</tr>
<tr>
<td></td>
<td>1C</td>
<td>20.A-D.1.A</td>
<td></td>
<td>MRM - Lane</td>
</tr>
<tr>
<td></td>
<td>3.1N</td>
<td>20.A-D.1.B</td>
<td></td>
<td>- Lane SI Value</td>
</tr>
<tr>
<td></td>
<td>56C&amp;N</td>
<td>20.A-D.2</td>
<td></td>
<td>- Lane &amp; SI Values (14 occurrences of 20.A.1)</td>
</tr>
<tr>
<td>SIMEAN</td>
<td>7.5N</td>
<td>12</td>
<td></td>
<td>MRM Mean for the Laneset</td>
</tr>
<tr>
<td>SISD</td>
<td>7.5N</td>
<td>12</td>
<td></td>
<td>MRM Std. Dev. for the Laneset</td>
</tr>
<tr>
<td>SIHI</td>
<td>3.1N</td>
<td>12</td>
<td></td>
<td>MRM High for the Laneset</td>
</tr>
<tr>
<td>SILOWVAL</td>
<td>3.1N</td>
<td>12</td>
<td></td>
<td>MRM Low for the Laneset</td>
</tr>
<tr>
<td>SICOUNT</td>
<td>3N</td>
<td>12</td>
<td></td>
<td>MRM Count of Obs. for the Laneset</td>
</tr>
</tbody>
</table>

* Key field
### PES Modification Program

#### Program Specification (continued)

**Skid Information File Record Layout**

<table>
<thead>
<tr>
<th>Field</th>
<th>Size/Type</th>
<th>Note</th>
<th>PES Item No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>* HWYDIST</td>
<td>2N</td>
<td>1</td>
<td>1</td>
<td>Highway District Number</td>
</tr>
<tr>
<td></td>
<td>2N</td>
<td>2</td>
<td></td>
<td>Maintenance Section Number</td>
</tr>
<tr>
<td>* CNTYNUM</td>
<td>3N</td>
<td>3</td>
<td></td>
<td>County Number</td>
</tr>
<tr>
<td>* HWYPREFIX</td>
<td>2C</td>
<td>4.A</td>
<td></td>
<td>Highway System</td>
</tr>
<tr>
<td>* HWYNUM</td>
<td>4N(C)</td>
<td>2</td>
<td>4.B</td>
<td>Highway Number</td>
</tr>
<tr>
<td>* HWYSUFFIX</td>
<td>1C</td>
<td>4.C</td>
<td></td>
<td>Highway Suffix</td>
</tr>
<tr>
<td>* BEGMPST</td>
<td>3N</td>
<td>5.A</td>
<td></td>
<td>Segment Begin - Milepost No</td>
</tr>
<tr>
<td>* BDISSIGN</td>
<td>1C</td>
<td>5.B</td>
<td></td>
<td>- Displacement Sign</td>
</tr>
<tr>
<td>* BMPSTDIS</td>
<td>3.1N</td>
<td>1</td>
<td>5.C</td>
<td>(9.9) - Displacement Value</td>
</tr>
<tr>
<td></td>
<td>3C</td>
<td>11</td>
<td></td>
<td>Laneset</td>
</tr>
<tr>
<td></td>
<td>6N</td>
<td>21.A</td>
<td></td>
<td>Skid General - Last Update (YMMDD) Date</td>
</tr>
<tr>
<td></td>
<td>2N</td>
<td>21.B</td>
<td></td>
<td>- Average SN Value</td>
</tr>
<tr>
<td></td>
<td>1C</td>
<td>5</td>
<td>22.A-D.1.A</td>
<td>Skid - Lane</td>
</tr>
<tr>
<td></td>
<td>2N</td>
<td>5</td>
<td>22.A-D.1.B</td>
<td>- Lane SN Values</td>
</tr>
<tr>
<td></td>
<td>42 C&amp;N</td>
<td>5</td>
<td>22.A-D.2</td>
<td>- Lane &amp; SN Value (14 occurrences of 22.A.1)</td>
</tr>
<tr>
<td>SKIDNUMN</td>
<td>2N</td>
<td>13</td>
<td></td>
<td>SN Mean for the Laneset</td>
</tr>
<tr>
<td>SNSD</td>
<td>2N</td>
<td>13</td>
<td></td>
<td>SN Std. Dev. for the Laneset</td>
</tr>
<tr>
<td>SKIDNUMH</td>
<td>2N</td>
<td>13</td>
<td></td>
<td>SN High for the Laneset</td>
</tr>
<tr>
<td>SKIDSDF</td>
<td>2N</td>
<td>13</td>
<td></td>
<td>SN Low for the Laneset</td>
</tr>
<tr>
<td>SNCOUNT</td>
<td>2N</td>
<td>13</td>
<td></td>
<td>SN Count of Obs. for the Laneset</td>
</tr>
</tbody>
</table>

* Key field
### PSS Modification Program
#### Program Specification (continued)

#### Structural Strength Index Information File Record Layout

<table>
<thead>
<tr>
<th>Field</th>
<th>Size/Type</th>
<th>Note</th>
<th>Item No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HWYDIST</td>
<td>2N</td>
<td></td>
<td>1</td>
<td>Highway District Number</td>
</tr>
<tr>
<td></td>
<td>2N</td>
<td></td>
<td>2</td>
<td>Maintenance Section Number</td>
</tr>
<tr>
<td>CNTYNUM</td>
<td>3N</td>
<td></td>
<td>3</td>
<td>County Number</td>
</tr>
<tr>
<td>HWYPREFIX</td>
<td>2C</td>
<td></td>
<td>4.A</td>
<td>Highway System</td>
</tr>
<tr>
<td>HWYNUM</td>
<td>4N(C)</td>
<td>2</td>
<td>4.B</td>
<td>Highway Number</td>
</tr>
<tr>
<td>HWYSUFFIX</td>
<td>1C</td>
<td></td>
<td>4.C</td>
<td>Highway Suffix</td>
</tr>
<tr>
<td>BEGMPSST</td>
<td>3N</td>
<td></td>
<td>5.A</td>
<td>Segment Begin - Milepost No</td>
</tr>
<tr>
<td>BDISSIGN</td>
<td>1C</td>
<td></td>
<td>5.B</td>
<td>- Displacement Sign</td>
</tr>
<tr>
<td>BMPSTDIS</td>
<td>3.1N</td>
<td>1</td>
<td>5.C</td>
<td>(9.9) - Displacement Value</td>
</tr>
</tbody>
</table>

|           |           |      | 3C       | 11                                               |
| RWSSIENV  | 6N        | 7    | 26.A.1-4.A | SSI Date (YVMMDD)                              |
| SISIGPH11  | 5.2N      | 8    | 27.A-D.1.A | SSI Lane Station 1                              |
| SISIGPH12  | 5.2N      | 8    | 27.A-D.1.C | Geophone Reading 1 Station 1                    |
| SISIGPH13  | 5.2N      | 8    | 27.A-D.1.D | Geophone Reading 2 Station 1                    |
| SISIGPH14  | 5.2N      | 8    | 27.A-D.1.E | Geophone Reading 3 Station 1                    |
| SISIGPH15  | 5.2N      | 8    | 27.A-D.1.F | Geophone Reading 4 Station 1                    |
| SISIGPH16  | 5.2N      | 8    | 27.A-D.1.G | Geophone Reading 5 Station 1                    |
| SISIGPH17  | 5.2N      | 8    | 27.A-D.1.H | Geophone Reading 6 Station 1                    |
| SISIGPH21  | 5.2N      | 8    | 27.A-D.2.A | SSI Lane Station 2                              |
| SISIGPH22  | 5.2N      | 8    | 27.A-D.2.B | Geophone Reading 1 Station 2                    |
| SISIGPH23  | 5.2N      | 8    | 27.A-D.2.C | Geophone Reading 2 Station 2                    |
| SISIGPH24  | 5.2N      | 8    | 27.A-D.2.D | Geophone Reading 3 Station 2                    |
| SISIGPH25  | 5.2N      | 8    | 27.A-D.2.E | Geophone Reading 4 Station 2                    |
| SISIGPH26  | 5.2N      | 8    | 27.A-D.2.F | Geophone Reading 5 Station 2                    |
| SISIGPH27  | 5.2N      | 8    | 27.A-D.2.G | Geophone Reading 6 Station 2                    |
| SISIGPH31  | 5.2N      | 8    | 27.A-D.3.A | SSI Lane Station 3                              |
| SISIGPH32  | 5.2N      | 8    | 27.A-D.3.B | Geophone Reading 1 Station 3                    |
| SISIGPH33  | 5.2N      | 8    | 27.A-D.3.C | Geophone Reading 2 Station 3                    |
| SISIGPH34  | 5.2N      | 8    | 27.A-D.3.D | Geophone Reading 3 Station 3                    |
| SISIGPH35  | 5.2N      | 8    | 27.A-D.3.E | Geophone Reading 4 Station 3                    |
| SISIGPH36  | 5.2N      | 8    | 27.A-D.3.F | Geophone Reading 5 Station 3                    |
|           |           |      |          |                                                  |

* Key Field

220
PES Modification Program
Program Specification (continued)

Structural Strength Index Information File Record Layout

<table>
<thead>
<tr>
<th>Field</th>
<th>Size/Type</th>
<th>Note</th>
<th>Item No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSIGPH37</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.3.H</td>
<td>Geophone Reading 7</td>
</tr>
<tr>
<td></td>
<td>1C</td>
<td>8</td>
<td>27.A-D.4.A</td>
<td>Station 3</td>
</tr>
<tr>
<td>SSIGPH41</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.4.B</td>
<td>Geophone Reading 1</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td>Station 4</td>
</tr>
<tr>
<td>SSIGPH42</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.4.C</td>
<td>Geophone Reading 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Station 4</td>
</tr>
<tr>
<td>SSIGPH43</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.4.D</td>
<td>Geophone Reading 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Station 4</td>
</tr>
<tr>
<td>SSIGPH44</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.4.E</td>
<td>Geophone Reading 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Station 4</td>
</tr>
<tr>
<td>SSIGPH45</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.4.F</td>
<td>Geophone Reading 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Station 4</td>
</tr>
<tr>
<td>SSIGPH46</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.4.G</td>
<td>Geophone Reading 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Station 4</td>
</tr>
<tr>
<td>SSIGPH47</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.4.H</td>
<td>Geophone Reading 7</td>
</tr>
<tr>
<td></td>
<td>1C</td>
<td>8</td>
<td>27.A-D.5.A</td>
<td>Station 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SSI Lane Station 5</td>
</tr>
<tr>
<td>SSIGPH51</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.5.B</td>
<td>Geophone Reading 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Station 5</td>
</tr>
<tr>
<td>SSIGPH52</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.5.C</td>
<td>Geophone Reading 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Station 5</td>
</tr>
<tr>
<td>SSIGPH53</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.5.D</td>
<td>Geophone Reading 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Station 5</td>
</tr>
<tr>
<td>SSIGPH54</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.5.E</td>
<td>Geophone Reading 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Station 5</td>
</tr>
<tr>
<td>SSIGPH55</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.5.F</td>
<td>Geophone Reading 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Station 5</td>
</tr>
<tr>
<td>SSIGPH56</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.5.G</td>
<td>Geophone Reading 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Station 5</td>
</tr>
<tr>
<td>SSIGPH57</td>
<td>5.2N</td>
<td>8</td>
<td>27.A-D.5.H</td>
<td>Geophone Reading 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Station 5</td>
</tr>
</tbody>
</table>
## PES Modification Program
### Program Specification (continued)

#### Score Information File Record Layout

<table>
<thead>
<tr>
<th>Field</th>
<th>Size/Type</th>
<th>Note</th>
<th>Item No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>* HWYDST</td>
<td>2N</td>
<td>1</td>
<td>1</td>
<td>Highway District Number</td>
</tr>
<tr>
<td>* HWYPREFX</td>
<td>2C</td>
<td>2</td>
<td>4.A</td>
<td>Maintenance Section Number</td>
</tr>
<tr>
<td>* HWYNUM</td>
<td>4N(C)</td>
<td>3</td>
<td>4.B</td>
<td>County Number</td>
</tr>
<tr>
<td>* BEGMPST</td>
<td>3N</td>
<td>5.A</td>
<td>5.A</td>
<td>Highway Number</td>
</tr>
<tr>
<td>* BDISSIGN</td>
<td>1C</td>
<td>5.B</td>
<td>5.B</td>
<td>Highway Suffix</td>
</tr>
<tr>
<td>* BMPSTDIS</td>
<td>3.1N</td>
<td>1</td>
<td>5.C</td>
<td>Segment Begin - Milepost No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Displacement Sign</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(9.9) - Displacement Value</td>
</tr>
<tr>
<td>*</td>
<td>3C</td>
<td>11</td>
<td>28.A-D.1</td>
<td>Laneset</td>
</tr>
<tr>
<td></td>
<td>4.2N</td>
<td>9</td>
<td>28.A-D.2</td>
<td>Scores - Identifier</td>
</tr>
<tr>
<td></td>
<td>4.2N</td>
<td>10</td>
<td>28.A-D.3</td>
<td>- MRM Utility</td>
</tr>
<tr>
<td></td>
<td>4.2N</td>
<td>10</td>
<td>28.A-D.4</td>
<td>- Skid Utility</td>
</tr>
<tr>
<td></td>
<td>4.2N</td>
<td>10</td>
<td>28.A-D.5</td>
<td>- Maintenance Cost</td>
</tr>
<tr>
<td>UVURS</td>
<td>4.2N</td>
<td>10</td>
<td>28.A-D.6</td>
<td>- Unweighted Visual</td>
</tr>
<tr>
<td></td>
<td>4.2N</td>
<td>10</td>
<td>28.A-D.7</td>
<td>- Adjusted Visual</td>
</tr>
<tr>
<td></td>
<td>4.2N</td>
<td>10</td>
<td>28.A-D.8</td>
<td>- Weighted Visual</td>
</tr>
<tr>
<td>PESPRS</td>
<td>4.2N</td>
<td>10</td>
<td>28.A-D.9</td>
<td>- Unweighted Pavement</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Pavement Score</td>
</tr>
</tbody>
</table>

* Key Field
PES Modification Program
Program Specification (continued)

Notes on Output Record Layouts

General Notes:

A. All decimal points are implied in the Original PES File. The Size/Type column in the output layouts include 1 extra position for the decimal which physically exists, except for the Beginning and Ending Milepoint Displacements. The Displacements have an implied decimal point in the output files also.

B. All dates are in the format of YYMMDD.

C. Refer to the original PES record layout for an explanation of each PES item.

D. The Visual Evaluation is taken on the worst lane in the roadway. There will be MRM and SSI data for all roadways in which a visual evaluation has been made. However, for a Highway Design Code of 1 or 2, the roadway would be R-L instead of R-V and L-P; hence the visual evaluation could be taken in lane 'L' and the MRM or the SSI could be taken in the 'R' lane. For Highway Design Code 1 or 2, the program only creates one record in each modified PES file (See Specific Note 3). If the Visual evaluation, MRM, SSI and the Score data are not available for all the roadways which exist for the segment (see Highway Design), the program reports the test segment and which roadway and data are missing. Currently, there will only be up to 4 visual evaluation lanes - 1 per roadway.

E. The following formula is used to calculate the standard deviation

\[
s = \sqrt{\frac{\left(\sum x_i - x\right)^2}{n - 1}}
\]

where \( x = \frac{\sum x_i}{n} \)

F. A field name is provided when the field directly corresponds to a field in the dBASE III Texas Flexible Pavement Database. The field name is the same as that used in the database.
Specific Notes:

The record layouts refer to these notes in the Note column of the layout.

1. Stored without the decimal place (eg. 3.0 is stored as 30)

2. See the Original PES File Layout documentation.

3. A separate record should be created for original PES items 18.A.1, 18.B.1, 18.C.1, 18.D.1, 18.E.1, and 18.F.1. See General Note D.

4. If the Visual Evaluation lane is in the R-V laneset, then the program extracts laneset R-V cells 1-15 for the MRM readings. The MRM reading are in the format of lane-SI value (eg. R01 - the SI value has an implied decimal point between the two digits). If the Visual Evaluation lane is in the I-P laneset, the program extracts the I-P laneset, MRM readings, etc. (If the Highway Design Code is 1 or 2, the SI values are in the R-V cells.)

5. If the Visual Evaluation lane is in the R-V laneset, then the program extracts the Skid cells 1-15 for the R-V laneset. If the Visual Evaluation lane is in the I-P laneset, the program extracts the I-P laneset skid values, etc. The skid values are in the format of lane-SN value (eg. L20). (If the Highway Design Code is 1 or 2, the Skid values are in the R-V cells.)

6. The program only extracts PES records with an 'A' in the Mandatory Sample Selection Flag field and an 'E', 'F', 'G', 'H', 'I', 'N', 'O', 'P', 'Q', or 'R' in the Other Sample Selection Flag field.

7. As with the MRM and Skid data, the SSI data is given for 4 possible roadways. If the Visual Evaluation Lane is in the R-V laneset, then the program extracts the R-V laneset SSI data, etc. (If the Highway Design Code is 1 or 2, the SSI values are in the R-V cells.)

8. The actual geophone reading for the lanesets are provided in item 27. There are 5 sets of 7 geophone readings per roadway. The lane in which the reading was taken as well as the 7 readings are given for each of the 5 sets of data. The program extracts the readings for the appropriate roadway based on the Visual Evaluation Lane.

9. See Footnote 7 of the Original PES record layout documentation.
10. The program picks up the appropriate roadway scores based on the Visual Evaluation Lane and Footnote 7 of the original PES File layout.

11. The laneset field will contain the laneset in which the Visual Evaluation Lane is in (eg. 'R-V', 'R-L', 'L-P', 'A-C', 'X-Z').

12. The program calculates the MRM Mean, Standard Deviation, High Value, Low Value, and Count of Observations for the laneset.

13. The program calculates the SN (Skid) Mean, High Value, Low Value, Standard Deviation, and Count of Observations in the laneset.
<table>
<thead>
<tr>
<th>Hwy</th>
<th>Hwy</th>
<th>Cnty</th>
<th>Begin</th>
<th>Hwy</th>
<th>Des Laneset</th>
<th>Vis</th>
<th>MRM</th>
<th>Skd</th>
<th>SSI</th>
<th>Scr</th>
</tr>
</thead>
<tbody>
<tr>
<td>xx</td>
<td>xx</td>
<td>xxx</td>
<td>xxxxxx</td>
<td>x xxx</td>
<td>x</td>
<td>x-x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Notes:
1. Highway - Highway System, Number, Suffix
2. Highway Design - Item 24 and Footnote 6 of the Original PES File Record Layout Documentation - Values 1-9
4. Place an 'X' in the column for which data is missing.
PROGRAM LISTING

(* File name:  PESMOD.PAS
* Program name:  pesmod
* Project 2456:  Texas Flexible Pavement Database Conversion
* TAMU/TTI
* Written by:  Victor Wong
* Written on:  Feb 2, 1988
* Last updated:  May 13, 1988
* Purpose:  This Turbo PASCAL program reads in PES data
*           and extract information for the project
*           and output it in comma-delimited format
* (suitable for DBASE III + and BASIC).
* )

{>>> PROGRAM PESMOD <<<}
program pesmod;

{>>> constant declarations <<<}
const
  comma = ', ';
  dect = '. ';
  blank = ' ';
  zero = '0';
  item_array_size = 15;

{>>> type declarations <<<}
type
  item_array_range = array[1..item_array_size] of real;
  characters = set of 'A'..'Z';
  file_str = string[30];

  laneset = (rv,lp,ac,xz);
  mmr_rec = record
    (1C  - lane)
    (2.1N - SI value)
    (15 sets for each laneset)
    (** NOTE: lane blank if no si **)  
    (** NOTE: SI value 0.0 if no si **)  
    (values 0.1 - 5.0 **)
end;

  skid_rec = record
    (1C  - lane)
    (2N  - SN value)
    (15 sets for each laneset)
    (** NOTE: lane blank when no SN **)  
    (** SN value 0.0 when no

data**)
end;
ssi_gen_rec = record
  date    : string[6];
  avg     : string[3];
  temp    : string[3];
end;

ssi_rec = record
  lane: array[1..5] of char;
  gph[1..5] of string[4];
  gph[6..7] of string[4];
end;

score_rec = record
  id      : string[3];
  mrm     : string[3];
  skid    : string[3];
  mncost  : string[3];
  uwvis   : string[3];
  adjvis  : string[3];
  rvw     : string[3];
  uwvimt  : string[3];
  pvmt    : string[3];
end;

{>>> variables declarations <<<}
var
  hwydist    : string[2];
  hwysect    : string[2];
  cntynum    : string[3];
  hwprefx    : string[2];
  hwnum      : string[4];
  hwysuffix  : char;
  bmpnum     : string[3];
  bmpdispsgn : char;
  bmpdispval : string[2];

{ SSI general}
{ 6N  - date YYMMDD}
{ 3N  - average SSI}
{ 3N  - temperature}
{  (1 set for each laneset)}
{  ** NOTE: avg, temp zero if no data}

{ SSI}
{ 1C  - Lane}
{ 4.2N - gphn1}
{ 4.2N - gphn2}
{ 4.2N - gphn3}
{ 4.2N - gphn4}
{ 4.2N - gphn5}
{ 4.2N - gphn6}
{ 4.2N - gphn7}
{  (5 such set for each laneset)}
{  ** NOTE: lane blank when no ssi
  gph zero when no data **}

{ scores}
{ 3C  - identifier}
{ 3.2N - MRM utility}
{ 3.2N - skid utility}
{ 3.2N - maintenance cost}
{ 3.2N - unweighted visual}
{ 3.2N - adjusted visual}
{ 3.2N - weighted visual}
{ 3.2N - unweighted pavement}
{ 3.2N - Pavement score}
{  (1 such set for each laneset)}
{  ** NOTE: id blank when no score
  others 0.0 when no cal.**}
empnum : string[3]; (3N - milepost no.)
empdispsgn : char; (1C - displacement sign)
empdispval : string[2]; (2.1N - displacement value)

intpst1 : string[3]; (3N - 1st post no.)
intpst2 : string[3]; (3N - 2nd post no.)
inttdis1 : string[2]; (2.1N - distance to 1st)
inttdis2 : string[2]; (2.1N - distance to 2nd)

rtcntybeg : string[4]; (4.1N - begin value)
rtcntyend : string[4]; (4.1N - end value)

seglnth : string[2]; (2.1N segment length)
segupdat : string[6]; (6N last updated date)
segmncost : string[9]; (9.2N segment maintenance cost)
adt : string[6]; (6N average daily traffic)
alsk : string[5]; (5N 18-kip equivalency (x 1000))
funclas : char; (1N functional classification)
umlanes : string[2]; (2N no. of lanes)
spdlimt : string[2]; (2N speed limit)
rissurf : string[2]; (2N RIS surface type)

vislane : array[1..6] of char; (1C - lane)
visdate : array[1..6] of string[6]; (6N - date evaluated)
viscomm : array[1..6] of string[2]; (2N - comment)
vispvmnt : array[1..6] of string[2]; (2N - pavement type)
visfail : array[1..6] of string[3]; (3N - fail-acp-b)
vistran : array[1..6] of string[3]; (3N - tran-b-appjntsp)
vislong : array[1..6] of string[3]; (3N - long-b-pocp)
virut : array[1..6] of string[3]; (3N - rut-spall-failjnt)
visallg : array[1..6] of string[3]; (3N - allg-crccks-pco-lngcrck)
visblk : array[1..6] of string[3]; (3N - bck-pocp-shatslb)
visptch : array[1..6] of string[3]; (3N - patch-pnch-failure)

** NOTE: lane blank if no eval. **

mmupdat : string[6]; (6N - last updated date YYMMDD)
mmavgsi : string[2]; (2.1N - average SI value)

** NOTE: 0.0 when none to avg **
skid_data : array [laneset] of skid_rec;

surfwidth : string[3]; (3N surface width)
hwydesign : char; (1N highway design type)

smplflgman : char; (1C - mandatory)
smplflgoth : char; (1C - other)

(** NOTE: mandatory non-blank
   -> segment must be tested
   other non-blank
   -> special-purpose

segment)

ssi_gen_data : array [laneset] of ssi_gen_rec;

ssi_data : array [laneset] of ssi_rec;

score_data : array [laneset] of score_rec;

lane_set : string[3]; (3N laneset)

smean : real; (MRM 6.5N Mean for the laneset)
sisd : real; (6.5N Std. Dev. for the laneset)
sihi : real; (2.1N high value for the laneset)
laneset)
silo : real; (2.1N low value for the laneset)
sicount : integer; (3N count of obs. for the laneset)

skidmean : real; (skid 2N Mean for the laneset)
skidsd : real; (2N Std. Dev for the laneset)
skidhi : real; (2N High value for the laneset)
skidlo : real; (2N Low value for the laneset)
skidcount : integer; (2N count of obs. for the laneset)

ls, ls1, begin ls: laneset;
l : characters;
i, j,
vis_count, visln_pos,
temp_si, temp_sn,
err_code : integer;
mm_stat array,
skid_stat array: item_array_range;
mm_set_flag,
skid_set_flag,
ssigen_set_flag,
ssi_set_flag,
score_set_flag,
vis_set_flag,
mmnln_flag,
skiDln_flag,
ssi_ln_flag,
vis_ln_flag   : boolean;
tempout,
report_file,
report_data_file,
in_file,
gen_file,
vis_file,
mmn_file,
skiD_file,
ssi_file,
scr_file     : text;
filename     : file_str;
krecords_checked,
records_checked,
krecord_no,
record_no    : integer;
record_no_pos : integer;

---------------------------------------------------------------------
--
(>>> procedure stat: It takes an array of real numbers and <<<)
(>>> find the high and low value, mean, std<<<)
(>>> deviation, and count of non-zero data <<<)

procedure stat (item_array: item array_range; array_size: integer;
               var hi,lo,mean,sd: real;
               var count: integer);
    (**variable declaration**) 
    var
        sum,sdu_sum: real;
        i: integer;
        still_have_data: boolean;

    (**beginning of the procedure**) 
    begin
        (**initialize variables**) 
        sum:= 0.0;
        sdu_sum:= 0.0;
        count:= 1;
        hi:= item_array[1];
        lo:= item_array[1];
        still_have_data:= true;

        (**loop to read data,perform summation and counting**) 
        while still_have_data do 
            if item_array[count]=0.0 then 
                still_have_data:= false 
            else 
                begin 

231
sum := sum + item_array[count];
if item_array[count] > hi then hi := item_array[count];
if item_array[count] < lo then lo := item_array[count];
count := count + 1;
if count > array_size then still_have_data := false
end;
count := count - 1;

{---calculate the mean---}
if count <> 0 then mean := sum/count;

{---calculate the std. dev.---}
if count = 1 then
sd := 0.0
else begin
for i := 1 to count do
sd_sum := sd_sum + sqr (item_array[i] - mean);
end;
sd := sqrt (sd_sum/(count-1));
end;

{---ending of the procedure stat---}

-----------------------------------------------------------------

{>>> procedure OpenInFile : This procedure asks user for the input
file name and tries to open it. <<<}

procedure OpenInFile (var in_file: text; var filename: file_str);
{---variable declaration---}
var
  infilename: string[20];
  ok: boolean;
  yr: string[2];

{---beginning of procedure openinfile---}
begin
repeat
  filename := '\PAVED\PES';
infilename := filename + '.DAT';
assign (in file, infilename);
{$I-} reset (in file) {$I+};
ok := (IOresult = 0);
if not ok then
begin
  clrscr;
gotoxy (14,14);
  writeln ('Cannot find the file ', infilename);
end;
until ok;
end;
{---ending of the procedure openinfile---}
procedure OpenOutFile : This procedure opens the output file. <<<

procedure OpenOutFile (var gen,
   vis,
   mum,
   skd,
   ssi,
   scr : text;
   filename : file_str);

(--beginning of the procedure openoutfile--)
begin
   assign (gen, '\PAVEDB\PESGEN.DIM');
   rewrite (gen);
   assign (vis, '\PAVEDB\PESVIS.DIM');
   rewrite (vis);
   assign (mum, '\PAVEDB\PESMUM.DIM');
   rewrite (mum);
   assign (skd, '\PAVEDB\PESSKD.DIM');
   rewrite (skd);
   assign (ssi, '\PAVEDB\PESSSI.DIM');
   rewrite (ssi);
   assign (scr, '\PAVEDB\PESSCR.DIM');
   rewrite (scr);
end;
(--ending of the procedure openoutfile--)

procedure OpenReportFile: This procedure asks the user for the name
<< of the report file and open it for output.

   filename: file_str);

(--variable declarations--)
var
   reportfilename: string[30];
   reportdatafile: string[30];

(--beginning of the procedure openreportfile--)
begin
   reportfilename:= filename + '.rep';
   assign (report_file, reportfilename);
   rewrite (report_file);
   reportdatafile:= filename + '.rpdata';
   assign (report_data_file, reportdatafile);
   rewrite (report_data_file);
end;
(-- ending of the procedure openreportfile --)

{----------------------------------------------------------
--}

{>>> procedure ProduceReport: This procedure produces the missing data
<<<}
{>>>}
{<<<}

procedure ProduceReport (var report_file, report_data_file: text);
    var
        record_line: integer;
        data_line: string[132];
    begin
        for i := 1 to 132 do data_line[i] := ' '; 
        close (report_data_file);
        reset (report_data_file);
        record_line := 0;
        while not eof (report_data_file) do begin
            if (record_line = 0) or (record_line = 50) then begin
                record_line := 0;
                {>>> write the heading for the report <<<}
                writeln (report_file, 'L');
                writeln (report_file, 'Texas Flexible
Pavement Database');
                writeln (report_file, 'PES
Modification Program');
                writeln (report_file, 'Data Report');
                writeln (report_file);
                writeln (report_file);
                writeln (report_file, 'Hwy
---Missing data----');
                writeln (report_file, 'Hwy Cnty Begin
pt. Des Laneset Vis MRM Skd SSI Scr');
                writeln (report_file, 'Dist Sec Num Highway Mile

end;
            readln (report_data_file, data_line);
            writeln (report_file, data_line);
            record_line := record_line + 1;
        end;
    end;

{----------------------------------------------------------
--}

{>>> beginning of the program <<<}
begin
for i:= 1 to 15 do read (in_file, lane[i], si[i]);

(--- read skid data ---)
read (in_file, skidupdat, skidavgsn);
for ls:= rv to xz do
  with skid_data[ls] do
    for i:= 1 to 15 do read (in_file, lane[i], sn[i]);

(--- read some more general data ---)
read (in_file, surfwidth, hwydesign);
read (in_file, smplflgman, smplflggoth);

(--- read ssi data ---)
for ls:= rv to xz do
  with ssi_gen_data[ls] do
    read (in_file, date, avg, temp);

for ls:= rv to xz do
  with ssi_data[ls] do
    for i:= 1 to 5 do
      begin
        read (in_file, lane[i], gph1[i], gph2[i]);
        read (in_file, gph3[i], gph4[i], gph5[i]);
        read (in_file, gph6[i], gph7[i]);
      end;

(--- read score data ---)
for ls:= rv to xz do
  with score_data[ls] do
    begin
      read (in_file, id, mrm, skid, mnccost, uwvis);
      read (in_file, adjvis, wvis, uwpvmt, pvmt);
    end;

(--- goto the next line ---)
readln (in_file);
if records_checked = 999 then
  begin
    krecords_checked:= krecords_checked + 1;
    records_checked:= 0;
  end
else
  records_checked:= records_checked + 1;

{>>> only records with an 'A' in the Mandatory Sample Selection<<<
{>>> Flag field and an 'E','F','G','H','I','N','O','P','Q' or <<<
{>>> 'R' in the Other Sample Selection Flag field should be <<<
{>>> processed. <<<
if (smplflgman = 'A') and ((smplflggoth='E') or (smplflggoth='F')
  or (smplflggoth='G') or (smplflggoth='H') or (smplflggoth='I')
  or (smplflggoth='N') or (smplflggoth='O') or (smplflggoth='P')
  or (smplflggoth='Q') or (smplflggoth='R') )then
  begin

236
let user know that the program is running
clrscr;
gotoxy (15,3);
write ('*******************************************************************************');
gotoxy (15,4);
write ('* Program PESMOD is running. *');
gotoxy (15,5);
write ('* This program will modify the PES data tape file. *');
gotoxy (15,6);
write ('*******************************************************************************');

assign input and output files
clrscr;
OpenInFile(in_file, filename);
OpenOutFile(gen_file, vis_file, mmn_file, skd_file, ssi_file, scr_file, filename);
OpenReportFile (report_file, report_data_file, filename);

initialize record counter
clrscr;
gotoxy (14,12);
write ('Processing...');
gotoxy (21,13);
write ('new modified records');
record_no:= 0;
record_no_pos:= 17;
records_checked:= 0;
records_checked:= 0;

while not eof (in_file) do
begin
read in data
begin
read some general data
begin
read (in_file, hwdist, hwysect, cntynum);
read (in_file, hwyprefx, hynum, hwsuffix);
read (in_file, bnpnum, bnpdispsgn, bnpdispval);
read (in_file, emnum, emdispsgn, emdispval);
read (in_file, intpst1, intpst2, intdis1, intdis2);
read (in_file, rctntybeg, rctntyend);
read (in_file, seglngth, sequpdat, segmncost);
read (in_file, adt, a18k, funclas, numlanes, spdlimit, rissurf);

read visual data
begin
for i:= 1 to 6 do
begin
read (in_file, vislane[i], visdate[i], viscomm[i], vispvmt[i]);
read (in_file, visfail[i], vistran[i], vislong[i], visrout[i]);
read (in_file, visallg[i], visblck[i], visptch[i]);
end;

read msn data
begin
read (in_file, mrmupdat, mrmavgcsi);
for ls:= rv to xz do
begin
with msn_data[ls] do
end;

end;
};
(>>> check and see what kind of data is available <<<)
(>>> for the different lanes in the four lanesets <<<)

(--- if highway design number is 1 or 2, ---)
(--- laneset rv and lp is considered the same ---)
if (hwydesign = '1') or (hwydesign='2') then
    begin_ls:= lp
else
    begin_ls:= rv;

(--- for each laneset, check to see what data is available ---)
for ls:= begin_ls to xz do
    begin
        ( if data is available, assign the set_flag to be true )
        mrm_set_flag:= false;
        skid_set_flag:= false;
        ssigen_set_flag:= false;
        ssi_set_flag:= false;
        score_set_flag:= false;
        vis_set_flag:= false;

        if mrm_data[ls].lane[1] <> '' then mrm_set_flag:= true;
        if skid_data[ls].lane[1] <> '' then skid_set_flag:= true;
        if ssi_gen_data[ls].date>'000000' then ssigen_set_flag:= true;
        if ssi_data[ls].lane[1] <> '' then ssi_set_flag:= true;
        if score_data[ls].id <> '' then score_set_flag:= true;
        vis_count:= 1;
        while (not vis_set_flag) and (vis_count <= 6) do
            begin
                if (vislane[vis_count] <> '') then
                    vis_set_flag:= true;
                    vis_count:= vis_count + 1;
                end;

    ( when there is some data for the laneset, proceed to write )
    ( records for the lanes in the laneset that have data )
    if mrm_set_flag or skid_set_flag or ssigen_set_flag
    or ssi_set_flag or score_set_flag or vis_set_flag
    then begin
        ( assign the correct lane range of the present laneset )
        if (hwydesign='1') or (hwydesign='2') then
            case ls of
                lp: l:=['R','L'];
                ac: l:=['A'..'C'];
                xz: l:=['X'..'Z'];
            end
            else
                case ls of
                    rv: l:=['R'..'V'];
                    lp: l:=['L'..'P'];
                    ac: l:=['A'..'C'];
                    xz: l:=['X'..'Z'];
                end;
{ for each laneset, write a record if data exists }

{ if data is available, assign the ln_flag to be true }
mm_ln_flag:= false;
skid_ln_flag:= false;
ssi_ln_flag:= false;
vis_ln_flag:= false;

if mm_set_flag and (mm_data[ls].lane[1] in l) then mm_ln_flag := true;
if skid_set_flag and (skid_data[ls].lane[1] in l) then skid_ln_flag := true;
if ssi_set_flag and (ssi_data[ls].lane[1] in l) then ssi_ln_flag := true;
{ This is assuming that there will only be }  
{ a set of visual data for one lane per laneset. }  
{ Meaning that there should not be data for }  
{ lane 'R' and 'S'. Otherwise, only the first one }  
{ will be picked up. }
vis_count:= 1;
visln_pos:= 0;
while (visln_pos = 0) and (vis_count <= 6) do begin
  if (vislane[vis_count] in l) then
    visln_pos:= vis_count;
    vis_count:= vis_count + 1;
  end;
if visln_pos <> 0 then vis_ln_flag:= true;

{ when there is some data for the laneset, proceed to write }
{ out the record }
if mm_ln_flag or skid_ln_flag or ssi_ln_flag 
or vis_ln_flag then begin  
  { check for the laneset }
  if (hwydesign = '1') or (hwydesign = '2') then case ls of
    lp: lane_set:= 'R-L';
    ac: lane_set:= 'A-C';
    xz: lane_set:= 'X-Z';
  end
else
  case ls of
    rv: lane_set:= 'R-V';
    lp: lane_set:= 'L-P';
    ac: lane_set:= 'A-C';
    xz: lane_set:= 'X-Z';
  end;
  { write out some key data }
write (gen file, hwydist, comma, hwysect, comma, 
cntynum, comma, '"', hwyprefix, '"', comma, 
hwynum, comma, '"', hwy suffix, '"', comma, 
...);
bmpnum, comma,        "", bmpdispsgn, "", comma,
bmpdispval, comma);  write (gen_file, "", lane_set, "", comma);

write (vis_file,      hwysect, comma,
hwydist, comma,      "", hwyprefix, "", comma,
cntynum, comma,      "", hwynum, comma,      "", hwynum, comma,      "", hwsuffix, "", comma,
bmpnum, comma,      "", bmpdispsgn, "", comma,
bmpdispval, comma);  write (vis_file, "", lane_set, "", comma);

write (mrm_file,      hwysect, comma,
hwydist, comma,      "", hwyprefix, "", comma,
cntynum, comma,      "", hwynum, comma,      "", hwynum, comma,      "", hwsuffix, "", comma,
bmpnum, comma,      "", bmpdispsgn, "", comma,
bmpdispval, comma);  write (mrm_file, "", lane_set, "", comma);

write (skd_file,      hwysect, comma,
hwydist, comma,      "", hwyprefix, "", comma,
cntynum, comma,      "", hwynum, comma,      "", hwynum, comma,      "", hwsuffix, "", comma,
bmpnum, comma,      "", bmpdispsgn, "", comma,
bmpdispval, comma);  write (skd_file, "", lane_set, "", comma);

write (ssi_file,      hwysect, comma,
hwydist, comma,      "", hwyprefix, "", comma,
cntynum, comma,      "", hwynum, comma,      "", hwynum, comma,      "", hwsuffix, "", comma,
bmpnum, comma,      "", bmpdispsgn, "", comma,
bmpdispval, comma);  write (ssi_file, "", lane_set, "", comma);

write (scr_file,      hwysect, comma,
hwydist, comma,      "", hwyprefix, "", comma,
cntynum, comma,      "", hwynum, comma,      "", hwynum, comma,      "", hwsuffix, "", comma,
bmpnum, comma,      "", bmpdispsgn, "", comma,
bmpdispval, comma);  write (scr_file, "", lane_set, "", comma);

{ write out some general data }
write (gen_file, empnum, comma,
"", empdispsgn, "", comma, empdispval, comma);  if (intptst1 = ') then
    write (gen_file, zero:3, comma)
else
    write (gen_file, intptst1, comma);
if (intptst2 = ') then
    write (gen_file, zero:3, comma)
else
    write (gen_file, intpsl2, comma);
if (intdi1 = ' ')
    write (gen_file, zero, dect, zero, comma)
else
    write (gen_file, copy(intdisl, 1, 1), dect,
           copy(intdisl, 2, 1), comma);
if (intdis2 = ' ')
    write (gen_file, zero, dect, zero, comma)
else
    write (gen_file, copy(intdis2, 1, 1), dect,
           copy(intdis2, 2, 1), comma);
write (gen_file,
       copy(rtcntybeg, 1, 3), dect, copy(rtcntybeg, 4, 1), comma,
       copy(rtcntyend, 1, 3), dect, copy(rtcntyend, 4, 1), comma,
       copy(seglnth, 1, 1), dect, copy(seglnth, 2, 1), comma,
       segupdat, comma,
       copy(segmncoast, 1, 7), dect, copy(segmncoast, 8, 2), comma,
       adt, comma, al8k, comma,
       funclas, comma, numlanes, comma,
       spdlimt, comma, rissurf, comma);

{ write out the visual data }
if vis_in_flag then
    begin { for existing data }
        write(vis_file,
               '', vislane[visn_pos], '', comma,
               visdate[visn_pos], comma,
               viscomm[visn_pos], comma,
               vispvmnt[visn_pos], comma,
               visfail[visn_pos], comma,
               vistran[visn_pos], comma,
               vislorg[visn_pos], comma,
               vis FLT[visn_pos], comma,
               visall[visn_pos], comma,
               visbck[visn_pos], comma,
               visptch[visn_pos]);
    end
else
    begin { for empty fields }
        write(vis_file, '', blank, '');
        for j := 1 to 10 do write(vis_file, comma, zero);
    end;
writeln (vis_file);

{ write out mmr data }
write (mnr_file, mmmupdat, comma, copy(mnravgsi, 1, 1),
       dect, copy(mnravgsi, 2, 1), comma);
if mmr_ln_flag then
    for i := 1 to 15 do { for existing data }
        write (mnr_file, '', mmr_data[ls].lane[i], '',
               comma, copy(mmr_data[ls].si[i], 1, 1), dect,
               copy(mmr_data[ls].si[i], 2, 1), comma)
else
for i:= 1 to 15 do { for empty fields }
    write (mrm_file, "", blank, "", comma,
            zero, comma);

{ write out skid data }
write (skd_file, skdupdat, comma, skidavgdsn, comma);
if skid ln flag then
    for i:= 1 to 15 do { for existing data }
        write (skd_file, ",", skid_data[ls].lane[i], ",",
               comma, skid_data[ls].sn[i], comma)
    else
        for i:= 1 to 15 do { for empty fields }
            write (skd_file, ",", blank, ",", comma,
                   zero, comma);

{ write out some more general data }
writeln (gen_file, surfwidth, comma,
        hwydesign, comma,
        ",", smplflgman, ",", comma,
        ",", smplflgoth, ")

{ write out ssi general data }
if ssigen_set_flag then
    with ssi_gen_data[ls] do { for existing data }
        write (ssi_file, date, comma,
               avg, comma,
               temp)
    else { for empty fields }
        write (ssi_file, zero, comma, zero, comma, zero);

{ write out ssi data }
if ssi ln_flag then
    with ssi_data[ls] do
        for i:= 1 to 5 do { for existing data }
            begin
                write (ssi_file, comma, ",", lane[i], ",")
                write (ssi_file,
                       comma, copy(gph1[i], 1, 2), decpt, copy(gph1[i], 3, 2),
                       comma, copy(gph2[i], 1, 2), decpt, copy(gph2[i], 3, 2),
                       comma, copy(gph3[i], 1, 2), decpt, copy(gph3[i], 3, 2),
                       comma, copy(gph4[i], 1, 2), decpt, copy(gph4[i], 3, 2),
                       comma, copy(gph5[i], 1, 2), decpt, copy(gph5[i], 3, 2),
                       comma, copy(gph6[i], 1, 2), decpt, copy(gph6[i], 3, 2),
                       comma, copy(gph7[i], 1, 2), decpt, copy(gph7[i], 3, 2))
            end
        else
            for i:= 1 to 5 do { for empty fields }
                begin
                    write (ssi file, comma, ",", blank, ",")
                    for j:= 1 to 7 do
                        write (ssi_file, comma, zero, decpt, zero)
                    end;
    writeln (ssi_file);
{ write out score data }
if score_set_flag then
with score_data[ls] do { for existing data }
    write(scr_file, "", id, "", comma,
        copy(mm, 1, 1), decpt, copy(mm, 2, 2), comma,
        copy(skid, 1, 1), decpt, copy(skid, 2, 2), comma,
        copy(mncost, 1, 1), decpt, copy(mncost, 2, 2), comma,
        copy(uwvis, 1, 1), decpt, copy(uwvis, 2, 2), comma,
        copy(adjvis, 1, 1), decpt, copy(adjvis, 2, 2), comma,
        copy(wvis, 1, 1), decpt, copy(wvis, 2, 2), comma,
        copy(uwpvmt, 1, 1), decpt, copy(uwpvmt, 2, 2), comma,
        copy(pvmt, 1, 1), decpt, copy(pvmt, 2, 2))
else
    begin { for empty fields }
        write(scr_file, ",", blank, blank, blank, ",");
        for i := 1 to 8 do
            write(scr_file, comma, zero, decpt, zero);
    end;
writeln(scr_file);

{--- calculate the statistics on si values ---}
sicount := 0;
sihi := 0.0;
silo := 0.0;
simean := 0.0;
sisd := 0.0;
if mmn_ln_flag then
begin
    with mmn_data[ls] do
        for i := 1 to 15 do
            begin
                val (si[i], temp_s, err_code);
                mmn_stat_array[i] := temp_s/10.0;
            end;
    stat(mmn_stat_array, item_array_size, sihi,
        silo, simean, sisd, sicount);
end;

{ write out the mmn stat. }
writeln(mmn_file, simean:7:5, comma, sisd:7:5, comma,
    sihi:3:1, comma,
    silo:3:1, comma, sicount:3);

{--- calculate the statistics on skid values ---}
skidcount := 0;
skidhi := 0;
skidlo := 0;
skidmean := 0;
skidsd := 0;
if skid_ln_flag then
begin
    with skid_data[ls] do
        for i := 1 to 15 do
begin
  val (sn[i], temp_sn, err_code);
  skid_stat_array[i] := temp_sn;
end;
stat (skid_stat_array, item_array_size, skidhi,
      skidlo, skidmean, skidsd, skidcount);
end;

{ write out the skid stat. }
writeln (skd_file, skidmean:2:0, comma, skidsd:2:0, comma,
         skidhi:2:0, comma, skidlo:2:0, comma,
         skidcount:2);

{ write out the missing data report for lanes }
write (report_data_file, '$');
write (report_data_file, hwydist, '$',
      hwysect, '$',
      cntynum, '$',
      hwyprefx, '$', 'hwnum', '$', 'hwsuffx', '$',
      bmpnum, '$', 'bmpdispgrn', '$', 'bmpdispval', '$',
      hwydesign, '$',
      lane_set, '$');
if vis_ln_flag then
  write (report_data_file, ' ')
else
  write (report_data_file, 'X');
write (report_data_file, ' ');
if mmm_ln_flag then
  write (report_data_file, ' ')
else
  write (report_data_file, 'X');
write (report_data_file, ' ');
if skid_ln_flag then
  write (report_data_file, ' ')
else
  write (report_data_file, 'X');
write (report_data_file, ' ');
if ssi_ln_flag then
  write (report_data_file, ' ')
else
  write (report_data_file, 'X');
write (report_data_file, ' ');
if score_set_flag then
  write (report_data_file, ' ')
else
  write (report_data_file, 'X');
write (report_data_file, ' ');
 writeln (report_data_file);

{--- increment the record counter ---}
if (record_no < 999) then
  begin
    gotoxy (record_no_pos, 15);
    record_no := record_no + 1;
    write (record_no:3);
end
else
begin
      gotoxy (record_no_pos-3,15);
krecord_no:=krecord_no+1;
write (krecord_no:3,'000');
record_no:=0;
end;
end; {then}

end; {else}
else
begin
  { write out the missing data report for lanes }
  write (report_data_file, ' ');
  writeln (report_data_file, hwydist, ' ',
hwysect, ' ',
cntynum, ' ',
hwprefix, ',hwnum, ',hwy_suffix, ' ',
bmpnum, ',bmpdispsgn, ',bmpdispval, ' ',
hwydesign, ' ',
'--', ' ', 'X X X X X X');

end; {else}
end; {for}
end; {if}
end; {while}

ProduceReport (report_file, report_data_file);

{>>> inform user that the process is done. <<<}
      gotoxy (45,13);
      write (krecords_checked:3,records_checked:3, ' PES records checked');
      gotoxy (14,14);
      write ('(NOTE: Missing data for PES records is reported in file ');
      gotoxy (14,15);
      write (' ',filename,'.REP.'));
      gotoxy (14,16);
      write ('Done.');

{>>> close input/output files <<<}
close (in_file);
close (gen_file);
close (vis_file);
close (mm_file);
close (skd_file);
close (ssi_file);
close (scr_file);
close (report_file);
close (report_data_file);

end.
This program is called after PESMOD.PAS is run. PESMOD.PAS creates the six comma delimited files that PESUP.PRG uses. These six files are appended to six dBASE temporary files (PESSKD.DBF, PESSMM.DBF, PESSSI.DBF, PESVISL.DBF, PESSCR.DBF, PESGEN.DBF). The program then checks to see if the same data has been added previously. If the data is already present in the Master Monitoring files, the program terminates itself. Otherwise, it adds data to the Master Monitoring files (Skid, Visual, Serviceability Index and Falling Weight) and the Location File.

The PESUP.PRG program checks to see if all the PES comma delimited files and PES temporary dBASE files are present before continuing further. An error report is produced for those SID numbers that do not have a matching PES record and those that have inconsistent data for Functional Classification, Number of Lanes, Ending Milepost and No Surface Layer. These programs and the temporary dBASE files are in the subdirectory \PAVEDB\EDITUPDT\PES.
PES UPDATE - PROGRAM FLOW DIAGRAM

FIGURE 12
PES Database Update Program
Program Specification

Program Name: PESUP.PRG

Purpose: Update the database with monitoring data from the PES File

Input File:
1) Modified PES temporary dBASE Files -
   Skid (PESSKD.DBF)
   Serviceability Index (PESMRM.DBF)
   Falling Weight SSI (PESSSI.DBF)
   Visual (PESVISL.DBF)
   Scores (PESSCR.DBF)
   Other Data (PESGEN.DBF)
2) dBASE III Location File (LOCATION.DBF)
3) dBASE III Layer ID File (LAYER.DBF)

Procedures/Edits:
See the Procedures pages.

Output Files:

Note: The following files are all dBASE III Files
1) SKID.DBF - Skid Measurement File
2) SI.DBF - Serviceability Index File
3) VISUAL.DBF - Visual Rating File
4) FALLIGHT.DBF - Falling Weight SSI File
5) LOCATION.DBF - Location File

dBASE Programs Called: PES_VISL.PRG
                      PES_MRM.PRG
                      PES_SSI.PRG
                      PES_SKID.PRG

Report: Missing and Inconsistent Data - see the Report Layout.

Note: The procedures pages are followed by the modified PES
dBASE temporary file layouts and the Missing and Inconsistent Data Report Layout.
1. New records are created for each active test section in the LOCATION File for the SKID, SI, SSI, and VISUAL files.

2. PESUPD.PRG matches the Modified PES File Records with the appropriate SID records using the LOCATION File. The files are matched to the Highway System, Number, Suffix, Beginning Milepoint Number, Beginning Milepoint Displacement Sign, Beginning Milepoint Displacement, and the Lane ID-Laneset.
   A LANEID of 'R' matches the LANESET 'R-L' and 'R-V'.
   A LANEID of 'L' matches the LANESET 'R-L' and 'L-P'.

3. For each type of monitoring data, the monitoring date is checked against the LAYER file to obtain the appropriate Structure Number and Layer Number for the new records. The Layer Number used is the last existing layer at the time of monitoring. The Structure Number corresponding to the chosen Layer Number is used.

4. For the SKID, SSI, and SI data, the information is used as it exists in the Modified PES File (The dates, however, are split into month, day, and year fields).

5. The VISUAL File data requires some mapping according to the following criteria:

<table>
<thead>
<tr>
<th>PES</th>
<th>Texas Flexible Pavement Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failures</td>
<td>FAILMILF</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>1 0 0</td>
<td>1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>2</td>
</tr>
<tr>
<td>0 0 1</td>
<td>3</td>
</tr>
<tr>
<td>Transverse Cracks</td>
<td>TRANCSRL</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 0 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>0 1 0</td>
<td>0 2 0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 3 0</td>
</tr>
<tr>
<td>Longitudinal Cracks</td>
<td>LONCSRL</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 0 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>0 1 0</td>
<td>0 2 0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 3 0</td>
</tr>
</tbody>
</table>
5. (cont.)

<table>
<thead>
<tr>
<th>PES</th>
<th>Rutting</th>
<th>RUTTSV</th>
<th>RUTTMD</th>
<th>RUTTSV</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>-------</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 0 0</td>
<td>-------</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0 1 0</td>
<td>-------</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>-------</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2 0 0</td>
<td>-------</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0 2 0</td>
<td>-------</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0 0 2</td>
<td>-------</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alligator Cracks</th>
<th>ALLGCRL</th>
<th>ALLGCRMD</th>
<th>ALLGCRLSV</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>-------</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 0 0</td>
<td>-------</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>-------</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0 0 1</td>
<td>-------</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Block Cracks</th>
<th>BLKCRMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>-------</td>
</tr>
<tr>
<td>1 0 0</td>
<td>-------</td>
</tr>
<tr>
<td>0 1 0</td>
<td>-------</td>
</tr>
<tr>
<td>0 0 1</td>
<td>-------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maintenance Patching</th>
<th>PATCHGD</th>
<th>PATCHFR</th>
<th>PATCHFR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>-------</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 0 0</td>
<td>-------</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>-------</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0 0 1</td>
<td>-------</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

6. PESUPD.PRG reports:

   a) Active test sections which do not have matching PES Records.

   b) The SID Number when the number of lanes is not the same on the PES and the Texas Flexible Pavement Database Files.

   c) The SID number when the Ending Milepoint, Ending Milepoint Displacement Sign, and Ending Milepoint Displacement are not the same on the PES and the Texas Flexible Pavement Database Files.

   d) The SID number if the Functional Classification is not the same on the PES and the Texas Flexible Pavement Database Files.

   e) The SID number if the Surface layer is absent in the Layer Identification file.
**PES-Skid File Layout**

**File Name:** PESSKD.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HWYDIST</td>
<td>*</td>
<td>2N</td>
<td>SDHPT Highway District</td>
</tr>
<tr>
<td>CNTYNUM</td>
<td>*</td>
<td>2N</td>
<td>County Number</td>
</tr>
<tr>
<td>HWYPREFX</td>
<td>*</td>
<td>2C</td>
<td>Highway Number Prefix</td>
</tr>
<tr>
<td>HWYNUM</td>
<td>*</td>
<td>4N</td>
<td>Highway Number</td>
</tr>
<tr>
<td>HWYSUFFX</td>
<td>*</td>
<td>1C</td>
<td>Highway Number Suffix</td>
</tr>
<tr>
<td>BEGMPST</td>
<td>*</td>
<td>3N</td>
<td>Beginning Milepost</td>
</tr>
<tr>
<td>BDISSIGN</td>
<td>*</td>
<td>1C</td>
<td>Beginning Displacement Sign</td>
</tr>
<tr>
<td>BMPSTDIS</td>
<td>*</td>
<td>2N</td>
<td>Beginning Milepost Displacement</td>
</tr>
<tr>
<td>LANESET</td>
<td>*</td>
<td>3C</td>
<td>Lane Set</td>
</tr>
<tr>
<td>LASTUPDT</td>
<td></td>
<td>6N</td>
<td>Last Update (YYMMDD)</td>
</tr>
<tr>
<td>SKIDNUMN</td>
<td></td>
<td>2N</td>
<td>Skid - Mean value</td>
</tr>
<tr>
<td>SKIDSTDD</td>
<td></td>
<td>2N</td>
<td>Skid - Standard Deviation</td>
</tr>
<tr>
<td>SKIDNUMH</td>
<td></td>
<td>2N</td>
<td>Skid - High Value</td>
</tr>
<tr>
<td>SKIDNUML</td>
<td></td>
<td>2N</td>
<td>Skid - Low Value</td>
</tr>
<tr>
<td>SKDCNTOB</td>
<td></td>
<td>2N</td>
<td>Skid - Count of Observation</td>
</tr>
</tbody>
</table>
### PES-Serviceability Index File Layout

**File Name:** PESRMN.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key Type</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HWYDIST</td>
<td>*</td>
<td>2N</td>
<td>SDHPT Highway District</td>
</tr>
<tr>
<td>CNTYNUM</td>
<td>*</td>
<td>2N</td>
<td>County Number</td>
</tr>
<tr>
<td>HNYPREFIX</td>
<td>*</td>
<td>2C</td>
<td>Highway Number Prefix</td>
</tr>
<tr>
<td>HNYNUM</td>
<td>*</td>
<td>4N</td>
<td>Highway Number</td>
</tr>
<tr>
<td>HNYSUFFIX</td>
<td>*</td>
<td>1C</td>
<td>Highway Number Suffix</td>
</tr>
<tr>
<td>BEGMPST</td>
<td>*</td>
<td>3N</td>
<td>Beginning Milepost</td>
</tr>
<tr>
<td>BDISSIGN</td>
<td>*</td>
<td>1C</td>
<td>Beginning Displacement Sign</td>
</tr>
<tr>
<td>BMPSTDIS</td>
<td>*</td>
<td>2N</td>
<td>Beginning Milepost Displacement</td>
</tr>
<tr>
<td>IANESET</td>
<td>*</td>
<td>3C</td>
<td>Lane Set</td>
</tr>
<tr>
<td>LASTUPDT</td>
<td></td>
<td>6N</td>
<td>Last Update (YMMDDD)</td>
</tr>
<tr>
<td>SIMEAN</td>
<td></td>
<td>7.5N</td>
<td>Serviceability Index - Mean</td>
</tr>
<tr>
<td>SISTDDEV</td>
<td></td>
<td>7.5N</td>
<td>Serviceability Index - Standard Deviation</td>
</tr>
<tr>
<td>SISHIVAL</td>
<td></td>
<td>3.1N</td>
<td>Serviceability Index - High Value</td>
</tr>
<tr>
<td>SLOWIVAL</td>
<td></td>
<td>3.1N</td>
<td>Serviceability Index - Low Value</td>
</tr>
<tr>
<td>SICOUNT</td>
<td></td>
<td>3N</td>
<td>Serviceability Index - Count of Observation</td>
</tr>
</tbody>
</table>
### PES-Falling Weight SSI File Layout

**File Name:** PESSSI.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HWYDIST</td>
<td>*</td>
<td>2N</td>
<td>SDHPT Highway District</td>
</tr>
<tr>
<td>CNTYNUM</td>
<td>*</td>
<td>2N</td>
<td>County Number</td>
</tr>
<tr>
<td>HWYPREFIX</td>
<td>*</td>
<td>2C</td>
<td>Highway Number Prefix</td>
</tr>
<tr>
<td>HWYNUM</td>
<td>*</td>
<td>4N</td>
<td>Highway Number</td>
</tr>
<tr>
<td>HWYSUFFIX</td>
<td>*</td>
<td>1C</td>
<td>Highway Number Suffix</td>
</tr>
<tr>
<td>BEGMPST</td>
<td>*</td>
<td>3N</td>
<td>Beginning Milepost</td>
</tr>
<tr>
<td>BDISSIGN</td>
<td>*</td>
<td>1C</td>
<td>Beginning Displacement Sign</td>
</tr>
<tr>
<td>BMPSTDIS</td>
<td>*</td>
<td>2N</td>
<td>Beginning Milepost Displacement</td>
</tr>
<tr>
<td>LANESET</td>
<td>*</td>
<td>3C</td>
<td>Lane Set</td>
</tr>
<tr>
<td>SSI_AVG</td>
<td></td>
<td>4.1N</td>
<td>Average SSI for the Roadway</td>
</tr>
<tr>
<td>SSITEMP</td>
<td></td>
<td>3N</td>
<td>Temperature For the Roadway</td>
</tr>
<tr>
<td>GEOPHON1</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 1 Geophone no. 1</td>
</tr>
<tr>
<td>GEOPHON2</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 2 Geophone no. 2</td>
</tr>
<tr>
<td>GEOPHON3</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 3 Geophone no. 3</td>
</tr>
<tr>
<td>GEOPHON4</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 4 Geophone no. 4</td>
</tr>
<tr>
<td>GEOPHON5</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 5 Geophone no. 5</td>
</tr>
<tr>
<td>GEOPHON6</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 6 Geophone no. 6</td>
</tr>
<tr>
<td>GEOPHON7</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 7 Geophone no. 7</td>
</tr>
<tr>
<td>GEOPHON8</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 8 Geophone no. 1</td>
</tr>
<tr>
<td>GEOPHON9</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 9 Geophone no. 2</td>
</tr>
<tr>
<td>GEOPHON10</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 10 Geophone no. 3</td>
</tr>
<tr>
<td>GEOPHON11</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 11 Geophone no. 4</td>
</tr>
<tr>
<td>GEOPHON12</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 12 Geophone no. 5</td>
</tr>
<tr>
<td>GEOPHON13</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 13 Geophone no. 6</td>
</tr>
<tr>
<td>GEOPHON14</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 14 Geophone no. 7</td>
</tr>
<tr>
<td>GEOPHON15</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 15 Geophone no. 1</td>
</tr>
<tr>
<td>GEOPHON16</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 16 Geophone no. 2</td>
</tr>
<tr>
<td>GEOPHON17</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 17 Geophone no. 3</td>
</tr>
<tr>
<td>GEOPHON18</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 18 Geophone no. 4</td>
</tr>
<tr>
<td>GEOPHON19</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 19 Geophone no. 5</td>
</tr>
<tr>
<td>GEOPHON20</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 20 Geophone no. 6</td>
</tr>
<tr>
<td>GEOPHON21</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 21 Geophone no. 7</td>
</tr>
<tr>
<td>GEOPHON22</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 22 Geophone no. 1</td>
</tr>
<tr>
<td>GEOPHON23</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 23 Geophone no. 2</td>
</tr>
<tr>
<td>GEOPHON24</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 24 Geophone no. 3</td>
</tr>
<tr>
<td>GEOPHON25</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 25 Geophone no. 4</td>
</tr>
<tr>
<td>GEOPHON26</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 26 Geophone no. 5</td>
</tr>
<tr>
<td>GEOPHON27</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 27 Geophone no. 6</td>
</tr>
<tr>
<td>GEOPHON28</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 28 Geophone no. 7</td>
</tr>
<tr>
<td>GEOPHON29</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 29 Geophone no. 1</td>
</tr>
<tr>
<td>GEOPHON30</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 30 Geophone no. 2</td>
</tr>
<tr>
<td>GEOPHON31</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 31 Geophone no. 3</td>
</tr>
<tr>
<td>GEOPHON32</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 32 Geophone no. 4</td>
</tr>
<tr>
<td>GEOPHON33</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 33 Geophone no. 5</td>
</tr>
<tr>
<td>GEOPHON34</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 34 Geophone no. 6</td>
</tr>
<tr>
<td>GEOPHON35</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 35 Geophone no. 7</td>
</tr>
<tr>
<td>GEOPHON36</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 36 Geophone no. 1</td>
</tr>
<tr>
<td>GEOPHON37</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 37 Geophone no. 2</td>
</tr>
<tr>
<td>GEOPHON38</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 38 Geophone no. 3</td>
</tr>
<tr>
<td>GEOPHON39</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 39 Geophone no. 4</td>
</tr>
<tr>
<td>GEOPHON40</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 40 Geophone no. 5</td>
</tr>
<tr>
<td>GEOPHON41</td>
<td></td>
<td>5.2N</td>
<td>SSI Reading 41 Geophone no. 1</td>
</tr>
</tbody>
</table>
## Pes-Visual File Layout

**File Name:** PESVISL.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HWYDIST</td>
<td>*</td>
<td>2N</td>
<td>SDHPT Highway District</td>
</tr>
<tr>
<td>CNTYNUM</td>
<td>*</td>
<td>2N</td>
<td>County Number</td>
</tr>
<tr>
<td>HNYPREFX</td>
<td>*</td>
<td>2C</td>
<td>Highway Number Prefix</td>
</tr>
<tr>
<td>HWYNUM</td>
<td>*</td>
<td>4N</td>
<td>Highway Number</td>
</tr>
<tr>
<td>HNYSUFX</td>
<td>*</td>
<td>1C</td>
<td>Highway Number Suffix</td>
</tr>
<tr>
<td>BEGMPLT</td>
<td>*</td>
<td>3N</td>
<td>Beginning Milepost</td>
</tr>
<tr>
<td>BDISSIGN</td>
<td>*</td>
<td>1C</td>
<td>Beginning Displacement Sign</td>
</tr>
<tr>
<td>BMPSTDIS</td>
<td>*</td>
<td>2N</td>
<td>Beginning Milepost Displacement</td>
</tr>
<tr>
<td>LANSET</td>
<td>*</td>
<td>3C</td>
<td>Lane Set</td>
</tr>
<tr>
<td>DATE</td>
<td></td>
<td>6N</td>
<td>Date (YMMDD)</td>
</tr>
<tr>
<td>FAILMILE</td>
<td></td>
<td>3N</td>
<td>Failures per Mile Occurance Rating</td>
</tr>
<tr>
<td>TRANCIR</td>
<td></td>
<td>3N</td>
<td>Transverse Crack Area Rating</td>
</tr>
<tr>
<td>LONCR</td>
<td></td>
<td>3N</td>
<td>Longitudinal Crack Area Rating</td>
</tr>
<tr>
<td>RUIT</td>
<td></td>
<td>3N</td>
<td>Rutting Area Rating</td>
</tr>
<tr>
<td>ALLGCR</td>
<td></td>
<td>3N</td>
<td>Alligator Crack Area Rating</td>
</tr>
<tr>
<td>BLKCR</td>
<td></td>
<td>3N</td>
<td>Block Crack Area Rating</td>
</tr>
<tr>
<td>PATCH</td>
<td></td>
<td>3N</td>
<td>Patching Area Rating</td>
</tr>
<tr>
<td>Field</td>
<td>Key</td>
<td>Size/Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
<td>-----------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>GEOPHON42</td>
<td>5.2N</td>
<td>SSI Reading 4</td>
<td>Geophone no. 2</td>
</tr>
<tr>
<td>GEOPHON43</td>
<td>5.2N</td>
<td>SSI Reading 4</td>
<td>Geophone no. 3</td>
</tr>
<tr>
<td>GEOPHON44</td>
<td>5.2N</td>
<td>SSI Reading 4</td>
<td>Geophone no. 4</td>
</tr>
<tr>
<td>GEOPHON45</td>
<td>5.2N</td>
<td>SSI Reading 4</td>
<td>Geophone no. 5</td>
</tr>
<tr>
<td>GEOPHON46</td>
<td>5.2N</td>
<td>SSI Reading 4</td>
<td>Geophone no. 6</td>
</tr>
<tr>
<td>GEOPHON47</td>
<td>5.2N</td>
<td>SSI Reading 4</td>
<td>Geophone no. 7</td>
</tr>
<tr>
<td>GEOPHON51</td>
<td>5.2N</td>
<td>SSI Reading 5</td>
<td>Geophone no. 1</td>
</tr>
<tr>
<td>GEOPHON52</td>
<td>5.2N</td>
<td>SSI Reading 5</td>
<td>Geophone no. 2</td>
</tr>
<tr>
<td>GEOPHON53</td>
<td>5.2N</td>
<td>SSI Reading 5</td>
<td>Geophone no. 3</td>
</tr>
<tr>
<td>GEOPHON54</td>
<td>5.2N</td>
<td>SSI Reading 5</td>
<td>Geophone no. 4</td>
</tr>
<tr>
<td>GEOPHON55</td>
<td>5.2N</td>
<td>SSI Reading 5</td>
<td>Geophone no. 5</td>
</tr>
<tr>
<td>GEOPHON56</td>
<td>5.2N</td>
<td>SSI Reading 5</td>
<td>Geophone no. 6</td>
</tr>
<tr>
<td>GEOPHON57</td>
<td>5.2N</td>
<td>SSI Reading 5</td>
<td>Geophone no. 7</td>
</tr>
</tbody>
</table>
## PES Update Program

**Score Information File Record Layout**

**File Name:** PESSCR.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HWYDIST</td>
<td>*</td>
<td>2N</td>
<td>Highway District Number</td>
</tr>
<tr>
<td>MAINTSEC</td>
<td>*</td>
<td>2N</td>
<td>Maintenance Section Number</td>
</tr>
<tr>
<td>CNTYNUM</td>
<td>*</td>
<td>3N</td>
<td>County Number</td>
</tr>
<tr>
<td>HWYPREFIX</td>
<td>*</td>
<td>2C</td>
<td>Highway System</td>
</tr>
<tr>
<td>HWYNUM</td>
<td>*</td>
<td>4N</td>
<td>Highway Number</td>
</tr>
<tr>
<td>HWYSUFFIX</td>
<td>*</td>
<td>1C</td>
<td>Highway Suffix</td>
</tr>
<tr>
<td>BEGMPOST</td>
<td>*</td>
<td>3N</td>
<td>Beginning Milepost Number</td>
</tr>
<tr>
<td>BDSSIGN</td>
<td>*</td>
<td>1C</td>
<td>Beginning Milepost Displacement Sign</td>
</tr>
<tr>
<td>BMDSTDIS</td>
<td>*</td>
<td>3.1N</td>
<td>Beginning Milepost Displacement Value</td>
</tr>
<tr>
<td>LANESET</td>
<td>*</td>
<td>3C</td>
<td>Laneset</td>
</tr>
<tr>
<td>SCRIDENT</td>
<td></td>
<td>3C</td>
<td>Identifier Score</td>
</tr>
<tr>
<td>SSIUTIL</td>
<td></td>
<td>4.2N</td>
<td>MRM Utility Score</td>
</tr>
<tr>
<td>SKIDUTIL</td>
<td></td>
<td>4.2N</td>
<td>Skid Utility Score</td>
</tr>
<tr>
<td>MAINTCST</td>
<td></td>
<td>4.2N</td>
<td>Maintenance Cost Score</td>
</tr>
<tr>
<td>UVURS</td>
<td></td>
<td>4.2N</td>
<td>Unweighted Visual Score</td>
</tr>
<tr>
<td>ADJVISL</td>
<td></td>
<td>4.2N</td>
<td>Adjusted Visual Score</td>
</tr>
<tr>
<td>WGTVISL</td>
<td></td>
<td>4.2N</td>
<td>Weighted Visual Score</td>
</tr>
<tr>
<td>UNWGTIPMT</td>
<td></td>
<td>4.2N</td>
<td>Unweighted Pavement Score</td>
</tr>
<tr>
<td>PVMTSCR</td>
<td></td>
<td>4.2N</td>
<td>Pavement Score</td>
</tr>
</tbody>
</table>

* Key Field
## File Name: PESGEN.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HWYDIST</td>
<td>*</td>
<td>2N</td>
<td>Highway District Number</td>
</tr>
<tr>
<td>MAINSEC</td>
<td>*</td>
<td>2N</td>
<td>Maintenance Section Number</td>
</tr>
<tr>
<td>CNTYNUM</td>
<td>*</td>
<td>3N</td>
<td>County Number</td>
</tr>
<tr>
<td>HWYPREFIX</td>
<td>*</td>
<td>2C</td>
<td>Highway System</td>
</tr>
<tr>
<td>HWYNUM</td>
<td>*</td>
<td>4N(C)</td>
<td>Highway Number</td>
</tr>
<tr>
<td>HWYSUFFIX</td>
<td>*</td>
<td>1C</td>
<td>Highway Suffix</td>
</tr>
<tr>
<td>BEGMPST</td>
<td>*</td>
<td>3N</td>
<td>Segment Begin - Milepost No</td>
</tr>
<tr>
<td>BDISSIGN</td>
<td>*</td>
<td>1C</td>
<td>- Displacement Sign</td>
</tr>
<tr>
<td>BDISDIS</td>
<td>*</td>
<td>3.1N</td>
<td>- Displacement Value</td>
</tr>
<tr>
<td>LANESET</td>
<td>*</td>
<td>3C</td>
<td>Laneset</td>
</tr>
<tr>
<td>ENDMPST</td>
<td>3N</td>
<td></td>
<td>Segment End - Milepost No.</td>
</tr>
<tr>
<td>EDISIGN</td>
<td>1C</td>
<td></td>
<td>- Displacement Sign</td>
</tr>
<tr>
<td>EMPSTDIS</td>
<td>3.1N</td>
<td></td>
<td>- Displacement Value</td>
</tr>
<tr>
<td>POST_1</td>
<td>3C/N</td>
<td></td>
<td>Intermediate Post - 1st Post</td>
</tr>
<tr>
<td>POST_2</td>
<td>3C/N</td>
<td></td>
<td>- 2nd Post</td>
</tr>
<tr>
<td>DISTT01</td>
<td>3.1N</td>
<td></td>
<td>- Distance to 1st</td>
</tr>
<tr>
<td>DISTT02</td>
<td>3.1N</td>
<td></td>
<td>- Distance to 2nd</td>
</tr>
<tr>
<td>BEGVAL</td>
<td>5.1N</td>
<td></td>
<td>Route-Thru-County - Begin Value</td>
</tr>
<tr>
<td>ENDVAL</td>
<td>5.1N</td>
<td></td>
<td>- End Value</td>
</tr>
<tr>
<td>SEGLENTH</td>
<td>3.1N</td>
<td></td>
<td>Segment Length</td>
</tr>
<tr>
<td>SEGLSUP</td>
<td>6N</td>
<td></td>
<td>Last Update Date (YYYYMMDD)</td>
</tr>
<tr>
<td>SEGMAINT</td>
<td>10.2N</td>
<td></td>
<td>Segment Maintenance Cost</td>
</tr>
<tr>
<td>AVGDTTRF</td>
<td>6N</td>
<td></td>
<td>Average Daily Traffic</td>
</tr>
<tr>
<td>KIP18EQ</td>
<td>5N</td>
<td></td>
<td>18-Kip Equivalency</td>
</tr>
<tr>
<td>FUNCAS</td>
<td>1N</td>
<td></td>
<td>Functional Classification</td>
</tr>
<tr>
<td>NOLANES</td>
<td>2N</td>
<td></td>
<td>Number of lanes</td>
</tr>
<tr>
<td>SPEEDLMT</td>
<td>2N</td>
<td></td>
<td>Speed Limit</td>
</tr>
<tr>
<td>RISURFTY</td>
<td>2N</td>
<td></td>
<td>RIS Surface Type</td>
</tr>
<tr>
<td>SEGSRWD</td>
<td>3N</td>
<td></td>
<td>Surface Width</td>
</tr>
<tr>
<td>HWYDSNTY</td>
<td>1N</td>
<td></td>
<td>Highway Design Type</td>
</tr>
<tr>
<td>SMPSELM</td>
<td>1C</td>
<td></td>
<td>Sample Selection Flag - Mandatory</td>
</tr>
<tr>
<td>SMPSELO</td>
<td>1C</td>
<td></td>
<td>- Other</td>
</tr>
</tbody>
</table>

* Key Field
Texas Flexible Pavement Database

PES Database Update Program
Missing and Inconsistent Data Report Layout

<table>
<thead>
<tr>
<th>SID Number</th>
<th>No PES Record</th>
<th>Number</th>
<th>Ending Lane</th>
<th>Functional Classification</th>
<th>No Surface Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXX</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Note: An 'X' is placed in the appropriate column(s) depending on whether or not there is a matching PES File record, or the Number of Lanes, the Ending Milepoint information, or the Functional Classification do not agree between the PES and Texas Flexible Pavement Database File.
CLEAR
* checks to see if all the files are present
? "Checking files . . ."
IF .NOT. FILE('\PAVEDB\FILES\PESSKD.DBF')
   ? "FES SKID FILE (PESSKD.DBF) not found. Please Check . . ."
   WAIT
   RETURN
ENDIF
IF .NOT. FILE('\PAVEDB\INDEXES\PESSKD.NDX')
   ? "FES SKID INDEX (PESSKD.NDX) not found. Please Check . . ."
   WAIT
   RETURN
ENDIF
IF .NOT. FILE('\PAVEDB\FILES\PESMRM.DBF')
   ? "FES SI FILE (PESMRM.DBF) not found. Please Check . . ."
   WAIT
   RETURN
ENDIF
IF .NOT. FILE('\PAVEDB\INDEXES\PESMRM.NDX')
   ? "FES SI INDEX (PESMRM.NDX) not found. Please Check . . ."
   WAIT
   RETURN
ENDIF
IF .NOT. FILE('\PAVEDB\FILES\PESSSI.DBF')
   ? "FES SSI FILE (PESSSI.DBF) not found. Please Check . . ."
   WAIT
   RETURN
ENDIF
IF .NOT. FILE('\PAVEDB\INDEXES\PESSSI.NDX')
   ? "FES SSI INDEX (PESSSI.NDX) not found. Please Check . . ."
   WAIT
   RETURN
ENDIF
IF .NOT. FILE('\PAVEDB\FILES\PESVISL.DBF')
   ? "FES VISUAL FILE (PESVISL.DBF) not found. Please Check . . ."
   WAIT
   RETURN
ENDIF
IF .NOT. FILE('\PAVEDB\INDEXES\PESVISL.NDX')

258
? "PES VISUAL INDEX (PESVISL.NDX) not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE("\PAVED\EDITUPDT\PES\PESTEMP1.DBF")
? "PES TEMPORARY FILE (PESTEMP1.DBF) not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE("\PAVED\EDITUPDT\PES\PESTEMP2.DBF")
? "PES TEMPORARY FILE (PESTEMP2.DBF) not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE("\PAVED\EDITUPDT\PES\PESTEMP3.DBF")
? "PES TEMPORARY FILE (PESTEMP3.DBF) not found. Please Check . . ."
WAIT
RETURN
ENDIF
IF .NOT. FILE("\PAVED\EDITUPDT\PES\PESTEMP4.DBF")
? "PES TEMPORARY FILE (PESTEMP4.DBF) not found. Please Check . . ."
WAIT
RETURN
ENDIF
@ 10, 10 SAY "Please wait. Setting up files . . ."
SET SAFETY OFF
USE \PAVED\FILES\PESSKD INDE \PAVED\INDEXES\PESSKD
ZAP
APPEND FROM \PAVED\EDITUPDT\PES\PESSKD.DLM DELIMITED
USE \PAVED\FILES\PESEGEN INDE \PAVED\INDEXES\PESEGEN
ZAP
APPEND FROM \PAVED\EDITUPDT\PES\PESEGEN.DLM DELIMITED
USE \PAVED\FILES\PESVISL INDE \PAVED\INDEXES\PESVISL
ZAP
APPEND FROM \PAVED\EDITUPDT\PES\PESVISL.DLM DELIMITED
USE \PAVED\FILES\PESSMFM INDE \PAVED\INDEXES\PESSMFM
ZAP
APPEND FROM \PAVED\EDITUPDT\PES\PESSMFM.DLM DELIMITED
USE \PAVED\FILES\PESSI INDE \PAVED\INDEXES\PESSI
ZAP
APPEND FROM \PAVED\EDITUPDT\PES\PESSI.DLM DELIMITED
USE \PAVED\FILES\PESSCR INDE \PAVED\INDEXES\PESSCR
ZAP
APPEND FROM \PAVED\EDITUPDT\PES\PESSCR.DLM DELIMITED
CLEAR
SET STAT ON
SET SAFETY OFF
@ 10, 10 SAY "Adding data to SKID file . . ."
* add data to the skid file
DO \PAVED\EDITUPDT\PES\PES_SKID

@ 12, 10 SAY "Adding data to VISUAL file . . ."
* add data to the visual file
DO \PAVEDB\EDITUPD\PES\PES_VISL

@ 14, 10 SAY "Adding data to FALLING WEIGHT file . . ."
* add data to the Falling Weight SSI file
DO \PAVEDB\EDITUPD\PES\PES_SSI

@ 16, 10 SAY "Adding data to SERVICEABILITY INDEX file . . ."
* add data to the Serviceability Index (SI) File
DO \PAVEDB\EDITUPD\PES\PES_MRM

* Print out Inconsistent Data Report
USE \PAVEDB\EDITUPD\PES\PESTEMP1
IF RECCOUNT() <> 0
   SORT ON S1D NO TO TEMP1
   USE TEMP1
   REPORT FORM \PAVEDB\REPORTS\PES_SKID TO PRINT
   EJECT
   USE
ENDIF
USE \PAVEDB\EDITUPD\PES\PESTEMP2
IF RECCOUNT() <> 0
   SORT ON S1D NO TO TEMP1
   USE TEMP1
   REPORT FORM \PAVEDB\REPORTS\PES_MRM TO PRINT
   EJECT
   USE
ENDIF
USE \PAVEDB\EDITUPD\PES\PESTEMP3
IF RECCOUNT() <> 0
   SORT ON S1D NO TO TEMP1
   USE TEMP1
   REPORT FORM \PAVEDB\REPORTS\PES_VISL TO PRINT
   EJECT
   USE TEMP1
ENDIF
USE \PAVEDB\EDITUPD\PES\PESTEMP4
IF RECCOUNT() <> 0
   SORT ON S1D NO TO TEMP1
   USE TEMP1
   REPORT FORM \PAVEDB\REPORTS\PES_SSI TO PRINT
   EJECT
   USE
ENDIF

SET STAT OFF
close database
RETURN
* SUBSYSTEM: EDIT & UPDATE SKID dBASE III FILE
* PROGRAM NAME: PES SKID.PRG  04/14/88
* MODIFIED ON: 09/19/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TIT
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: ADD THE PES DATA TO THE SKID FILE
*
* THE FOLLOWING FILES ARE USED BY THIS PROGRAM
* LOCATION.DBF
* LOCSTD .NDX
* PESTEMP1.DBF (A temporary PES file to print out the Missing )
* (and Inconsistent Data Report)
* SKID.DBF
* SKID.NDX
* LAYER .DBF
* LAYNDX .NDX
* PESSKD .DBF
* PESSKD .NDX
* PESGEN .DBF
* PESGEN .NDX
*
SELECT 1
USE \PAVEDB\FILES\LOCATION
SELE 2
USE \PAVEDB\FILES\PESSKD INDEX \PAVEDB\INDEXES\PESSKD
SELE 3
USE \PAVEDB\EDITUPDT\PES\PESTEMP1
DELE ALL
PACK
SELE 4
USE \PAVEDB\FILES\SKID INDEX \PAVEDB\INDEXES\SKID
SELE 5
USE \PAVEDB\FILES\PESGEN INDEX \PAVEDB\INDEXES\PESGEN
SELE 6
USE \PAVEDB\FILES\LAYER INDEX \PAVEDB\INDEXES\LAYNDX
MSID_ADDED = 0
SELE 1
DO WHILE .NOT. EOF()
* check if record is active
IF .NOT. ACTVFLAG
    SKIP
LOOP
ENDIF
* assign to memory variables
MSID   = SID NO
MHWYDIST = STR(HWYDIST,2)
MCNTYNUM = STR(CNTYNUM,3)
MHWYPREFX = HWYPREFX
MHWNUM  = STR(HWNUM,4)
MHWSUFFIX = HWWSUFFIX
MEBGMST  = STR(EBGMST,3)
MBDISSIGN = BDISSIGN
MBMPSIDIS = STR(MBMPSTDIS,2)
MLANEID  = LANEID
MENDMST  = ENDMST
MEDISSIGN = EDISSIGN
MBMPSIDIS = MBMPSIDIS
MPFOUND1 = .F.
MPFOUND2 = .F.
MERRORS = 0
MLANESET = 'R-L'

* find the record in PES SKID file
SELECT 5
seek MHWYDIST+MCNTYNUM+MHWYPREFX+MHWNUM+MHWSUFFIX+MEBGMST+MBDISSIGN+
+MBMPSIDIS+MLANESET
    IF FOUND()
        MPFOUND1 = .T.
    ELSE
        IF MLANEID = 'R'
            MLANESET = 'R-V'
        ELSE
            MLANESET = 'L-P'
        ENDIF
        seek MHWYDIST+MCNTYNUM+MHWYPREFX+MHWNUM+MHWSUFFIX+MEBGMST+MBDISSIGN+
+MBMPSIDIS+MLANESET
        IF FOUND()
            MPFOUND1 = .T.
        ENDF
    ENDF
SEEK 2
seek MHWYDIST+MCNTYNUM+MHWYPREFX+MHWNUM+MHWSUFFIX+MEBGMST+MBDISSIGN+
+MBMPSIDIS+MLANESET
    IF FOUND()
        MPFOUND2 = .T.
    ENDF
    IF MPFOUND1 .AND. MPFOUND2
        SEEK 5
        IF ENDMST <> MENDMST .OR. EDISSIGN <> MEDISSIGN .OR. MBMPSIDIS <>
        MBMPSIDIS
        SEEK 3
        IF MERRORS = 0
            APPEND BLANK
        ENDF
        REPLACE SID NO WITH MSID
        REPLACE END MST WITH 'X'
        MERRORS = MERRORS + 1
    SEEK 5
ENDIF
IF A->NUMLANES <> (NOLANES/2)
    SELE 3
    IF MERRORS = 0
        APPEND BLANK
    ENDF
    REPLACE SID NO WITH MSID
    REPLACE NO IANES WITH 'X'
    MERRORS = MERRORS + 1
    SELE 5
ENDIF
IF A->FUCNAS = 0
    REPL A->FUCNAS WITH FUCNAS
ELSE
    IF A->FUCNAS <> FUCNAS
        IF MERRORS = 0
            APPEND BLANK
        ENDF
        REPLACE SID NO WITH MSID
        REPLACE NO FUCCL WITH 'X'
        MERRORS = MERRORS + 1
    ENDF
ENDIF
SELE 2
IF LASTUPDT = 0
    * Adds to inconsistent report because key is 0
    SELE 3
    IF MERRORS = 0
        APPEND BLANK
    ENDF
    REPLACE SID NO WITH MSID
    REPLACE NO FES WITH 'X'
    REPLACE NO FUCCL WITH ' '
    REPLACE NO IANES WITH ' '
    REPLACE END MFS T WITH ' '
    MERRORS = MERRORS + 1
ENDIF
ELSE
    SELE 3
    APPEND BLANK
    REPLACE SID NO WITH MSID
    REPLACE NO FES WITH 'X'
ENDIF
IF MERRORS = 0 .AND. MFOUND1 .AND. MFOUND2
    * get the structure number and the layer number for the record
    * Assign to memory variables
SELE 2
    MDATE = STR(LASTUPDT,6)
    MYEAR = VAL(LEFT(MDATE,2))
    MMONTH = VAL(SUBSTR(MDATE,3,2))
SELECT 6
    * find the sid # in layer file

seek str(MSID,4)
IF FOUND()
    DO WHILE MSID = SID_NO
        SKIP
    ENDDO
    SKIP -1
    * find the layer # which corresponds to the skid year
    do while jobcmpyr > myear
        skip -1
    enddo
    * if the layer year and skid year are the same, need to
    * check the month to get the right layer number
    DO WHILE JOBCMPIYR = MYEAR .AND. JOBCMIMO > MMONIH
        SKIP -1
        IF SID NO <> MSID
            SKIP
            ERRORS = ERRORS + 1
            EXIT
        ENDF
    ENDDO
* check to avoid if layer is a base, subbase or subgrade
* if not, then replace the structure and layer # from layer file
IF LAYDESC=5 .OR. LAYDESC=6 .OR. LAYDESC=7 .OR. LAYDESC=11
    * Print out on report because layer is a base
    SELE 3
    APPEND BLANK
    REPLACE SID NO WITH MSID
    REPLACE NO SURF WITH 'X'
    ERRORS = ERRORS + 1
ELSE
    MSTRUCNUM = F->STRUCNUM
    MLAYNUM = F->LAYNUM
ENDIF
ELSE
    SELE 3
    APPEND BLANK
    REPLACE SID NO WITH MSID
    REPLACE NO SURF WITH 'X'
    ERRORS = ERRORS + 1
ENDIF
IF ERRORS = 0
    SELE 4
    * check to see if data is already present
    SEEK
    STR(MSID,4)+STR(MSTRUCNUM,2)+STR(MLAYNUM,2)+STR(MYEAR,2)+STR(MMONIH,2)
    IF FOUND()
        CLEAR
        ? "This program has already been run previously. "
        ? "Terminating program and Returning to main menu"
        ? ""
        ? ""
        ? " "
        WAIT
ENDIF
CLOSE ALL
RETURN TO MASTER
ENDIF
* add the data to the skid file
MSID_ADDED = MSID_ADDED + 1
APPEND BLANK
REPLACE SID NO WITH MSID
REPLACE STRUCNUM WITH MSTRUCNUM
REPLACE LAYNUM WITH MLAYNUM
REPLACE YEAR WITH MYEAR
REPLACE MONTH WITH MMONTH
REPLACE SKIDNUMM WITH B->SKIDNUMN
REPLACE SKIDNUMH WITH B->SKIDNUMH
REPLACE SKIDNUML WITH B->SKIDNUML
ENDIF
ENDIF
SELECT 1
SKIP
STORE 0 TO MSTRUCNUM, MLAYNUM, MSID, MYEAR, MMONTH
ENDDO

close database
RETURN
**Edit & Update Subsystem**  
**Program Listing**

*  
**SUBSYSTEM:** EDIT & UPDATE VISUAL FILE  
**PROGRAM NAME:** PES VISIL.PRG  
**PROJECT 2456:** TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION  
**TAMU/TITI**  
**AUTHOR:** TREVOR X. PEREIRA  
**PURPOSE:** ADD THE PES DATA TO THE VISUAL FILE  
*  
**THE FOLLOWING FILES ARE USED BY THIS PROGRAM**  
* LOCATION.DBF  
* LOCID .NDX  
* PESTEMP3.DBF (A temporary PES file to print out the Missing)  
* (and Inconsistent Data Report for VISUAL file)  
* VISUAL .DBF  
* VISUAL .NDX  
* LAYER .DBF  
* LAYNDX .NDX  
* PESVISL .DBF  
* PESVISL .NDX  
* PESSCR .DBF  
* PESSCR .NDX  
*  
**SELECT 1**  
USE \PAVDB\FILES\LOCATION  
**SELE 2**  
USE \PAVDB\FILES\PESVISL INDEX \PAVDB\INDEXES\PESVISL  
**SELE 3**  
USE \PAVDB\EDITUPDT\PES\PESTEMP3  
**DELE ALL**  
**PACK**  
**SELE 5**  
USE \PAVDB\FILES\PESEG INDE \PAVDB\INDEXES\PESEG  
**SELE 6**  
USE \PAVDB\FILES\LAYER INDEX \PAVDB\INDEXES\LAYNDX  
**SELE 7**  
USE \PAVDB\FILES\PESSCR INDEX \PAVDB\INDEXES\PESSCR  

**MSID_ADDED = 0**  
**SELE 1**  
DO WHILE .NOT. EOF()  
  * check if record is active  
  IF .NOT. ACTVFLAG  
    **SKIP**  
  **LOOP**  
  **ENDIF**  
  * assign to memory variables  
  **MSID = SID_NO**  
  **MHWDIST = STR(HWYDIST,2)**  
  **MCNTYNUM = STR(CNTYNUM,3)**
MHWYPREFIX = HWYPREFIX
MHWYNUM = STIR(HWYNUM,4)
MHWYSUFFIX = HWYSUFFIX
MBEGMPST = STIR(BEGMPST,3)
MBDISSIGN = BDISSIGN
MEMPSTIDIS = STIR(EMPSTIDIS,2)
MLANEID = LANEID
MENDMPST = ENDMpst
MEDISSIGN = EDISSIGN
MEMPSTIDIS = EMPSTIDIS
MFOUND1 = .F.
MFOUND2 = .F.
MFOUND3 = .F.
MERRORS = 0
MLANESet = 'R-L'

* find the record in PES SI file
SELE 2
  seek MHWYDIST+MCNTYNUM+MHWYPREFIX+MHWYNUM+MHYSUFFIX+MBEGMPST+MBDISSIGN+
+MEMPSTIDIS+MLANESet
  IF FOUND()
    MFOUND1 = .T.
  ELSE
    IF MLANEID = 'R'
      MLANESet = 'R-V'
    ELSE
      MLANESet = 'L-P'
    ENDIF
    seek MHWYDIST+MCNTYNUM+MHWYPREFIX+MHWYNUM+MHYSUFFIX+MBEGMPST+MBDISSIGN+
+MEMPSTIDIS+MLANESet
    IF FOUND()
      MFOUND1 = .T.
    ENDIF
  ENDIF
SELE 7
  seek MHWYDIST+MCNTYNUM+MHWYPREFIX+MHWYNUM+MHYSUFFIX+MBEGMPST+MBDISSIGN+
+MEMPSTIDIS+MLANESet
  IF FOUND()
    MFOUND2 = .T.
  ENDIF
SELE 5
  seek MHWYDIST+MCNTYNUM+MHWYPREFIX+MHWYNUM+MHYSUFFIX+MBEGMPST+MBDISSIGN+
+MEMPSTIDIS+MLANESet
  IF FOUND()
    MFOUND3 = .T.
  ENDIF

IF MFOUND1 .AND. MFOUND2 .AND. MFOUND3
  SELE 5
    IF ENDMpST <> MENDMPST .OR. EDISSIGN <> MEDISSIGN .OR. MEMPSTIDIS <>
    EMPSTIDIS
      SELE 3
        IF MERRORS = 0
          APPEND BLANK

267
ENDIF
REPLACE SID NO WITH MSID
REPLACE END MPST WITH 'X'
MERRORS = MERRORS + 1
SELE 5
ENDIF
IF A->NUMLANES <> (NOLANES/2)
SELE 3
IF MERRORS = 0
APPEND BLANK
ENDIF
REPLACE SID NO WITH MSID
REPLACE NO_LANES WITH 'X'
MERRORS = MERRORS + 1
SELE 5
ENDIF
IF A->FUNCLAS = 0
REPL A->FUNCLAS WITH FUNCLAS
ELSE
IF A->FUNCLAS <> FUNCLAS
IF MERRORS = 0
APPEND BLANK
ENDIF
REPLACE SID NO WITH MSID
REPLACE NO_FUNCL WITH 'X'
MERRORS = MERRORS + 1
ENDIF
ENDIF
SELE 2
IF DATE = 0
* Adds to inconsistent report because key is 0
SELE 3
IF MERRORS = 0
APPEND BLANK
ENDIF
REPLACE SID NO WITH MSID
REPLACE NO_FES WITH 'X'
REPLACE NO_FUNCL WITH ' ':
REPLACE NO_LANES WITH ' ':
REPLACE END MPST WITH ' '
MERRORS = MERRORS + 1
ENDIF
ELSE
SELE 3
APPEND BLANK
REPLACE SID NO WITH MSID
REPLACE NO_FES WITH 'X'
ENDIF
IF MFOUND1 .AND. MFOUND2 .AND. MFOUND3 .AND. MFOUND4 .AND. MERRORS = 0
* get the structure number and the layer number for the record
* Assign to memory variables
SELE 2
MDATE = STR(DATE,6)
MYEAR = VAL(LEFT(MDATE,2))
MMONIH = VAL(SUBSTR(MDATE, 3, 2))
MDAY = VAL(SUBSTR(MDATE, 5, 2))

SELECT 6
* find the sid # in layer file
seek STR(MSID, 4)
IF FOUND()
    DO WHILE MSID = SID_NO
        SKIP
    ENDDO
    SKIP -1
    * find the layer # which corresponds to the skid year
    do while jobcompyr > myear
        skip -1
    enddo
* if the layer year and skid year are the same, need to
* check the month to get the right layer number
    DO WHILE JOBCOMPYR = MYEAR .AND. JOBCOMPMO > MMONIH
        SKIP -1
        IF SID NO <> MSID
            SKIP
            MERRORS = MERRORS + 1
        EXIT
    ENDDO
* check to avoid if layer is a base, subbase or subgrade
* if not, then replace the structure and layer # from layer file
    IF LAYDESC=5 .OR. LAYDESC=6 .OR. LAYDESC=7 .OR. LAYDESC=11
        SELE 3
        APPEND BLANK
        REPLACE SID NO WITH MSID
        REPLACE NO SURF WITH 'X'
        MERRORS = MERRORS + 1
    ELSE
        MSNUCNUM = F->STRUCTNUM
        MLAYNUM = F->LAYNUM
    ENDDO
ELSE
    SELE 3
    APPEND BLANK
    REPLACE SID NO WITH MSID
    REPLACE NO SURF WITH 'X'
    MERRORS = MERRORS + 1
ENDIF
IF MERRORS = 0
    * add the data to the skid file
    SELE 5
    USE
    SELE 6
    USE
    SELE 4
    USE \PAVEDB\FILES\VISUAL INDEX \PAVEDB\INDEXES\VISUAL
    APPEND BLANK
    REPLACE SID NO WITH MSID
REPLACE STRUCNUM WITH MSTRUCNUM
REPLACE LAYNUM WITH MLAYNUM
REPLACE YEAR WITH MYEAR
REPLACE ACTYEAR WITH MYEAR
REPLACE ACIMONTH WITH MMONTH
DO CASE
  CASE $B>RUITT = 000$
    REPL RUTTSGL WITH 0
    REPL RUTTIMD WITH 0
    REPL RUTTSV WITH 0
  CASE $B>RUITT = 100$
    REPL RUTTSGL WITH 0
    REPL RUTTIMD WITH 1
    REPL RUTTSV WITH 0
  CASE $B>RUITT = 010$
    REPL RUTTSGL WITH 0
    REPL RUTTIMD WITH 2
    REPL RUTTSV WITH 0
  CASE $B>RUITT = 001$
    REPL RUTTSGL WITH 0
    REPL RUTTIMD WITH 3
    REPL RUTTSV WITH 0
  CASE $B>RUITT = 200$
    REPL RUTTSGL WITH 0
    REPL RUTTIMD WITH 0
    REPL RUTTSV WITH 1
  CASE $B>RUITT = 020$
    REPL RUTTSGL WITH 0
    REPL RUTTIMD WITH 0
    REPL RUTTSV WITH 2
  CASE $B>RUITT = 002$
    REPL RUTTSGL WITH 0
    REPL RUTTIMD WITH 0
    REPL RUTTSV WITH 3
ENDCASE
DO CASE
  CASE $B>BLKCR = 000$
    REPL BLKCRKSL WITH 0
    REPL BLKCRKMD WITH 0
    REPL BLKCRKSV WITH 0
  CASE $B>BLKCR = 100$
    REPL BLKCRKSL WITH 0
    REPL BLKCRKMD WITH 1
    REPL BLKCRKSV WITH 0
  CASE $B>BLKCR = 010$
    REPL BLKCRKSL WITH 0
    REPL BLKCRKMD WITH 2
    REPL BLKCRKSV WITH 0
  CASE $B>BLKCR = 001$
    REPL BLKCRKSL WITH 0
    REPL BLKCRKMD WITH 3
    REPL BLKCRKSV WITH 0
ENDCASE
DO CASE
CASE B->ALLGCR = 000
  REPL ALLGCRSL WITH 0
  REPL ALLGCRMD WITH 0
  REPL ALLGCRSV WITH 0
CASE B->ALLGCR = 100
  REPL ALLGCRSL WITH 0
  REPL ALLGCRMD WITH 1
  REPL ALLGCRSV WITH 0
CASE B->ALLGCR = 010
  REPL ALLGCRSL WITH 0
  REPL ALLGCRMD WITH 2
  REPL ALLGCRSV WITH 0
CASE B->ALLGCR = 001
  REPL ALLGCRSL WITH 0
  REPL ALLGCRMD WITH 3
  REPL ALLGCRSV WITH 0
ENDCASE
DO CASE
CASE B->LONGCR = 000
  REPL LONGCRSL WITH 0
  REPL LONGCRMD WITH 0
  REPL LONGCRSV WITH 0
CASE B->LONGCR = 100
  REPL LONGCRSL WITH 0
  REPL LONGCRMD WITH 1
  REPL LONGCRSV WITH 0
CASE B->LONGCR = 010
  REPL LONGCRSL WITH 0
  REPL LONGCRMD WITH 2
  REPL LONGCRSV WITH 0
CASE B->LONGCR = 001
  REPL LONGCRSL WITH 0
  REPL LONGCRMD WITH 3
  REPL LONGCRSV WITH 0
ENDCASE
DO CASE
CASE B->TRANCR = 000
  REPL TRANCRSL WITH 0
  REPL TRANCRMD WITH 0
  REPL TRANCRSV WITH 0
CASE B->TRANCR = 100
  REPL TRANCRSL WITH 0
  REPL TRANCRMD WITH 1
  REPL TRANCRSV WITH 0
CASE B->TRANCR = 010
  REPL TRANCRSL WITH 0
  REPL TRANCRMD WITH 2
  REPL TRANCRSV WITH 0
CASE B->TRANCR = 001
  REPL TRANCRSL WITH 0
  REPL TRANCRMD WITH 3
  REPL TRANCRSV WITH 0
ENDCASE
DO CASE
CASE B->PATCH = 000
    REPL PATCHGD WITH 0
    REPL PATCHFR WITH 0
CASE B->PATCH = 100
    REPL PATCHGD WITH 0
    REPL PATCHFR WITH 1
    REPL PATCHPR WITH 0
CASE B->PATCH = 010
    REPL PATCHGD WITH 0
    REPL PATCHFR WITH 2
    REPL PATCHPR WITH 0
CASE B->PATCH = 001
    REPL PATCHGD WITH 0
    REPL PATCHFR WITH 3
    REPL PATCHPR WITH 0
ENDCASE
DO CASE
    CASE B->FAILMILE = 000
        REPL FAILMILE WITH 0
    CASE B->FAILMILE = 100
        REPL FAILMILE WITH 1
    CASE B->FAILMILE = 010
        REPL FAILMILE WITH 2
    CASE B->FAILMILE = 001
        REPL FAILMILE WITH 3
ENDCASE
REPL PESPVIRS WITH G->PVMISCR
REPL UVURS WITH G->UNWGHIVL
    sele 4
    USE
    sele 5
    USE \PAVEDB\FILES\PESGEN INDE \PAVEDB\INDEXES\PESGEN
    sele 6
    USE \PAVEDB\FILES\LAYER INDEX \PAVEDB\INDEXES\LAYNDX
ENDIF
ELSE
    sele 3
    APPEND BLANK
    REPL SID NO WITH MSID
    REPL NO_PES WITH 'X'
ENDIF
    sele 1
    SKIP
ENDDO
CLOSE DATABASES
RETURN
* SUBSYSTEM:  EDIT & UPDATE FALLING WEIGHT SSI FILE
* PROGRAM NAME:  PES_SSI.PRG  05/16/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR:  TREVOR X. PEREIRA
* PURPOSE:  ADD THE PES DATA TO THE SSI FILE
*
* THE FOLLOWING FILES ARE USED BY THIS PROGRAM
* LOCATION.DBF
* LOCRTID .NDX
* PESTEMP4.DBF (A temporary PES file to print out the Missing )
* (and Inconsistent Data Report for SI file)
* FALLWGT.DBF
* FALLWGT.NDX
* LAYER .DBF
* LAYNDX .NDX
* PESSI .DBF
* PESSI .NDX
*
SELECT 1
USE \PAVEDB\FILES\LOCATION
SELE 2
USE \PAVEDB\FILES\PESSI INDEX \PAVEDB\INDEXES\PESSI
SELE 3
USE \PAVEDB\EDITUPDT\PES\PESTEMP4
DELE ALL
PACK
SELE 4
USE \PAVEDB\FILES\FALLWGT INDEX \PAVEDB\INDEXES\FALLWGT
SELE 5
USE \PAVEDB\FILES\PESSGEN INDE \PAVEDB\INDEXES\PESSGEN
SELE 6
USE \PAVEDB\FILES\LAYER INDEX \PAVEDB\INDEXES\LAYNDX

MSID_ADDED = 0
*
SELE 1
DO WHILE .NOT. EOF()
  * check if record is active
  IF .NOT. ACTVFLAG
    SKIP
  LOOP
ENDIF
  * assign to memory variables
  MSID   = SID NO
  MHWYDIST = STR(HWYDIST,2)
  MONTYNUM = STR(CNTYNUM,3)
MHWYPREFIX = HWYPREFIX
MHWYNUM = STR(HWYNUM,4)
MHWYSUFFIX = HWYSUFFIX
MBEGMST = STR(BEGMST,3)
MBDISSIGN = EDISSIGN
MEMPSIDIS = STR(MEMPSIDIS,2)
MLANEID = LANEID
MENDMST = ENDMST
MEDISSIGN = EDISSIGN
MEMPSIDIS = EMPSTDIS
MFIND1 = .F.
MFIND2 = .F.
MERRORS = 0
MLANESET = 'R-L'

* find the record in PES SSI file
SELECT 5
  seek MHWYDIST+MCNTYNUM+MHWYPREFIX+HWYNUM+HWYSUFFIX+MBEGMST+MBDISSIGN+
+MEMPSIDIS+MLANESET
  IF FOUND()
    MFIND1 = .T.
  ELSE
    IF MLANEID = 'R'
      MLANESET = 'R-V'
    ELSE
      MLANESET = 'L-P'
    ENDIF
    seek MHWYDIST+MCNTYNUM+MHWYPREFIX+HWYNUM+HWYSUFFIX+MBEGMST+MBDISSIGN+
+MEMPSIDIS+MLANESET
    IF FOUND()
      MFIND1 = .T.
    ENDIF
  ENDDIF
SELE 2
  seek MHWYDIST+MCNTYNUM+MHWYPREFIX+HWYNUM+HWYSUFFIX+MBEGMST+MBDISSIGN+
+MEMPSIDIS+MLANESET
  IF FOUND()
    MFIND2 = .T.
  ENDIF
IF MFIND1 .AND. MFIND2
SELE 5
  IF ENDMST <> MENDMST .OR. EDISSIGN <> MEDISSIGN .OR. MEMPSIDIS <>
EMPSTDIS
    SELE 3
    IF MERRORS = 0
      APPEND BLANK
    ENDIF
    REPLACE STD_NO WITH MSID
    REPLACE END_MPST WITH 'X'
    MERRORS = MERRORS + 1
    SELE 5
  ENDDIF
  IF A->NUMLANES <> (NOLANES/2)
    SELE 3
IF ERRORS = 0
   APPEND BLANK
ENDIF
REPLACE SID NO WITH MSID
REPLACE NO IANES WITH 'X'
ERRORS = ERRORS + 1
SELE 5
ENDIF
IF A->FUNCLAS = 0
   REPL A->FUNCLAS WITH FUNCLAS
ELSE
   IF A->FUNCLAS <> FUNCLAS
      IF ERRORS = 0
         APPEND BLANK
      ENDIF
      REPLACE SID NO WITH MSID
      REPLACE NO FUNCL WITH 'X'
      ERRORS = ERRORS + 1
     ENDIF
ENDIF
SELE 2
IF SSIDATE = 0
   * Adds to inconsistent report because key is 0
   SELE 3
   IF ERRORS = 0
      APPEND BLANK
   ENDIF
   REPLACE SID NO WITH MSID
   REPLACE NO PES WITH 'X'
   REPLACE NO FUNCL WITH ' '
   REPLACE NO IANES WITH ' '
   REPLACE END MPST WITH ' '
   ERRORS = ERRORS + 1
ENDIF
ELSE
   SELE 3
   APPEND BLANK
   REPLACE SID NO WITH MSID
   REPLACE NO PES WITH 'X'
ENDIF
IF MFOUND1 .AND. MFOUND2 .AND. ERRORS = 0
   * get the structure number and the layer number for the record
   * Assign to memory variables
   SELE 2
   MDATE = STR(SSIDATE,6)
   MYEAR = VAL(LEFT(MDATE,2))
   MMMONTH = VAL(SUBSTR(MDATE,3,2))
   MDAY = VAL(SUBSTR(MDATE,5,2))
SELECT 6
   * find the sid # in layer file
   seek str(MSID,4)
   IF FOUND()
DO WHILE MSID = SID_NO
  SKIP
ENDDO
SKIP -1
* find the layer # which corresponds to the skid year
   do while jobcmyr > myear
     skip -1
   enddo
* if the layer year and skid year are the same, need to
* check the month to get the right layer number
DO WHILE JOBCMYR = MYEAR .AND. JOBCMNYO > MMONTH
  SKIP -1
  IF SID NO <> MSID
     SKIP
     MERRORS = MERRORS + 1
  EXIT
  ENDF
ENDDO
* check to avoid if layer is a base, subbase or subgrade
* if not, then replace the structure and layer # from layer file
IF LAYDESC=5 .OR. LAYDESC=6 .OR. LAYDESC=7 .OR. LAYDESC=11
  SELE 3
  APPEND BLANK
  REPLACE SID NO WITH MSID
  REPLACE NO SURF WITH 'X'
  MERRORS = MERRORS + 1
ELSE
  MSTRUCNUM = F->STRUCNUM
  MLAYNUM    = F->LAYNUM
ENDIF
ELSE
  SELE 3
  APPEND BLANK
  REPLACE SID NO WITH MSID
  REPLACE NO SURF WITH 'X'
  MERRORS = MERRORS + 1
ENDIF
IF MERRORS = 0
* add the data to the skid file
MSID ADDED = MSID_ADDED + 1
SELE 4
APPEND BLANK
REPLACE SID NO WITH MSID
REPLACE STRUCNUM WITH MSTRUCNUM
REPLACE LAYNUM WITH MLAYNUM
REPLACE YEAR WITH MYEAR
REPLACE MONTH WITH MMONTH
REPLACE DAY WITH MDAY
REPL RSSTIAVG WITH B->SSI AVG
REPL SSITEMP WITH B->SSITEMP
* Lane 1
REPL SSIGP11 WITH B->GEOPHON11
REPL SSIGP12 WITH B->GEOPHON12

276
REPL SSIGP13 WITH B->GEOPHON13
REPL SSIGP14 WITH B->GEOPHON14
REPL SSIGP15 WITH B->GEOPHON15
REPL SSIGP16 WITH B->GEOPHON16
REPL SSIGP17 WITH B->GEOPHON17
* Lane 2
REPL SSIGP21 WITH B->GEOPHON21
REPL SSIGP22 WITH B->GEOPHON22
REPL SSIGP23 WITH B->GEOPHON23
REPL SSIGP24 WITH B->GEOPHON24
REPL SSIGP25 WITH B->GEOPHON25
REPL SSIGP26 WITH B->GEOPHON26
REPL SSIGP27 WITH B->GEOPHON27
* Lane 3
REPL SSIGP31 WITH B->GEOPHON31
REPL SSIGP32 WITH B->GEOPHON32
REPL SSIGP33 WITH B->GEOPHON33
REPL SSIGP34 WITH B->GEOPHON34
REPL SSIGP35 WITH B->GEOPHON35
REPL SSIGP36 WITH B->GEOPHON36
REPL SSIGP37 WITH B->GEOPHON37
* Lane 4
REPL SSIGP41 WITH B->GEOPHON41
REPL SSIGP42 WITH B->GEOPHON42
REPL SSIGP43 WITH B->GEOPHON43
REPL SSIGP44 WITH B->GEOPHON44
REPL SSIGP45 WITH B->GEOPHON45
REPL SSIGP46 WITH B->GEOPHON46
REPL SSIGP47 WITH B->GEOPHON47
* Lane 5
REPL SSIGP51 WITH B->GEOPHON51
REPL SSIGP52 WITH B->GEOPHON52
REPL SSIGP53 WITH B->GEOPHON53
REPL SSIGP54 WITH B->GEOPHON54
REPL SSIGP55 WITH B->GEOPHON55
REPL SSIGP56 WITH B->GEOPHON56
REPL SSIGP57 WITH B->GEOPHON57
ENDIF
ELSE
SELE 3
APPEND BLANK
REPL SID_NO WITH MSID
REPL NO_PES WITH 'X'
ENDIF
SELE 1
SKIP
ENDDO
CLOSE DATABASE
RETURN
* SUBSYSTEM: EDIT & UPDATE SERVICEABILITY INDEX FILE
* PROGRAM NAME: PES_MRM.PRG 05/13/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: ADD THE PES DATA TO THE SI FILE
*
* THE FOLLOWING FILES ARE USED BY THIS PROGRAM
* LOCATION.DBF
* LOCISID .NDX
* PESTEMP2.DBF (A temporary PES file to print out the Missing )
* (and Inconsistent Data Report for SI file)
* SI .DBF
* SI .NDX
* LAYER .DBF
* LAYNDX .NDX
* PESMRM .DBF
* PESMRM .NDX
*
SELECT 1
USE \PAVEDB\FILES\LOCATION
SELE 2
USE \PAVEDB\FILES\PESMRM INDEX \PAVEDB\INDEXES\PESMRM
SELE 3
USE \PAVEDB\EDITUPDT\PES\PESTEMP2
DELE ALL
PACK
SELE 4
USE \PAVEDB\FILES\SI INDEX \PAVEDB\INDEXES\SI
SELE 5
USE \PAVEDB\FILES\PESGEN INDE \PAVEDB\INDEXES\PESGEN
sele 6
USE \PAVEDB\FILES\LAYER INDEX \PAVEDB\INDEXES\LAYNDX
MSID_ADDED = 0

SELE 1
DO WHILE .NOT. EOF()
* check if record is active
IF .NOT. ACTVFLAG
    SKIP
ENDIF
* assign to memory variables
MSID = SID_NO
MHWDIST = STR(MHWDIST,2)
MCNTYNUM = STR(MCNTYNUM,3)
MHWPREFIX = HWPREFIX
MHWNUM = STR(HWNUM, 4)
MHWSUFX = HWYSUFX
MBEGMPST = STR(BEGMPST, 3)
MBDISSIGN = EDISSIGN
MEMPSIDIS = STR(MEMPSIDIS, 2)
MLANEID = LANEID
MENDMPST = ENDMFST
MEDISSIGN = EDISSIGN
MEMPSIDIS = MEMPSIDIS
MFOUND1 = .F.
MFOUND2 = .F.
MERRORS = 0
MLANESET = 'R-L'

* find the record in PES SI file
SELECT 5
seek MHWDIST+MCNTNUM+MHYPREFX+MHWNUM+MHYSUFX+MBEGMPST+MBDISSIGN+
+MEMPSIDIS+MLANESET
IF FOUND()
  MFOUND1 = .T.
ELSE
  IF MLANEID = 'R'
    MLANESET = 'R-V'
  ELSE
    MLANESET = 'L-P'
  ENDIF
seek MHWDIST+MCNTNUM+MHYPREFX+MHWNUM+MHYSUFX+MBEGMPST+MBDISSIGN+
+MEMPSIDIS+MLANESET
IF FOUND()
  MFOUND1 = .T.
ENDIF
ENDIF
SELE 2
seek MHWDIST+MCNTNUM+MHYPREFX+MHWNUM+MHYSUFX+MBEGMPST+MBDISSIGN+
+MEMPSIDIS+MLANESET
IF FOUND()
  MFOUND2 = .T.
ENDIF
IF MFOUND1 .AND. MFOUND2
SELE 5
  IF ENDMFST <> MENDMPST .OR. EDISSIGN <> MEDISSIGN .OR. MEMPSIDIS <>
  MEMPSIDIS
    SELE 3
    IF MERRORS = 0
      APPEND BLANK
    ENDFD
    REPLACE SID NO WITH MSID
    REPLACE END MFST WITH 'X'
    MERRORS = MERRORS + 1
    SELE 5
  ENDIF
  IF A->NUMLANES <> (NOLANES/2)
    SELE 3
    IF MERRORS = 0

APPEND BLANK
ENDIF
REPLACE SID NO WITH MSID
REPLACE NO_LANES WITH 'X'
MERRORS = MERRORS + 1
SELE 5
ENDIF
IF A->FUNCLAS = 0
  REPL A->FUNCLAS WITH FUNCLAS
ELSE
  IF A->FUNCLAS <> FUNCLAS
    IF MERRORS = 0
      APPEND BLANK
    ENDF
    REPLACE SID NO WITH MSID
    REPLACE NO_LANES WITH 'X'
    MERRORS = MERRORS + 1
  ENDF
ENDIF
SELE 2
IF LASTUPDT = 0
  * Adds to inconsistent report because key is 0
  SELE 3
  IF MERRORS = 0
    APPEND BLANK
  ENDF
  REPLACE SID NO WITH MSID
  REPLACE NO_LANES WITH 'X'
  REPLACE NO_FUNCL WITH ' '
  REPLACE NO_LANES WITH ' '
  REPLACE END_MPST WITH ' '
  MERRORS = MERRORS + 1
ENDIF
ELSE
  SELE 3
  APPEND BLANK
  REPLACE SID NO WITH MSID
  REPLACE NO_LANES WITH 'X'
ENDIF
IF MFOUND1 .AND. MFOUND2 .AND. MERRORS = 0
  * get the structure number and the layer number for the record
  * Assign to memory variables
  SELE 2
  MDATE = STR(LASTUPDT,6)
  MYEAR = VAL(LEFT(MDATE,2))
  MMONTH = VAL(SUBSTR(MDATE,3,2))
  MDAY = VAL(SUBSTR(MDATE,5,2))
SELECT 6
  * find the sid # in layer file
  seek str(MSID,4)
  IF FOUND()
    DO WHILE MSID = SID_NO

280
SKIP
ENDDO
SKIP -1
* find the layer # which corresponds to the skid year
DO while jobcmpyr > myear
  skip -1
enddo
* if the layer year and skid year are the same, need to
* check the month to get the right layer number
DO WHILE JOBcmpyr = MYEAR .AND. JOBcmpyr > MMMH
  SKIP -1
  IF SID NO <> MSID
    SKIP
    MERRORS = MERRORS + 1
    EXIT
  ENDF
ENDO

* check to avoid if layer is a base, subbase or subgrade
* if not, then replace the structure and layer # from layer file
IF LAYDESC=5 .OR. LAYDESC=6 .OR. LAYDESC=7 .OR. LAYDESC=11
  SELE 3
  APPEND BLANK
  REPLACE SID NO WITH MSID
  REPLACE NO SURF with 'X'
  MERRORS = MERRORS + 1
ELSE
  MSTRUCTNUM = F->STRUCTNUM
  MLAYNUM = F->LAYNUM
ENDIF
ELSE
  SELE 3
  APPEND BLANK
  REPLACE SID NO WITH MSID
  REPLACE NO SURF with 'X'
  MERRORS = MERRORS + 1
ENDIF
IF MERRORS = 0
* add the data to the skid file
SELE 4
APPEND BLANK
REPLACE SID NO WITH MSID
REPLACE STRUCTNUM WITH MSTRUCTNUM
REPLACE LAYNUM WITH MLAYNUM
REPLACE YEAR WITH MYEAR
REPLACE ACTYEAR WITH MYEAR
REPLACE ACIMONTH WITH MMMH
REPLACE ACTDAY WITH MDAY
REPLACE SICOUNT WITH B->SICOUNT
REPLACE SIMEAN WITH B->SIMEAN
REPLACE SISD WITH B->SISDDEV
REPLACE SILOWVAL WITH B->SILOWVAL
REPLACE SIHIVAL WITH B->SIHIVAL
ENDIF

281
ELSE
  SELE 3
  APPEND BLANK
  REPL SID NO WITH MSID
  REPL NO PES WITH 'X'
ENDIF
SELE 1
SKIP
 ENDDO
CLOSE DATABASES
RETURN
Section 3: Inventory Data Update
Narrative on Adding Data Inventory Data

The ENTRLAYR.PRG program displays the ADD Inventory Data Menu. The program uses the procedure ADDLAYR.PRG to add data to the Inventory Files (Location, Layer Identification, Geometric and Shoulder, Surface, Subgrade and Layer Thickness Across The Road). Procedure ADDLAYR.PRG consists of the following five programs: LOCATION to add data to Location File, LAYER for Layer Identification File, GEOSHO for Geometric and Shoulder File, SURFACE for Surface File, SUBGRADE for Subgrade file.

Each of these five programs within ADDLAYR.PRG draws the screen forms to allow the user to enter data for a particular file. This data is stored in temporary dBASE files (LOCN.NEW.DBF, LAYR.NEW.DBF, LAYT.NEW.DBF, GEOS.NEW.DBF, SURF_NEW.DBF and SUBG.NEW.DBF).

After the user has entered all the data, he must backup the temporary dBASE files and run the Edit/Check programs to check the data entered. To accomplish this, ENTRLAYR.PRG calls INV_BKUP.PRG to backup up the newly entered data. After backup of the temporary files, INV_BKUP.PRG calls INV_UPDT.PRG which in turn calls the six Edit/Check procedures (LOCNCHCK.PRG, LAYRCHEK.PRG, LAYTCHEK.PRG, GEOSCHEK.PRG, SURFCHEK.PRG, SUBGCHEK.PRG). If there are no errors in the newly entered data, INV_UPDT.PRG updates the master files. If errors are present, an error listing is printed out.

The programs and the temporary dBASE files for this section (ADD Inventory Data) are stored in the subdirectory \PAVEDB\EDITUPDT.

The inventory add process is illustrated in Figures 13 through 15. Figure 13 depicts the add process on a global level, Figure 14 illustrates the high level program flow, logic, and Figure 15 charts the programs procedures, and input and output files used in the inventory add process.
Inventory Data - Add Process

FIGURE 13
INVENTORY DATA - ADD PROCESS
PROGRAM FLOW DIAGRAM

A -> B -> C
INV_UPDT_PRG
INV_BRUP_PRG
ADDLAYR_PRG
ENTRYR_PRG

FIGURE 15
INVENTORY DATA - ADD PROCESS
PROGRAM FLOW DIAGRAM (Continued)

FIGURE 15 (Continued)
INVENTORY DATA - ADD PROCESS
PROGRAM FLOW DIAGRAM (Continued)

FIGURE 15 (Continued)
Program Name: ENIRLAYR.PRG

Purpose: To display the ADD Inventory data menu and to call the programs that add data to the Inventory master files.

Procedures\Edits:

The following are the procedures:
1) Add data to the files using procedure ADDLAYR.PRG (includes programs LOCATION, LAYER, LAYTHICK, GEOUSHO, SURFACE and SUBGRADE).
2) Run the Edit/Check procedures to flag errors in the newly entered data (LOCALHEK.PRG, LAYRCHEK.PRG, LAYTCHEK.PRG, GEOCSHEK.PRG, SURFCHEK.PRG and SUBGCHEK.PRG). Each of these programs have a number of dBASE procedures within them - refer to dBASE Programs Called.

Input\Output Files:

<table>
<thead>
<tr>
<th>Files</th>
<th>Indexes</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOC_NAME.DEF</td>
<td>LOC_NNEW.NDX</td>
</tr>
<tr>
<td>LAYR_NAME.DEF</td>
<td>LAYR_NAME.NDX</td>
</tr>
<tr>
<td>LAYT_NAME.DEF</td>
<td>LAYT_NAME.NDX</td>
</tr>
<tr>
<td>GEOSUR_NAME.DEF</td>
<td>GEO_NAME.NDX</td>
</tr>
<tr>
<td>SURF_NAME.DEF</td>
<td>SURF_NAME.NDX</td>
</tr>
<tr>
<td>SUBG_NAME.DEF</td>
<td>SUBG_NAME.NDX</td>
</tr>
</tbody>
</table>

Temporary files -

Master Files -

LOCATION.DEF | LOCSD.IDE.NDX |
LAYER.DEF | LAYNIDEX.NDX |
LAYTHICK.DEF | LAYNDIX.NDX |
GEOUSHO.DEF | GEOIDEX.NDX |
SURFACE.DEF | SURFNDX.NDX |
SUBG_NAME.DEF | SUBGNDX.NDX |
dBASE Programs Called (See Program Flow Diagram):

- ADDLAYR.PRG
  - LOCATION
  - LAYER
  - LAYTHICK
  - GEOSHO
  - SURFACE
  - SUBGRADE
- INV_BKUP.PRG
- INV_UPDT.PRG
- LOCNCHK.PRG
- LAYRCHEK.PRG
  - SIDCHEK1
  - LAYRSVER
  - LAYDESC
  - LAYRCHEK3
  - LOCCHK1
- LAYTCHEK.PRG
  - LAYTCHEK1
  - LAYTCHEK3
  - SIDCHEK2
- GEOSCHEK.PRG
  - GEOSCHEK1
  - GEOSCHEK3
  - SIDCHEK3
- SURFCHK.PRG
  - SURFCHK1
  - SURFCHK3
  - SIDCHEK4
- SUBGCHK.PRG
  - SUBGCHK1
  - SUBGCHK3
  - SIDCHEK5
ADD INVENTORY SCREENS

The following screens are produced by ENIRLAYR.PRG and its subprograms except for the next menu (Edit & Update menu 3.0) which is produced by EDITUPDT.PRG.

<table>
<thead>
<tr>
<th>TEXAS FLEXIBLE PAVEMENT DATABASE</th>
<th>3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit &amp; Update</td>
<td></td>
</tr>
</tbody>
</table>

1 - Pavement Condition Data
2 - Inventory Data
3 - Traffic Data
4 - Tables

OPTION ➔

Choice 2 Inventory Data asks the user if he wants to ADD or CHANGE data to the inventory files. If he wants to ADD, ENIRLAYR.PRG program is run which produces the next screen (ADD Inventory Data 3.2.A).

<table>
<thead>
<tr>
<th>TEXAS FLEXIBLE PAVEMENT DATABASE</th>
<th>3.2.A</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDIT &amp; UPDATE</td>
<td></td>
</tr>
<tr>
<td>ADD Inventory Data</td>
<td></td>
</tr>
</tbody>
</table>

1 - Location
2 - Layer ID
3 - Geometric & Shoulder
4 - Surface
5 - Subgrade
6 - Layer Thickness Across The Road

K - Check New Data Entered and ADD to Files
E - Edit New Data Entered

OPTION ➔
Choice 1 Location displays the first screen on this page to let the user add Location data.
Choice 2 Layer ID displays the second screen on this page to let the user add Layer data.

### TEXAS FLEXIBLE PAVEMENT DATABASE
EDIT & UPDATE - ADD Inventory Location File

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID Number</td>
<td>0</td>
</tr>
<tr>
<td>Highway Ident.</td>
<td>0</td>
</tr>
<tr>
<td>Mile Post</td>
<td>0 0 TO 0 0</td>
</tr>
<tr>
<td>Mile Point</td>
<td>0.000 TO 0.000</td>
</tr>
<tr>
<td>HPMS Sample Number</td>
<td></td>
</tr>
<tr>
<td>Functional Classification</td>
<td>0</td>
</tr>
<tr>
<td>Active</td>
<td>T</td>
</tr>
<tr>
<td>Inactive Date</td>
<td>0/0</td>
</tr>
<tr>
<td>Previous SID</td>
<td>0</td>
</tr>
<tr>
<td>Next SID</td>
<td>0</td>
</tr>
<tr>
<td>Comment</td>
<td></td>
</tr>
<tr>
<td>District</td>
<td>0</td>
</tr>
<tr>
<td>County</td>
<td>0</td>
</tr>
<tr>
<td>Control/Section</td>
<td>0/0</td>
</tr>
<tr>
<td>Lane Identification</td>
<td></td>
</tr>
<tr>
<td>Mile Point Date</td>
<td>0/0</td>
</tr>
<tr>
<td>HPMS Section Subdivision</td>
<td>0</td>
</tr>
<tr>
<td>Number of Lanes</td>
<td>0</td>
</tr>
</tbody>
</table>

### TEXAS FLEXIBLE PAVEMENT DATABASE
EDIT & UPDATE - ADD Inventory Layer Identification

<table>
<thead>
<tr>
<th>Structure Number</th>
<th>Layr No.</th>
<th>Layer Desc.</th>
<th>Center Thick</th>
<th>Material Type Class.</th>
<th>Date Job Compltd Mnth Year</th>
<th>Widened Mnth Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
</tbody>
</table>
ADD INVENTORY SCREENS (continued)

Choice 3  Geometric & Shoulder displays the first screen on this page to enter data into.
Choice 4  Surface displays the second screen on this page to enter data into.

Texas Flexible Pavement Database
Edit & Update - Add Inventory
Geometric & Shoulder Information

<table>
<thead>
<tr>
<th>Sid Number</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure Number</td>
<td>0</td>
</tr>
<tr>
<td>Type of Pavement (See TTI Codes)</td>
<td>0</td>
</tr>
<tr>
<td>Lane Width (Feet)</td>
<td>0</td>
</tr>
<tr>
<td>Outside Shoulder Width (Feet)</td>
<td>0</td>
</tr>
<tr>
<td>Shoulder Surface Type</td>
<td>0</td>
</tr>
<tr>
<td>Shoulder Base Type (See Base Type Code, Table A.6)</td>
<td>0</td>
</tr>
<tr>
<td>Shoulder Surface Thickness (Inches)</td>
<td>0.0</td>
</tr>
<tr>
<td>Shoulder Base Thickness (Inches)</td>
<td>0.00</td>
</tr>
<tr>
<td>Widened Flag (0-2)</td>
<td>0</td>
</tr>
</tbody>
</table>

Texas Flexible Pavement Database
Edit & Update - Add Inventory
Surface Layer

<table>
<thead>
<tr>
<th>SID NUMBER</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure Number</td>
<td>0</td>
</tr>
<tr>
<td>Layer Number</td>
<td>0</td>
</tr>
<tr>
<td>Aggregate Application Rate</td>
<td>0</td>
</tr>
<tr>
<td>Type Admixture</td>
<td>0.00</td>
</tr>
<tr>
<td>Percent Admixture (Mean Asphalt Content)</td>
<td>0.00</td>
</tr>
<tr>
<td>Asphalt Application Rate</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Choice 3  Subgrade displays the first screen on this page to enter data into.
Choice 4  Layer Thickness Across The Road displays the second screen on this page to enter data into.

TEXAS FLEXIBLE PAVEMENT DATABASE
EDIT & UPDATE - ADD Inventory
Subgrade File

SID NUMBER  0
Percent Passing No. 200 Sieve  0.0
Texas Triaxial Class  0.0
Liquid Limit  0.0
Plasticity Index  0.0
Permeability Index  0.00

TEXAS FLEXIBLE PAVEMENT DATABASE
EDIT & UPDATE - ADD Inventory
Layer Thickness Across The Road

SID NUMBER  0

<table>
<thead>
<tr>
<th>Structure Number</th>
<th>Layer Number</th>
<th>Thickness - From Center</th>
<th>Distance From Center</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3rd Pos 2nd Pos 1st Pos Center</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.00 0.00 0.00 0.00</td>
<td>0.0 0.0 0.0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.00 0.00 0.00 0.00</td>
<td>0.0 0.0 0.0</td>
</tr>
</tbody>
</table>

299
PROGRAM LISTING

* SUBSYSTEM: EDIT & UPDATE INVENTORY dBASE III FILES
* PROGRAM NAME: ENTRIRAYR.PRG 06/06/88
* CALLED FROM: EDITUPDT.PRG
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TII
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: THIS PROGRAM DISPLAYS THE ADD INVENTORY DATA MENU
*

* checks to see if there are any inventory files in the process of
* being changed
*
MCOUNT = 0
IF FILE("\PAVEDB\EDITUPDT\LOCNCHG.DBF")
    MCOUNT = MCOUNT + 1
ENDIF
IF FILE("\PAVEDB\EDITUPDT\LAYRCHG.DBF")
    MCOUNT = MCOUNT + 1
ENDIF
IF FILE("\PAVEDB\EDITUPDT\LAYTCHG.DBF")
    MCOUNT = MCOUNT + 1
ENDIF
IF FILE("\PAVEDB\EDITUPDT\GEOSCHNG.DBF")
    MCOUNT = MCOUNT + 1
ENDIF
IF FILE("\PAVEDB\EDITUPDT\SURFCHNG.DBF")
    MCOUNT = MCOUNT + 1
ENDIF
IF FILE("\PAVEDB\EDITUPDT\SUBGCHNG.DBF")
    MCOUNT = MCOUNT + 1
ENDIF
IF MCOUNT <> 0
  CLEAR
  @ 2, 7 SAY "      ! ! ! WARNING ! ! !"
  @ 6, 7 SAY "You were in the process of changing data in the inventory"
  @ 6, 7 SAY " files. Therefore you cannot add data until you finish."
  @ 7, 7 SAY " the Change process."
  @ 8, 7 SAY "To complete the change process, "
  @ 9, 7 SAY " 1) Choose choice '3' from the Edit & Update Menu."
  @ 10, 7 SAY " 2) Type 'C' to the next question to Change Data"
  @ 11, 7 SAY " 3) Now you can either"
  @ 12, 7 SAY " A) Change the data"
  @ 13, 7 SAY " OR"
  @ 14, 7 SAY " B) Correct the Data and run the Edit/Check programs"
  @ 15, 7 SAY " i.e. Choice 'K'."
  @ 16, 7 SAY " Press any key to continue . . ."
CLEAN TYPE
WAIT ""
CLEAR

300
DO SURFACE
CASE EDITCOLL = "5"
   @ 19, 23 SAY "Please enter SID Number ===> " GET MSID_NO PICTURE "9999"
   READ
   CLEAR
   DO SUBGRADE
CASE EDITCOLL = "6"
   @ 19, 23 SAY "Please enter SID Number ===> " GET MSID_NO PICTURE "9999"
   READ
   CLEAR
   DO LAYTHICK
CASE EDITCOLL = "K"
   SET PROC TO
   DO \PAVEDB\EDITUPDT\INV_BRUP
   CLEA TYPE
   SET PROC TO \PAVEDB\EDITUPDT\ADDLAYR
CASE EDITCOLL = "E"
   SET PROC TO
   USE \PAVEDB\EDITUPDT\LOCN_NEW
   IF RECOUNT() <> 0
      SET FORMAT TO \PAVEDB\EDITUPDT\LOCATION_FMT
      CHANGE
      USE \PAVEDB\EDITUPDT\LOCN_NEW INDE \PAVEDB\EDITUPDT\LOCN_NEW
      PACK
   ENDF
   USE \PAVEDB\EDITUPDT\LAYR_NEW
   IF RECOUNT() <> 0
      SET FORMAT TO \PAVEDB\EDITUPDT\LAYER_FMT
      CHANGE
      USE \PAVEDB\EDITUPDT\LAYR_NEW INDE \PAVEDB\EDITUPDT\LAYR_NEW
      PACK
   ENDF
   USE \PAVEDB\EDITUPDT\LAYT_NEW
   IF RECOUNT() <> 0
      SET FORMAT TO \PAVEDB\EDITUPDT\LAYTHICK_FMT
      CHANGE
      USE \PAVEDB\EDITUPDT\LAYT_NEW INDE \PAVEDB\EDITUPDT\LAYT_NEW
      PACK
   ENDF
   USE \PAVEDB\EDITUPDT\GEOS_NEW
   IF RECOUNT() <> 0
      SET FORMAT TO \PAVEDB\EDITUPDT\GEOSH_FMT
      CHANGE
      USE \PAVEDB\EDITUPDT\GEOS_NEW INDE \PAVEDB\EDITUPDT\GEOS_NEW
      PACK
   ENDF
   USE \PAVEDB\EDITUPDT\SURF_NEW
   IF RECOUNT() <> 0
      SET FORMAT TO \PAVEDB\EDITUPDT\SURFACE_FMT
      CHANGE
      USE \PAVEDB\EDITUPDT\SURF_NEW INDE \PAVEDB\EDITUPDT\SURF_NEW
      PACK
   ENDF
   USE \PAVEDB\EDITUPDT\SUBG_NEW
RETURN
ENDIF

***************
* Inventory Enter data menu
***************
CLEAR
GETREPLY = ""
* Display the Inventory data menu on the screen
MOCONTINUE = .T.
DO WHILE MOCONTINUE
STORE " " TO EDITCOLL
MSID_NO = 0
DO WHILE .NOT. (EDITCOLL $ '12345678')
  @ 3, 21 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE 3.3.A"
  @ 4, 21 SAY "EDIT & UPDATE"
  @ 5, 21 SAY "ADD Inventory Data"
  @ 8, 15 SAY "1 - Location"
  @ 9, 15 SAY "2 - Layer ID"
  @ 10, 15 SAY "3 - Geometric & Shoulder"
  @ 11, 15 SAY "4 - Surface"
  @ 12, 15 SAY "5 - Subgrade"
  @ 13, 15 SAY "6 - Layer Thickness Across The Road"
  @ 15, 15 SAY "K - Check New Data Entered and ADD to Files"
  @ 16, 15 SAY "E - Edit New Data Entered"
  @ 18, 40 SAY "OPTION ------> " GET EDITCOLL
  @ 2, 9 TO 20, 65 DOUBLE
READ
  IF READKEY() = 12
    MOCONTINUE = .F.
    SET PROC TO
    EXIT
  ENDFID
ENDDO
IF MOCONTINUE
  SET PROCEDURE TO \PAVEDB\EDITUPDT\ADDLAYR
  DO CASE
    CASE EDITCOLL = "1"
      CLEAR
      DO LOCATION
    CASE EDITCOLL = "2"
      @ 19, 23 SAY "Please enter SID Number ------> " GET MSID_NO PICTURE "9999"
      READ
      CLEAR
      DO LAYER
    CASE EDITCOLL = "3"
      @ 19, 23 SAY "Please enter SID Number ------> " GET MSID_NO PICTURE "9999"
      READ
      CLEAR
      DO GEOSHO
    CASE EDITCOLL = "4"
      @ 19, 23 SAY "Please enter SID Number ------> " GET MSID_NO PICTURE "9999"
      READ
      CLEAR
      DO
IF RECOUNT() <> 0
  SET FORMAT TO \PAVEDB\EDITUPDT\SUBGRADE_FMT
  CHANGE
  USE \PAVEDB\EDITUPDT\SUBG_NEW INDE \PAVEDB\EDITUPDT\SUBG_NEW
  PACK
ENDIF
  SET FORMAT TO
  ENDCASE
  CLEAR
ENDIF
  CLEAR
  SET PROCEDURE TO
ENDDO
RETURN
PROGRAM LISTING

* * SUBSYSTEM:  EDIT & UPDATE INVENTORY dBASE III FILES
* PROGRAM NAME:  ADDLAYR.PRG  06/06/88
* CALLED FROM:  ENTRLAYR.PRG
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR:  TREVOR X. PEREIRA
* PURPOSE:  THIS PROCEDURE CALLS UP THE RESPECTIVE INVENTORY
* DATA FILES TO ENTER DATA INTO. THE FOLLOWING PROCEDURES
* CAN BE FOUND HERE:
*   - LAYER
*   - LAYTHICK
*   - GEOSHO
*   - SURFACE
*   - SUBGRADE
*   - LOCATION
*

*************************
* PROCEDURE LAYER
*************************

PROC LAYER
*
* To enter data into the layer identification file
*
STORE .T. TO MAGAIN
STORE 0 TO MSISTRUNUM1, MIAYNUM1, MIAYDESC1, MIAYMAT1, MJOBCMPM1
STORE 0.0 TO MCENTH1K1
STORE 0 TO MJOBCMPY1, MWIDEN1M1, MWIDENLY1

USE \PAVedb\EDITUPDT\LAYR_NEW INDEX \PAVedb\EDITUPDT\LAYR_NEW
STORE 0 TO MSISTRUNUM, MIAYNUM, MIAYDESC, MIAYMATC
STORE 0.0 TO MCENTH1K
STORE 0 TO MJOBCMPY, MJOBCMPM, MWIDENLY, MWIDENLM, MWIDEN

DO WHILE .NOT. (MSURE $ 'Yy')
  * initialize all variables
  CLEAR
  MSURE = " "
  DO WHILE .NOT. (MSURE $ 'Yy')
    * set up entry screen and get data
    @ 3, 23  SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
    @ 4, 23  SAY "EDIT & UPDATE - ADD Inventory"
    @ 5, 23  SAY " Layer Identification"
    @ 7, 55  SAY "Sid Number "
    @ 7, 66  SAY MSID_NO PICTURE "9999"
    @ 10, 54  SAY "Date"
    @ 11, 4  SAY " Material Job Compltdn"
    @ 12, 4  SAY "Structure Layr Layer Center Type "

304
\begin{verbatim}
@ 13, 4 SAY " Number  No.  Desc.  Thick  Class.  Mnth  Year"
Mnth  Year"
@ 14, 4 SAY "--------  ----  -----  ---------  ----  ----

@ 15, 8 SAY STR(MSTRUCNUM1,1)
@ 15, 17 SAY STR(MLAYNUM1,2)
@ 15, 24 SAY STR(MLAYDESC1,2)
@ 15, 30 SAY STR(MCENTHKL,4,1)
@ 15, 40 SAY STR(MLAYMATCL,2)
@ 15, 48 SAY STR(MJOBCMPL,2)
@ 15, 55 SAY STR(MJOBCMPY1,2)
@ 15, 62 SAY STR(MWIDENL1M1,2)
@ 15, 67 SAY STR(MWIDENLY1,2)
@ 16, 8 GET MSTRUCNUM PICTURE "9" RANGE 1,9
@ 16, 17 GET MLAYNUM PICTURE "99" RANGE 1,13
@ 16, 24 GET MLAYDESC PICTURE "99" RANGE 1,14
@ 16, 30 GET MCENTHKL PICTURE "99.99" RANGE 0,25.0
@ 16, 40 GET MLAYMATC PICTURE "99" RANGE 01,45
@ 16, 48 GET MJOBCMPL PICTURE "99" RANGE 1,12
@ 16, 55 GET MJOBCMPY PICTURE "99" RANGE 28,87
@ 16, 62 GET MWIDENL1M PICTURE "99" RANGE 0,12
@ 16, 67 GET MWIDENLY PICTURE "99" RANGE 0,87
@ 2, 2 TO 18, 74 DOUBLE
READ

* set up the escape key if want to abort
IF READKEY() = 12
    CLEAR
    USE
    RETURN
ENDIF
@ 20, 23 SAY "IS THE DATA CORRECT ? " GET MSURE PICTURE "/!
READ

* if data is said to be not correct, the flags (sure and required) are
* set to true so that the do loop will be executed again.
IF MSURE = "Y"
    MSTRUCNUM1 = MSTRUCNUM
    MLAYNUM1 = MLAYNUM
    MLAYDESC1 = MLAYDESC
    MCENTHKL = MCENTHKL
    MLAYMATCL = MLAYMATC
    MJOBCMPL = MJOBCMPL
    MJOBCMPY1 = MJOBCMPY
    MWIDENL1M = MWIDENL1M
    MWIDENLY1 = MWIDENLY1
    APPEND BLANK
    REPLACE SID NO WITH MSID NO
    REPLACE MSTRUCNUM WITH MSTRUCNUM
    REPLACE MLAYNUM WITH MLAYNUM
    REPLACE MLAYDESC WITH MLAYDESC
    REPLACE MCENTHKL WITH MCENTHKL
    REPLACE MLAYMATCL WITH MLAYMATC
\end{verbatim}
REPLACE JOBOMPYR WITH MJOBOMPY
REPLACE JOBOMPYO WITH MJOBOMPM
REPLACE WIDENLYR WITH MWIDENLY
REPLACE WIDENLIMO WITH MWIDENLIN
REPLACE ERRORMCHAR WITH .F.

STORE 0 TO MISTRUCNUM, MLAYNUM, MLAYDESC, MLAYMATC
STORE 0.0 TO MCENTHThK
STORE 0.0 TO MJOBOMPY, MJOBOMPM, MWIDENLY, MWIDENLIN, MWIDEN
ELSE
  @20, 0 CLEAR
ENDIF
ENDDO
ENDO
USE
CLEAR
RETURN

***************
* PROCEDURE LAYTHICK
***************
PROC LAYTHICK

*
* To enter data into the layer thickness across the road file
*
CLEAR
STORE 0 TO MISTRUCNUM, MLAYNUM
STORE 0.0 TO MFC3THK, MFC2THK, MFC1THK
STORE 0.0 TO MCENTHThK, MFC3DIS, MFC2DIS, MFC1DIS
STORE 0 TO MISTRUCNUM2, MLAYNUM2
STORE 0.0 TO MFC3THK2, MFC2THK2, MFC1THK2
STORE 0.0 TO MCENTHThK2, MFC3DIS2, MFC2DIS2, MFC1DIS2
USE \PAVEDB\EDITUPDT\LAYT_NEW INDEX \PAVEDB\EDITUPDT\LAYT_NEW
STORE .T. TO MAGAIN
DO WHILE MAGAIN
  CLEAR
  MSURE = " "
  DO WHILE .NOT. (MSURE $ 'Y')
    * set up entry screen and get data
    @ 3, 23 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
    @ 4, 23 SAY " EDIT & UPDATE - ADD Inventory"
    @ 5, 23 SAY " Layer Thickness Across The Road"
    @ 7, 56 SAY "SID NUMBER "
    @ 7, 67 SAY "MSID NO PICTURE "9999"
    @ 10, 25 SAY "Thickness - From Center Distance From Center"
    @ 11, 3 SAY "Structure Layer ____________________________"
    @ 12, 4 SAY "Number Number 3rd Pos 2nd Pos 1st Pos Center 3rd Pos
    2nd Pos 1st Pos"
    @ 13, 3 SAY "-------- ------ ------ ------ ------ ------
    ------- ------"
14,  6  SAY  str(MSTRUCNUM2,2)
14, 16  SAY  str(MLAYNUM2,2)
14, 24  SAY  str(MFC3THK2,5,2)
14, 32  SAY  str(MFC2THK2,5,2)
14, 40  SAY  str(MFC1THK2,5,2)
14, 47  SAY  str(MCENTTHK2,5,2)
14, 56  SAY  str(MFC3DIS2,4,1)
14, 64  SAY  str(MFC2DIS2,4,1)
14, 72  SAY  str(MFC1DIS2,4,1)
15,  6  GET  MSTRUCNUM  PICTURE "99" RANGE 1,9
15, 16  GET  MLAYNUM  PICTURE "99" RANGE 1,13
15, 24  GET  MFC3THK  PICTURE "99.99" RANGE 0.0,25.0
15, 32  GET  MFC2THK  PICTURE "99.99" RANGE 0.0,25.0
15, 40  GET  MFC1THK  PICTURE "99.99" RANGE 0.1,25.0
15, 47  GET  MCENTTHK  PICTURE "99.99" RANGE 0.1,25.0
15, 56  GET  MFC3DIS  PICTURE "99.9" RANGE 0.0,99.0
15, 64  GET  MFC2DIS  PICTURE "99.9" RANGE 0.0,99.0
15, 72  GET  MFC1DIS  PICTURE "99.9" RANGE 0.0,99.0
2,   0  TO  16, 79  DOUBLE
READ

* set up the escape key if want to abort
IF READKEY() = 12
  CLEAR
  USE
  RETURN
ENDIF

20, 23  SAY  "IS THE DATA CORRECT?"  GET MSURE PICTURE "!"
READ

* if data is not correct, redraw the entry screen; otherwise add
* the data to the file
IF MSURE = "Y"
  * data is added to the Layer Thickness Across the Road File
  MSTRUCNUM2 = MSTRUCNUM
  MLAYNUM2 = MLAYNUM
  MFC3THK2 = MFC3THK
  MFC2THK2 = MFC2THK
  MFC1THK2 = MFC1THK
  MCENTTHK2 = MCENTTHK
  MFC3DIS2 = MFC3DIS
  MFC2DIS2 = MFC2DIS
  MFC1DIS2 = MFC1DIS

  APPEND BLANK
  REPLACE SID NO WITH MSID NO
  REPLACE STRUCNUM WITH MSTRUCNUM
  REPLACE LAYNUM WITH MLAYNUM
  REPLACE FC3THK WITH MFC3THK
  REPLACE FC2THK WITH MFC2THK
  REPLACE FC1THK WITH MFC1THK
  REPLACE CENTTHK WITH MCENTTHK
  REPLACE FC3DIS WITH MFC3DIS
REPLACE FC2DIS WITH MFC2DIS
REPLACE FC1DIS WITH MFC1DIS
REPLACE ERRORCHAR WITH .F.
STORE 0 TO MSRCNUM, MLAYNUM
STORE 0.0 TO MFC3THK, MFC2THK, MFC1THK
STORE 0.0 TO MCENTTHK, MFC3DIS, MFC2DIS, MFC1DIS
ELSE
  @ 20, 0 CLEAR
ENDIF
ENDDO
USE
CLEAR
RETURN

***************
* PROCEDURE GEOMETRIC & SHOULDER
***************
PROC GEOSHO

*
* TO ENTER DATA INTO THE GEOMETRIC AND SHOULDER
* INFORMATION FILE
*

* set parameters and initialize variables
STORE .T. TO MAGAIN
CLEAR
USE \PAVE\EDIT\GEO\NEW INDEX \PAVE\EDIT\GEO\NEW
DO WHILE MAGAIN
  CLEAR
  MSURE = ""
  STORE 0 TO MOCONST, MPAVEYP, MLANEWID, MMONLANE, MOUTSHOW
  STORE 0 TO MSHOSFY, MSHOBSFY, MWIDEN
  STORE 0.0 TO MSHOSFT, MSHOBSFT
  DO WHILE NOT. (MSURE $ 'Yy')
    * set up data entry screen
      @ 3, 23 SAY "TExAS FLEXIBLE PAVEMENT DATABASE"
      @ 4, 23 SAY "EDIT & UPDATE - ADD Inventory"
      @ 5, 23 SAY "Geometric & Shoulder Information"
      @ 7, 56 SAY "Sid Number"
      @ 7, 67 SAY MSID NO PICTURE "9999"
      @ 8, 50 SAY "Structure Number"
      @ 8, 69 GET MOCONST PICTURE "99" RANGE 1,9
      @ 10, 8 SAY "Type of Pavement (See TTI Codes)"
      @ 10, 61 GET MPAVEYP PICTURE "99" RANGE 1,27
      @ 11, 8 SAY "Lane Width (Feet)"
      @ 11, 61 GET MLANEWID PICTURE "99" RANGE 8,15
      @ 12, 8 SAY "Outside Shoulder Width (Feet)"
      @ 12, 61 GET MOUTSHOW PICTURE "99" RANGE 0,15
      @ 13, 8 SAY "Shoulder Surface Type"
      @ 13, 61 GET MSHOSFY PICTURE "99" RANGE 1,6
@ 14, 8 SAY "Shoulder Base Type (See Base Type Code, Table A.6)"
@ 14, 61 GET MSHOBSTY PICTURE "99" RANGE 0.33
@ 15, 8 SAY "Shoulder Surface Thickness (Inches)"
@ 15, 59 GET MSHOSFTH PICTURE "99.9" RANGE 0.0,10.1
@ 16, 8 SAY "Shoulder Base Thickness (Inches)"
@ 16, 58 GET MSHOBSTH PICTURE "99.99" RANGE 0.0,21.0
@ 17, 8 SAY "Widened Flag (0-2)"
@ 17, 62 GET MWIDEN PICTURE "9" RANGE 0,2
@ 1, 3 TO 19, 74 DOUBLE
* get data
READ
* set up the escape key if want to abort
IF READKEY() = 12
  CLEAR
  USE
  RETURN
ENDIF
@ 20, 23 SAY "Is The Data Entered Correct? " GET MSURE PICTURE "!"
READ

* If the data is not correct, the inventory screen is displayed again
* Otherwise, add data to the Geometric And Shoulder Information File
IF MSURE = "Y"
  APPEND BLANK
  REPLACE SID NO WITH MSID NO
  REPLACE STRUCNO WITH MCONSTRU
  REPLACE PAVENTYP WITH MPAVETYP
  REPLACE LANEWID WITH MLANEWID
  REPLACE OUNISHOW WITH MOUTISHOW
  REPLACE SHOSFTYP WITH MSOSFTY
  REPLACE SHOBSTYP WITH MSOBSTY
  REPLACE SHOSFTHK WITH MSOSFTH
  REPLACE SHOBSTHK WITH MSOBSTH
  REPLACE WIDENFLG WITH MWIDEN
  REPLACE ERRORCHAR WITH .F.
ELSE
  @ 20, 0 CLEAR
ENDIF
ENDDO
ENDDO
USE
CLEAR
RETURN

***************************
* PROCEDURE SURFACE
***************************
PROC SURFACE

*
* TO ENTER DATA INTO SURFACE dBASE III FILE
*
* set parameters and initialize variables
STORE .T. TO MAGAIN
CLEAR
STORE 0 TO MOCONINU, MLAYNUM, MAGAPPLR
STORE " " TO MADMXTYP
STORE 0.0 TO MASAPPLR, MADMXPFR
USE \PAVEDB\EDITUPDT\SURF_NEW INDEX \PAVEDB\EDITUPDT\SURF_NEW

DO WHILE MAGAIN
  CLEAR
  MSURE = " 
  DO WHILE .NOT. (MSURE = \"Y\")
  * set up menu and get data
  @ 4, 20 SAY " TEXAS FLEXIBLE PAVEMENT DATABASE"
  @ 5, 20 SAY " EDIT & UPDATE - Inventory Data"
  @ 6, 20 SAY " ADD Surface Layer"
  @ 8, 57 SAY "SID NUMBER "
  @ 8, 68 SAY MSID NO PICTURE "9999"
  @ 10, 3 SAY " 
  @ 11, 3 SAY " Aggregate
  Asphalt"
  @ 12, 3 SAY "Structure Layer Application Type (Mean
  Asphalt Application"
  @ 13, 3 SAY "Number Number Rate Admixture Content)
  Rate"
  @ 14, 3 SAY "----------- ----------- ---------- -----------
  "
  @ 15, 6 GET MOCONINU PICTURE "99" RANGE 1,9
  @ 15, 16 GET MLAYNUM PICTURE "99" RANGE 1,13
  @ 15, 26 GET MAGAPPLR PICTURE "999" RANGE 0,200
  @ 15, 36 GET MADMXTYP PICTURE "XXXXXXXXXXXXX"
  @ 15, 56 GET MADMXPFR PICTURE "9.99" RANGE 0.0,8.0
  @ 15, 71 GET MASAPPLR PICTURE "9.99" RANGE 0,.6
  @ 2, 1 TO 17, 78 DOUBLE
  READ
  * set up escape key if want to abort
  IF READKEY() = 12
    CLEAR
    USE
    RETURN
  ENDIF
  @ 20, 23 SAY "Is The Data Entered Correct ? " GET MSURE PICT \\"!"
  READ

  IF MSURE = \"Y\"
  * if data so far is correct, add it to the surface file
  APPEND BLANK
  REPLACE SID NO WITH MSID NO
  REPLACE STRUCNUM WITH MOCONINU
  REPLACE IAYNUM WITH MLAYNUM
  REPLACE AGAPPLRT WITH MAGAPPLR
  REPLACE ADMXTRP WITH MADMXTYP

310
REPLACE ADMXPER WITH MADMXPER
REPLACE ASAPFLR WITH MASAPFLR
REPLACE ERRORCHAR WITH .F.
STORE 0 TO MCONTNU, MLAYNUM, MAGAPFLR
STORE " " TO MADMXTYP
STORE 0.0 TO MADMXPER
STORE 0.0 TO MASAPFLR, MADMXPER
ELSE
   @ 20, 0 CLEAR
ENDIF
ENDDO
ENDDO

* clear all memory variables
USE
CLEAR
RETURN

*********************
* PROCEDURE SUBGRADE
*********************
PROC SUBGRADE

*
* TO ENTER DATA INTO THE SUBGRADE FILE
*

* set parameters and initialize all variables
CLEAR
STORE 0.0 TO MPPSV200, MPLASTIX, MLIQLIM, MTRIAX, MPERMIX
USE \PAVE\EDIT\SUB\SUBG\NEW INDEX \PAVE\EDIT\SUB\NEW
STORE .T. TO MAGAIN

MSURE = " "
CLEAR
DO WHILE .NOT. (MSURE $ 'Y')
   * set up data entry screen
   @ 3, 20 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
   @ 4, 20 SAY "EDIT & UPDATE - ADD Inventory"
   @ 5, 20 SAY "Subgrade File"
   @ 7, 55 SAY "SID NUMBER"
   @ 7, 66 SAY MSID NO PICTURE "9999"
   @ 10, 15 SAY "Percent Passing No. 200 Sieve"
   @ 10, 47 GET MPPSV200 PICTURE "99.9"
   @ 11, 15 SAY "Texas Triaxial Class"
   @ 11, 48 GET MTRIAX PICTURE "9.9" RANGE 0.0, 6.0
   @ 12, 15 SAY "Liquid Limit"
   @ 12, 47 GET MLIQLIM PICTURE "99.9" RANGE 0.0, 80.0
   @ 13, 15 SAY "Plasticity Index"
   @ 13, 47 GET MPLASTIX PICTURE "99.9" RANGE 0.0, 70.0
   @ 14, 15 SAY "Permeability Index"
   @ 14, 46 GET MPERMIX PICTURE "99.99" RANGE 0.00, 10.00
   @ 1, 8 TO 17, 75   DOUBLE

311
* get data
READ
* set up the escape key if want to abort
IF READKEY() = 12
  CLEAR
  USE
  MSURE = ""
  MAGAIN = .F.
  EXIT
ENDIF
* verify if data is correct
@ 18, 23 SAY "Is The Data Correct ? "  GET MSURE PICT '!' READ
IF MSURE = "y"
  * add data to the subgrade file
  APPEND BLANK
  REPLACE SID NO  WITH MSID_NO
  REPLACE STRUCNUM WITH 1
  REPLACE LAYNUM WITH 1
  REPLACE PFSV200 WITH MPPSV200
  REPLACE PLASTIX WITH MPLASTIX
  REPLACE LIQLIM WITH MLIQLIM
  REPLACE TXTRIAXL WITH MTXTRIAX
  REPLACE PERMX WITH MPERMX
  REPLACE ERRORCHAR WITH .F.
  STORE 0.0 TO MPPSV200, MPLASTIX, MLIQLIM, MTXTRIAX, MPERMX
ELSE
  @ 20, 0 CLEAR
ENDIF
ENDDO
USE
CLEAR
RETURN

***************
* PROCEDURE ADD LOCATION RECORD
***************
PROCEDURE LOCATION
CLEAR
*
* To enter data into the transaction LOCATION identification file
*
STORE .T. TO MAGAIN
STORE 0 TO MHWYDIST, MCNTYNM, MOCTRL, MSECTION, MHWYNUM, MBEGMNT, MENDMNT, MMPNIMO
STORE 0 TO MMENYR, MBEGMPS, MEMPSIDIS, MENDMPS, MEMPSIDIS, MFUNCTCLS, MPHMSSEC
STORE 0 TO MINACIMR, MINACTYR, MPREVSID, MNEXISID, MNUMLANES, MSID NO
STORE " " TO MHWYPREFX
STORE " " TO MHWYSUFFX, MLANEID, MBDISSIGN, MEDISSIGN
STORE " " TO MHPMSSAM
STORE " " " TO MCOOMMENT
STORE .T. TO MACIVFLAG

USE \PAVEDB\EDITUPDT\LOCN_NEW INDEX \PAVEDB\EDITUPDT\LOCN_NEW

312
DO WHILE MAGAIN
  CLEAR
  MESURE = ""
DO WHILE .NOT. (MESURE $ 'Yy')
  * set up entry screen and get data
    @ 2, 23 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
    @ 3, 23 SAY "EDIT & UPDATE - ADD Inventory"
    @ 4, 23 SAY "Location File"
    @ 6, 6 SAY "SID Number"
    @ 6, 18 GET MSID NO PICTURE "9999"
    @ 6, 44 SAY "District"
    @ 6, 53 GET MHWYDIST PICTURE "99" RANGE 0, 25
    @ 6, 59 SAY "County"
    @ 6, 66 GET MONTYNUM PICTURE "999" RANGE 0, 500
    @ 8, 6 SAY "Highway Ident."
    @ 8, 21 GET MHWYPREFIX
    @ 8, 24 GET MHWYNUM PICTURE "9999"
    @ 8, 29 GET MHWYSUFFIX
    @ 8, 44 SAY "Control/Section"
    @ 8, 60 GET MCONTROL PICTURE "9999"
    @ 8, 64 SAY "/"
    @ 8, 65 GET MSECTION PICTURE "99"
    @ 10, 6 SAY "Mile Post"
    @ 10, 17 GET MESEGMPST PICTURE "9999"
    @ 10, 21 GET MEDISSIGN
    @ 10, 23 GET MESPSTDIS PICTURE "99"
    @ 10, 26 SAY "TO"
    @ 10, 29 GET MENDMPST PICTURE "9999"
    @ 10, 33 GET MEDISSIGN
    @ 10, 35 GET MESPSTDIS PICTURE "99"
    @ 10, 44 SAY "Lane Identification"
    @ 10, 64 GET MLANEID
    @ 12, 6 SAY "Mile Point"
    @ 12, 17 GET MSEGMENT PICTURE "99.999"
    @ 12, 24 SAY "TO"
    @ 12, 27 GET MENDMENT PICTURE "99.999"
    @ 12, 44 SAY "Mile Point Date"
    @ 12, 61 GET MMPNTIMO PICTURE "99" RANGE 0, 12
    @ 12, 63 SAY "/"
    @ 12, 64 GET MMPNTYR PICTURE "99"
    @ 14, 6 SAY "HPMS Sample Number"
    @ 14, 25 GET MPMSSAMPLE
    @ 14, 44 SAY "HPMS Section Subdivision"
    @ 14, 69 GET MPMSSEGC PICTURE "9"
    @ 16, 6 SAY "Functional Classification"
    @ 16, 32 GET MPUCNCLAS PICTURE "99"
    @ 16, 44 SAY "Number of Lanes"
    @ 16, 61 GET MNUMLANES PICTURE "999"
    @ 18, 6 SAY "Active ?"
    @ 18, 15 GET MACTVFLAG PICTURE "L"
    @ 18, 19 SAY "Inactive Date"
    @ 18, 33 GET MINACIMO PICTURE "99" RANGE 0, 12
    @ 18, 35 SAY "/"
    @ 18, 36 GET MINACITYR PICTURE "99"
@18, 41 SAY "Previous SID"
@18, 54 GET MPREVSID PICTURE "9999"
@18, 61 SAY "Next SID"
@18, 70 GET MNEXTSID PICTURE "9999"
@20, 6 SAY "Comment"
@20, 15 GET MCOMMENT PICTURE
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 1, 2 TO 21, 77 DOUBLE
READ
* set up the escape key if want to abort
IF READKEY() = 12
  CLEAR
  USE
  RETURN
ENDIF
@22, 23 SAY "IS THE DATA CORRECT ? " GET MSURE PICTURE "!"
READ
* if data is said to be not correct, the flags (sure and required) are
* set to true so that the do loop will be executed again.
IF MSURE = "Y"
  APPEND BLANK
  REPLACE SID NO WITH MSID NO
  REPLACE HWYDIST WITH MHWYDIST
  REPLACE CNTYNUM WITH MCNTYNUM
  REPLACE CONTROL WITH MCONTROL
  REPLACE SECTION WITH MSECTION
  REPLACE HWYPREFIX WITH MHWPREFIX
  REPLACE HWYNUM WITH MHWNUM
  REPLACE HWYSUFFIX WITH MHWYSUFFIX
  REPLACE BEGMENT WITH MBEGMENT
  REPLACE ENDMENT WITH MENDMENT
  REPLACE MMINIMO WITH MMPNIMO
  REPLACE MPTYR WITH MMPNYR
  REPLACE LANEID WITH MLANEID
  REPLACE BEGFAST WITH MBEGFAST
  REPLACE EDISSIGN WITH MBEDISIGN
  REPLACE EMEPIDIS WITH MEMEPIDIS
  REPLACE ENDFAST WITH MENDFAST
  REPLACE EDISSIGN WITH MEDISSIGN
  REPLACE EMEPIDIS WITH MEMEPIDIS
  REPLACE FNCIAS WITH MFNCIAS
  REPLACE HMMSSAM WITH MGMMSSAM
  REPLACE HMMSEC WITH MGMMSEC
  REPLACE INACIMO WITH MINACIMO
  REPLACE INACTYR WITH MINACTYR
  REPLACE PREVSID WITH MPREVSID
  REPLACE NEXTSID WITH MNEXTSID
  REPLACE ACTIVFLAG WITH MACTIVFLAG
  REPLACE NUMLANES WITH MNUMLANES
  REPLACE COMMENT WITH MCOMMENT
  REPLACE ERRORCHAR WITH .F.
STORE 0 TO
MHWYDIST, MCNTYNUM, MCONTROL, MSECTION, MHWNUM, MBEGMENT, MENDMENT, MMPNIMO
STORE 0 TO
MMPNYR, MBEGMTST, MEMPSTDIS, MENDMST, MEMPSTDIS, MFUNCIAS, MHPMSSEC
STORE 0 TO MINACIMO, MINACTYR, MPREVSID, MNEXTSID, MNUMLANES, MSID_NO
STORE " " TO MHWYPREFX
STORE " " TO MHWYSUFFIX, MLANEID, MBDISSIGN, MEDISSIGN
STORE " " TO MHPMSSAM
STORE " " TO MCOMMENT
STORE .T. TO MACTVFLAG
ELSE
   @ 20, 0 CLEAR
ENDIF
ENDDO
ENDDO
USE
CLEAR
RETURN
* SUBSYSTEM: EDIT & UPDATE
* PROGRAM NAME: INV BKUP.PRG 06/23/88
* CALLED FROM: ENTRIAVR.PRG
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TII
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: BACK UP MASTER INVENTORY FILES (ORIGINAL FILES) AND
  TRANSACTION INVENTORY FILES (NEWLY ENTERED DATA)
*
SET TALK OFF
SET ECHO OFF
SET ESCAPE OFF
CLEAR
CLEAR TYPE
STORE 0 TO MSIZE, MNUMFIELDS, MHEADER, MTOTALSIZE, MOUNT
STORE "" TO MNULL
MBACKDRV = MDRIVE2 + ":
@ 5,5 say "The new data you entered is about to be checked for errors."
@ 8,5 SAY "First, the Inventory files will be backed up. For this,"
@ 9,5 say "place a FORMATTED DISKETTE in drive " + MBACKDRV + " and close the
  drive."
@ 12,5 say "If you DO NOT want to continue, press the ESC key."
@ 13,5 say "Otherwise press any other key to continue."
READ
IF READKEY() = 12
  CLEAR
  RETURN
ENDIF
CLEAR
SET TYPE TO 0
*
backing up LOCATION files
USE \PAVEDB\EDITUPDT\LOCN_NEW INDEX \PAVEDB\EDITUPDT\LOCN_NEW
MOUNT = RECCOUNT()
USE
IF MOUNT <> 0
  CLEAR
  @ 9,5 SAY "Please wait. Backing up LOCATION files . . ."
  SET DEFAULT TO &MBACKDRV
  IF FILE('LOCATION.DBF')
    DELE FILE LOCATION.DBF
 ENDIF
  SET DEFAULT TO &MDRIVE
*
Checking diskspace on backup drive and backing up LOCATION master files
USE \PAVEDB\FILES\LOCATION
STORE RECCOUNT() * RECSIZE() TO MSIZE

316
MNULL = ""
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MTOTALSIZE = MSIZE + MHEADER + 20
SET DEFAULT TO &BACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDrive
DO WHILE MDISKSPACE < MTOTALSIZE
? "Not enough space on diskette in drive " + &BACKDRV
? "Please replace with another diskette and press any key to continue"
WAIT ""
IF READKEY() = 12
? "PROCESS ABORTED"
RETURN
ENDIF
SET DEFAULT TO &BACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDrive
ENDDO
USE
IF FILE("pavedb\files\backup\LOCATION.DBF")
    DELE FILE "pavedb\files\backup\LOCATION.DBF"
    SET TALK ON
ENDIF
COPY FILE "pavedb\files\LOCATION.DBF" TO &BACKDRV\LOCATION.DBF
COPY FILE "pavedb\files\LOCATION.DBF" TO "pavedb\files\backup\LOCATION.DBF"

* Backing up LOCATION transaction files i.e. newly entered data
* to drive Backup directory
IF FILE("pavedb\editupdt\backup\locn_0ld.dbf")
    SET TALK OFF
    DELE FILE "pavedb\editupdt\backup\locn_0ld.dbf"
    SET TALK ON
ENDIF
IF FILE("pavedb\editupdt\backup\locn_new.dbf")
    RENA "pavedb\editupdt\backup\locn_new.dbf" TO "pavedb\editupdt\backup\locn_0ld.dbf"
ENDIF
COPY FILE "pavedb\editupdt\locn_new.dbf" TO "pavedb\editupdt\backup\locn_new.dbf"
CLEAR
SET TALK OFF
ENDIF

* backing up layer id files
USE "pavedb\editupdt\layr_new index "pavedb\editupdt\layr_new"
MOUNT = RECOUNT()
USE
IF MOUNT <> 0
    CLEAR
    @ 9,5 SAY "Please wait. Backing up Layer ID . . ."
    SET DEFAULT TO &BACKDRV
ENDIF

317
IF FILE('LAYER.DBF')
   DELE FILE LAYER.DBF
ENDIF

* Checking disk space on backup drive and backing up Layer ID master files
SET DEFaul TO $MDRive
USE \PAVEDB\FILES\LAYER
STORE REcCOUNt() * RECSIZE() TO MSIZE
MNULL = ""
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MTOTALSIZE = MSIZE + MHEADER + 20
SET DEFAULT TO $MBACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO $MDRive
DO WHILE MDISKSPACE < MTOTALSIZE
    ?? "Not enough space on diskette in drive " + $MBACKDRV
    ?? "Please replace with another diskette and press any key to continue"
    WAIT ""
    IF READkey() = 12
        ?? "PROCESS ABORTED"
        RETURN
    ENDFiE
    SET DEFAULT TO $MBACKDRV
    MDISKSPACE = DISKSPACE()
    SET DEFAULT TO $MDRive
ENDDO
USE
IF FILE('\pavedb\files\backup\LAYER.DBF')
   DELE FILE $PAVEDB\FILES\backup\LAYER.DBF
ENDIF
SET TALK ON
COPY FILE \PAVEDB\FILES\LAYER.DBF TO $MBACKDRV\LAYER.DBF
COPY FILE \PAVEDB\FILES\LAYER.DBF TO $PAVEDB\FILES\backup\LAYER.DBF

* Backing up Layer ID transaction files i.e. newly entered data
* to Backup directory
IF FILE('\PAVEDB\EDITUPDT\backup\LAYR_OLD.DBF')
    SET TALK OFF
    DELE FILE \PAVEDB\EDITUPDT\backup\LAYR_OLD.DBF
    SET TALK ON
ENDIF
IF FILE('\PAVEDB\EDITUPDT\backup\LAYR_NEW.DBF')
    RENA \PAVEDB\EDITUPDT\backup\LAYR_NEW.DBF TO \PAVEDB\EDITUPDT\backup\LAYR_OLD.DBF
ENDIF
COPY FILE \PAVEDB\EDITUPDT\LAYR_NEW.DBF TO \PAVEDB\EDITUPDT\backup\LAYR_NEW.DBF
CLEAR
SET TALK OFF
ENDIF
* backing up LAYTHICK THICKNESS files
USE \PAVEDB\EDITUPDT\LAYT_NEW INDEX \PAVEDB\EDITUPDT\LAYT_NEW
MOUNT = RECCOUNT()
USE
IF MOUNT <> 0
    CLEAR
    @ 9,5 SAY "Please wait. Backing up LAYER THICKNESS Across the Road files ..
    ."
    SET DEFAULT TO &MBACKDRV
    IF FILE('LAYTHICK.DBF')
        DELETE FILE LAYTHICK.DBF
    ENDDIF

* Checking diskpace on Backup drive and backing up LAYTHICK ID master files
SET DEFAULT TO &Mdrive
USE \PAVEDB\FILES\LAYTHICK
STORE RECCOUNT() * RECSIZE() TO MSIZE
MNULL = ""
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MTOTALSIZE = MSIZE + MHEADER + 20
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &Mdrive
DO WHILE MDISKSPACE < MTOTALSIZE
    ? "Not enough space on diskette in drive " + MBACKDRV
    ? "Please replace with another diskette and press any key to continue"
    WAIT ""
    IF READKEY() = 12
        ? "PROCESS ABORTED"
        RETURN
    ENDDIF
    SET DEFAULT TO &MBACKDRV
    MDISKSPACE = DISKSPACE()
    SET DEFAULT TO &Mdrive
ENDDO
USE
IF FILE('pavedb\files\backup\LAYTHICK.DBF')
    DELETE FILE \PAVEDB\FILES\BACKUP\LAYTHICK.DBF
ENDDIF
SET TALK ON
COPY FILE \PAVEDB\FILES\LAYTHICK.DBF TO &MBACKDRV\LAYTHICK.DBF
COPY FILE \PAVEDB\FILES\LAYTHICK.DBF TO \PAVEDB\FILES\BACKUP\LAYTHICK.DBF

* Backing up LAYTHICK transaction files i.e. newly entered data
* to Backup directory
IF FILE('\PAVEDB\EDITUPDT\BACKUP\LAYT_OLD.DBF')
    SET TALK OFF
    DELETE FILE \PAVEDB\EDITUPDT\BACKUP\LAYT_OLD.DBF
    SET TALK ON
ENDDIF
 IF FILE('\PAVEDB\EDITUPDT\BACKUP\LAYT_NEW.DBF')
RENA \PAVEDB\EDITUPDT\BACKUP\LAYT_NEW.DBF TO \PAVEDB\EDITUPDT\BACKUP\LAYT_OLD.DBF ENDIF
COPY FILE \PAVEDB\EDITUPDT\LAYT_NEW.DBF TO \PAVEDB\EDITUPDT\BACKUP\LAYT_NEW.DBF CLEAR
SET TALK OFF ENDIF

* backing up GEOMETRIC & SHOULDER files
USE \PAVEDB\EDITUPDT\GEOS_NEW INDEX \PAVEDB\EDITUPDT\GEOS_NEW MOUNT = RECOUNT()
USE
IF MOUNT <> 0
CLEAR
@ 9,5 SAY "Please wait.  Backing up GEOMETRIC & SHOULDER files . . ."
SET DEFAULT TO &MBACKDRV
IF FILE('GEOSHO.DBF')
    DELE FILE GEOSHO.DBF
ENDIF

* Checking disk space on Backup drive and backing up GEOMETRIC & SHOULDER master files
SET DEFAULT TO &MDRIVE
USE \PAVEDB\FILES\GEOSHO
STORE RECOUNT() * RECSIZE() TO MSIZE
MNULL = ""
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MIOTALSIZE = MSIZE + MHEADER + 20
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDRIVE
DO WHILE MDISKSPACE < MIOTALSIZE
? "Not enough space on diskette in drive " + MBACKDRV
? "Please replace with another diskette and press any key to continue"
WAIT ""
IF READKEY() = 12
    "PROCESS ABORTED"
    RETURN
ENDIF
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDRIVE
ENDDO
USE
IF FILE('\pavedb\files\backup\GEOSHO.DBF')
    DELE FILE \PAVEDB\FILES\backup\GEOSHO.DBF
    SET TALK ON
ENDIF
COPY FILE \PAVEDB\FILES\GEOSHO.DBF TO &MBACKDRV\GEOSHO.DBF
COPY FILE \PAVEDB\FILES\GEOSHO.DBF TO \PAVEDB\FILES\BACKUP\GEOSHO.DBF
* Backing up GESHO transaction files i.e. newly entered data
* to drive Backup directory
IF FILE('PAVEDB\EDITUPDT\BACKUP\GEOS_OLD.DBF')
SET TALK OFF
DELE FILE PAVEDB\EDITUPDT\BACKUP\GEOS_OLD.DBF
SET TALK ON
ENDIF
IF FILE('PAVEDB\EDITUPDT\BACKUP\GEOS_NEW.DBF')
RENA PAVEDB\EDITUPDT\BACKUP\GEOS_NEW.DBF TO PAVEDB\EDITUPDT\BACKUP\GEOS_OLD.DBF
ENDIF
COPY FILE PAVEDB\EDITUPDT\GEOS_NEW.DBF TO PAVEDB\EDITUPDT\BACKUP\GEOS_NEW.DBF
CLEAR
SET TALK OFF
ENDIF

* backing up SURFACE THICKNESS files
USE PAVEDB\EDITUPDT\SURF_NEW INDEX PAVEDB\EDITUPDT\SURF_NEW
MOUNT = RECOUNT()
USE
IF MOUNT <> 0
    CLEAR
    @ 9,5 SAY "Please wait. Backing up SURFACE files . . ."
    SET DEFAULT TO &MBACKDRV
    IF FILE('SURFACE.DBF')
        DELE FILE SURFACE.DBF
    ENDIF

* Checking disk space on Backup drive and backing up SURFACE master files
SET DEFAULT TO &MDRIVE
USE PAVEDB\FILES\SURFACE
STORE RECOUNT() * RECSIZE() TO MSIZE
MNULL = ""
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDO
MHEADER = (32 * MNUMFIELDS) + 34
MTOTALSIZE = MSIZE + MHEADER + 20
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDRIVE
DO WHILE MDISKSPACE < MTOTALSIZE
    ? "Not enough space on diskette in drive " + MBACKDRV
    ? "Please replace with another diskette and press any key to continue"
    WAIT ""
    IF READKEY() = 12
        ? "PROCESS ABORTED"
        RETURN
    ENDIF
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDRIVE
ENDDO
USE
IF FILE("\pavedb\files\backup\SURFACE.DBF")
    DELETE FILE \PAVEDB\FILES\BACKUP\SURFACE.DBF
ENDIF
SET TALK ON
COPY FILE \PAVEDB\FILES\SURFACE.DBF TO &BACKDRV\SURFACE.DBF
COPY FILE \PAVEDB\FILES\SURFACE.DBF TO \PAVEDB\FILES\BACKUP\SURFACE.DBF

* Backing up SURFACE transaction files i.e. newly entered data
* to drive Backup directory
IF FILE("\PAVEDB\EDITUPDT\BACKUP\SURF_OLD.DBF")
    SET TALK OFF
    DELETE FILE \PAVEDB\EDITUPDT\BACKUP\SURF_OLD.DBF
    SET TALK ON
ENDIF
IF FILE("\PAVEDB\EDITUPDT\BACKUP\SURF_NEW.DBF")
    RENAME \PAVEDB\EDITUPDT\BACKUP\SURF_NEW.DBF TO \PAVEDB\EDITUPDT\BACKUP\SURF_OLD.DBF
ENDIF
COPY FILE \PAVEDB\EDITUPDT\SURF_NEW.DBF TO \PAVEDB\EDITUPDT\BACKUP\SURF_NEW.DBF
CLEAR
SET TALK OFF
ENDIF

* backing up SUBGRADE THICKNESS files
USE \PAVEDB\EDITUPDT\SUBG_NEW INDEX \PAVEDB\EDITUPDT\SUBG_NEW MOUNT = RECCOUNT()
USE
IF MOUNT <> 0
CLEAR
@ 9,5 SAY "Please wait. Backing up SUBGRADE files . . ."
SET TALK OFF
SET DEFAULT TO &BACKDRV
IF FILE("SUBGRADE.DBF")
    DELETE FILE SUBGRADE.DBF
ENDIF

* Checking disk space on Backup drive and backing up SUBGRADE master files
SET DEFAULT TO &MDELETE
USE \PAVEDB\FILES\SUBGRADE
STORE RECCOUNT() * RECSIZE() TO MSIZE
MNULL = ""
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MTOTALSIZE = MSIZE + MHEADER + 20
SET DEFAULT TO &BACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDELETE
DO WHILE MDISKSPACE < MTOTALSIZE
    ? "Not enough space on diskette in drive " + &BACKDRV

322
? "Please replace with another diskette and press any key to continue"
WAIT " "
IF READKEY() = 12
   ? "PROCESS ABORTED"
   RETURN
ENDIF
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDRIVE
ENDDO
USE
IF FILE("\pavedb\files\backup\SUBGRADE.DBF")
   DELE FILE \PAVEDB\FILES\BACKUP\SUBGRADE.DBF
ENDIF
SET TALK ON
COPY FILE \PAVEDB\FILES\SUBGRADE.DBF TO &MBACKDRV\SUBGRADE.DBF
COPY FILE \PAVEDB\FILES\SUBGRADE.DBF TO \PAVEDB\FILES\BACKUP\SUBGRADE.DBF

* Backing up SUBGRADE transaction files i.e. newly entered data
* to drive Backup directory
IF FILE("\PAVEDB\EDITUPDT\BACKUP\SUBG_OLD.DBF")
   SET TALK OFF
   DELE FILE \PAVEDB\EDITUPDT\BACKUP\SUBG_OLD.DBF
   SET TALK ON
ENDIF
IF FILE("\PAVEDB\EDITUPDT\BACKUP\SUBG_NEW.DBF")
   RENA \PAVEDB\EDITUPDT\BACKUP\SUBG_NEW.DBF TO \PAVEDB\EDITUPDT\BACKUP\SUBG_OLD.DBF
ENDIF
COPY FILE \PAVEDB\EDITUPDT\SUBG_NEW.DBF TO \PAVEDB\EDITUPDT\BACKUP\SUBG_NEW.DBF
CLEAR
SET TALK OFF
ENDIF

SET ESCAPE OFF
SET TALK OFF
SET ECHO OFF
SET TYPE TO 20
DO \PAVEDB\EDITUPDT\INV_UPDT
RETURN
SET ALTE TO \PAVEDB\EDITUPDT\ERRORS
SET ALTE OFF

* Checking LOCATION File

USE \PAVEDB\EDITUPDT\LOCN_NEW INDEX \PAVEDB\EDITUPDT\LOCN_NEW
MOUNT = REACCOUNT()
USE
IF MOUNT <> 0
 * Running the edit/check programs
 CLEAR
 @ 5,5 say "The new LOCATION data you entered is"
 @ 6,5 SAY "being checked for errors."
 USE \PAVEDB\EDITUPDT\LOCN_NEW INDEX \PAVEDB\EDITUPDT\LOCN_NEW
 DO WHILE .NOT. EOF()
  REPLACE ERRORCHAR WITH .F.
  SKIP
 ENDDO
 USE
 DO \PAVEDB\EDITUPDT\LOCNCHEK

 CLEAR
 @ 5, 5 SAY "Please Wait. Updating Master Database . . . ."
 * Appending data to the Master database if there are no errors
 SEEK 1
 USE \PAVEDB\EDITUPDT\LOCN_NEW INDEX \PAVEDB\EDITUPDT\LOCN_NEW
 SEEK 2
 USE \PAVEDB\FILES\LOCATION INDEX \PAVEDB\INDEXES\LOCSID
 SEEK 1
 DO WHILE .NOT. EOF()
 IF .NOT. ERRORCHAR
SELE 2
APPEND BLANK
REPLACE SID NO WITH A->SID NO
REPLACE HWYDIST WITH A->HWYDIST
REPLACE CNTYNUM WITH A->CNTYNUM
REPLACE CONTROL WITH A->CONTROL
REPLACE SECTION WITH A->SECTION
REPLACE HWYPREFIX WITH A->HWYPREFIX
REPLACE HWYNUM WITH A->HWYNUM
REPLACE HWYSUBFIX WITH A->HWYSUBFIX
REPLACE BEGMPNT WITH A->BEGMPNT
REPLACE ENDMNT WITH A->ENDMNT
REPLACE MNIMO WITH A->MNIMO
REPLACE MNITYR WITH A->MNITYR
REPLACE LANEID WITH A->LANEID
REPLACE BEGMPST WITH A->BEGMPST
REPLACE EDISSIGN WITH A->EDISSIGN
REPLACE EMDPSTID WITH A->EMDPSTID
REPLACE ENDMPST WITH A->ENDMPST
REPLACE EDISSIGN WITH A->EDISSIGN
REPLACE EMDPSTID WITH A->EMDPSTID
REPLACE FUCILAS WITH A->FUCILAS
REPLACE HMSSAM WITH A->HMSSAM
REPLACE HMSSSEC WITH A->HMSSSEC
REPLACE INACTIMO WITH A->INACTIMO
REPLACE INACTYR WITH A->INACTYR
REPLACE PREVSID WITH A->PREVSID
REPLACE NEXSID WITH A->NEXSID
REPLACE ACTVFLAG WITH A->ACTVFLAG
REPLACE NUMLANES WITH A->NUMLANES
REPLACE COMMENT WITH A->COMMENT
SELE 1
DELE
ENDIF
SKIP
ENDDO
SELE 1
PACK
ENDIF
SELE 1
USE
SELE 2
USE

*  
* Checking layer id file  
*  
USE "PAVE\EDITUPDT\LAYR_NEW INDEX "PAVEED\EDITUPDT\LAYR_NEW
MOUNT = RB0COUNT()  
USE
IF MOUNT <> 0
  @ 5,5 SAY "The new LAYER IDENTIFICATION data you entered is"
  @ 6,5 SAY "being checked for errors."
  * Running the edit/check programs

325
USE \PAVEDB\EDITUPDT\LAYR_NEW INDE \PAVEDB\EDITUPDT\LAYR_NEW
DO WHILE .NOT. EOF()
    REPLACE ERRORCHAR WITH .F.
    SKIP
ENDDO
USE
SET PROC TO \PAVEDB\EDITUPDT\LAYRCHK
DO LAYRCHK3
DO LAYRSVER
DO SIDCHEK1
DO LOCCHEK1
DO LAYDESC
CLEAR
@ 5, 5 SAY "Please Wait. Updating Master Database . . . ."
* Appending data to the Master database if there are no errors
SELE 1
    USE \PAVEDB\EDITUPDT\LAYR_NEW INDEX \PAVEDB\EDITUPDT\LAYR_NEW
SELE 2
    USE \PAVEDB\FILES\LAYER INDEX \PAVEDB\INDEXES\LAYNDX
SELE 1
DO WHILE .NOT. EOF()
    * If there is any error in any record for a SID number then do not
    * add it to the master file
    MERROR = 0
    MTESTSID = A->SID_NO
    DO WHILE .NOT. EOF() .AND. A->SID_NO = MTESTSID
        IF A->ERRORCHAR
            MERROR = MERROR + 1
        ENDIF
        SKIP
    ENDDO
    IF MERROR = 0
        LOCATE FOR SID NO = MTESTSID
    DO WHILE .NOT. EOF() .AND. A->SID_NO = MTESTSID
        SELE 2
        APPEND BLANK
        REPLACE SID NO WITH A->SID NO
        REPLACE STRUCNUM WITH A->STRUCNUM
        REPLACE LAYNUM WITH A->LAYNUM
        REPLACE LAYDESC WITH A->LAYDESC
        REPLACE CENTTHK WITH A->CENTTHK
        REPLACE LAYMATCL WITH A->LAYMATCL
        REPLACE JOBCMPYR WITH A->JOBCMPYR
        REPLACE JOBCMPO WITH A->JOBCMPO
        REPLACE WIDENLYR WITH A->WIDENLYR
        REPLACE WIDENIMO WITH A->WIDENIMO
        SELE 1
        DEL
        SKIP
    ENDDO
    ENDF
    ENDDO
    SELE 1
    PACK

326
ENDIF
SELE 1
USE
SELE 2
USE

* Checking Layer Thickness Across the Road File
*
USE \\
PAVEDB/EDITUPDT/LAYT_NEW INDEX \\
PAVEDB/EDITUPDT/LAYT_NEW
MOUNT = RECOUNT()
USE
IF MOUNT <> 0
  @ 5,5 SAY "The new LAYER THICKNESS ACROSS THE ROAD data you entered is"
  @ 6,5 SAY "being checked for errors."
* Running the edit/check programs
USE \\
PAVEDB/EDITUPDT/LAYT_NEW INDEX \\
PAVEDB/EDITUPDT/LAYT_NEW
DO WHILE .NOT. EOF()
    REPLACE ERRORCHAR WITH .F.
    SKIP
ENDDO
USE
SET PROC TO \\
PAVEDB/EDITUPDT/LAYTCHK
DO LAYTCHK1
DO LAYTCHK3
DO SIDCHK2
CLEAR
@ 5, 5 SAY "Please Wait. Updating Master Database . . . ."
* Appending data to the Master database if there are no errors
SELE 1
USE \\
PAVEDB/EDITUPDT/LAYT_NEW INDEX \\
PAVEDB/EDITUPDT/LAYT_NEW
SELE 2
USE \\
PAVEDB/FILES/LAYTHICK INDEX \\
PAVEDB/INDEXES/LAYTINDX
SELE 1
DO WHILE .NOT. EOF()
    IF .NOT. ERRORCHAR
        SELE 2
        APPEND BLANK
        REPLACE SID NO WITH A->SID NO
        REPLACE STRUCNUM WITH A->STRUCNUM
        REPLACE LAYNUM WITH A->LAYNUM
        REPLACE FC3THK WITH A->FC3THK
        REPLACE FC2THK WITH A->FC2THK
        REPLACE FC1THK WITH A->FC1THK
        REPLACE CENTTHK WITH A->CENTTHK
        REPLACE FC3DIS WITH A->FC3DIS
        REPLACE FC2DIS WITH A->FC2DIS
        REPLACE FC1DIS WITH A->FC1DIS
        SELE 1
        DELE
    ENDF
    SKIP
ENDDO
SELE 1
PACK
ENDDF
SELE 1
USE
SELE 2
USE

* Checking Geometric & Shoulder Information File
*
USE \PAVEDB\EDITUPDT\GEOS_NEW INDEX \PAVEDB\EDITUPDT\GEOS_NEW
MOUNT = REccount()
USE
IF MOUNT <> 0
* Running the edit/check programs
CLEAR
@ 5,5 say "The new GEOMETRIC & SHOULDER data you entered is"
@ 6,5 say "being checked for errors."
USE \PAVEDB\EDITUPDT\GEOS_NEW INDEX \PAVEDB\EDITUPDT\GEOS_NEW
DO WHILE .NOT. EOF()
    REPLACE ERRORCHAR WITH .F.
    SKIP
ENDDO
USE
SET PROC TO \PAVEDB\EDITUPDT\GEOSCHEK
DO GEOSCHK1
DO GEOSCHK3
DO SIDCHEK3
SET PROC TO
CLEAR
@ 5, 5 SAY "Please Wait. Updating Master Database . . . ." 
* Appending data to the Master database if there are no errors
SELE 1
USE
USE \PAVEDB\EDITUPDT\GEOS_NEW INDEX \PAVEDB\EDITUPDT\GEOS_NEW
SELE 2
USE
USE \PAVEDB\FILES\GEOSHO INDEX \PAVEDB\INDEXES\GEONDX
SELE 1
DO WHILE .NOT. EOF()
IF .NOT. ERRORCHAR
SELE 2
APPEND BLANK
REPLACE SID NO WITH A->SID NO
REPLACE STRUCNUM WITH A->STRUCNUM
REPLACE PAVETYP WITH A->PAVETYP
REPLACE LANEWID WITH A->LANEWID
REPLACE OUTSHOWD WITH A->OUTSHOWD
REPLACE SHOSFTYP WITH A->SHOSFTYP
REPLACE SHOBSTYP WITH A->SHOBSTYP
REPLACE SHOSFTHK WITH A->SHOSFTHK
REPLACE SHOBSTHK WITH A->SHOBSTHK

328
REPLACE WIDENFLG WITH A->WIDENFLG
SELE 1
DELE
ENDIF
SKIP
ENDDO
SELE 1
PACK
ENDDO
SELE 1
USE
SELE 2
USE

*
* Checking Surface Information File
*
USE \PAVEDB\EDITUPDT\SURF_NEW INDEX \PAVEDB\EDITUPDT\SURF_NEW
MOUNT = RECCOUNT()
USE
IF MOUNT <> 0
  clear
  @ 5,5 say "The new Surface Information File data you entered is"
  @ 6,5 SAY "being checked for errors."
  *
  * Running the edit/check programs
  CLEAR
  USE \PAVEDB\EDITUPDT\SURF_NEW INDEX \PAVEDB\EDITUPDT\SURF_NEW
  DO WHILE .NOT. EOF()
    REPLACE ERRORCHAR WITH .F.
    SKIP
  ENDDO
  USE
  SET PROC TO \PAVEDB\EDITUPDT\SURFCHKEK
  DO SURFCHKEK1
  DO SURFCHKEK3
  DO SIDCHEK4
  CLEAR
  @ 5,5 SAY "Please Wait. Updating Master Database . . . ."
  *
  * Appending error-free SID's to the Master database
  SELE 1
  USE
  USE \PAVEDB\EDITUPDT\SURF_NEW INDEX \PAVEDB\EDITUPDT\SURF_NEW
  SELE 2
  USE
  USE \PAVEDB\FILES\SURFACE INDEX \PAVEDB\INDEXES\SURFNDX
  SELE 1
  DO WHILE .NOT. EOF()
    IF .NOT. A->ERRORCHAR
      SELE 2
      APPEND BLANK
      REPLACE SID NO WITH A->SID NO
      REPLACE STRUCNUN WITH A->STRUCNUN
    ELSE
REPLACE LAYNUM WITH A->LAYNUM
REPLACE AGAPPLRT WITH A->AGAPPLRT
REPLACE ADMXTYP WITH A->ADMXTYP
REPLACE ADMXPER WITH A->ADMXPER
REPLACE ASAPPLRT WITH A->ASAPPLRT
SELE 1
DELETE
ENDIF
SKIP
ENDDO
SELE 1
PACK
ENDIF
SELE 1
USE
SELE 2
USE

* Checking Subgrade File *
USE \PAVEDB\EDITUPDT\SUBG_NEW INDEX \PAVEDB\EDITUPDT\SUBG_NEW
MOUNT = RECCOUNT()
USE
IF MOUNT <> 0
  CLEAR
  CLEAR TYPE
  @ 5,5 SAY "The new SUBGRADE FILE data you entered is"
  @ 6,5 SAY "being checked for errors."

  * Running the edit/check programs
  CLEAR
  USE \PAVEDB\EDITUPDT\SUBG_NEW INDEX \PAVEDB\EDITUPDT\SUBG_NEW
  DO WHILE .NOT. EOF()
    REPLACE ERRORCHAR WITH .F.
  SKIP
  ENDDO
  USE
  SET PROC TO \PAVEDB\EDITUPDT\SUBGCHK
  DO SUBGCHK1
  DO SUBGCHK3
  DO SICHECK5
  CLEAR
  @ 5,5 SAY "Please Wait. Updating Master Database . . . ."
  * Appending data to the Master database if there are no errors
  SELE 1
  USE
  USE \PAVEDB\EDITUPDT\SUBG_NEW INDEX \PAVEDB\EDITUPDT\SUBG_NEW
  SELE 2
  USE
  USE \PAVEDB\FILES\SUBGRADE INDEX \PAVEDB\INDEXES\SUBGNDX
  SELE 1
DO WHILE .NOT. EOF()
    IF .NOT. ERRORCHAR
        SELE 2
        APPEND BLANK
        REPLACE SID NO WITH A->SID_NO
        REPLACE STRUCNUM WITH A->STRUCNUM
        REPLACE LAYNUM WITH A->LAYNUM
        REPLACE PPSV200 WITH A->PPSV200
        REPLACE PLASTIX WITH A->PLASTIX
        REPLACE LIQILIM WITH A->LIQILIM
        REPLACE TXTRIAXL WITH A->TXTRIAXL
        REPLACE PERMIX WITH A->PERMIX
    ENDIF
    DELETE
    SKIP
    ENDDO
    SELE 1
    PACK
ENDIF
SELE 1
USE
SELE 2
USE
SELE 1
CLOSE ALTE
SET PRINT ON
? "LIST OF ERRORS IN NEWLY ENTERED INVENTORY DATA FILES"
? "-----------------------------------------------"
TYPE \PAVEDB\EDITUPDT\ERRORS.TXT
SET PRINT OFF
RETURN
PROGRAM LISTING

* SUBSYSTEM: EDIT & UPDATE
* PROGRAM NAME: LOCNCHEK.PRG 06/17/88
* CALLED FROM: INV_UPDT.PRG
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TII
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: CHECKS LOCATION DATA
*

SET TALK OFF
CLEAR
SELE 1
USE \PAVEDB\EDITUPDT\LOCN_NEW INDE \PAVEDB\EDITUPDT\LOCN_NEW
SELE 2
USE \PAVEDB\FILES\LOCATION INDE \PAVEDB\INDEXES\LOCSID
M_ERROR = "LOCATION FILE - INCORRECT DATA ENTERED"
SELE 1
MERRORCNT = 0
SET ALERT ON
? M_ERROR
DO WHILE .NOT. EOF()
    MSID_NO = SID NO
    * calculates the correct Sid Number
    STORE 0 TO VAR1, VAR2, VAR3, COMPARE
    VAR1 = INT(MSID_NO/1000)
    VAR2 = MOD(INT(MSID_NO/100),10)
    VAR2 = VAR2 * 2
    VAR3 = MOD(INT(MSID_NO/10),10)
    VAR3 = VAR3 * 3
    VAR4 = MOD(VAR1,10)+VAR2+VAR3
    COMPARE = MOD(VAR4,10)
    * compares calculated Sid number with Sid number entered
    IF COMPARE <> MOD(MSID_NO,10)
        MERRORCNT = MERRORCNT + 1
        DISPLAY FIELD SID NO OFF
        ? "SID NUMBER IS NOT VALID"
        ? " "
    ENDIF
    * verifies that Sid number has been entered
    IF MSID NO = 0
        MERRORCNT = MERRORCNT + 1
        DISPLAY FIELD SID NO OFF
        ? " SID NUMBER IS NOT VALID"
        ? ""
    ENDIF
SELE 2
SEEK MSID NO
IF FOUND()
    MERRORCNT = MERRORCNT + 1
    DISPLAY FIELD SID NO OFF
    ? " SID Number is already present in Location file"
    ? "
ENDIF
SELE 1
IF HWYDIST < 0 .OR. HWYDIST > 27
    DISPLAY FIELDS SID NO, HWYDIST OFF
    MERRORCNT = MERRORCNT + 1
ENDIF
IF CNTYNUM < 0 .OR. CNTYNUM > 270
    DISPLAY FIELDS SID NO, CNTYNUM OFF
    MERRORCNT = MERRORCNT + 1
ENDIF
DO CASE
    CASE HWYPREFIX = "FM"
    CASE HWYPREFIX = "SH"
    CASE HWYPREFIX = "IH"
    CASE HWYPREFIX = "US"
    CASE HWYPREFIX = "SP"
    CASE HWYPREFIX = "LP"
    CASE HWYPREFIX = "FR"
    OTHERWISE
        DISPLAY FIELDS SID NO, HWYPREFIX OFF
        MERRORCNT = MERRORCNT + 1
ENDCASE
IF HWYNUM = 0
    DISPLAY FIELDS SID NO, HWYNUM OFF
    MERRORCNT = MERRORCNT + 1
ENDIF
IF EDISSIGN = "+" .OR. EDISSIGN = "-"
ELSE
    DISPLAY FIELDS SID NO, EDISSIGN OFF
    MERRORCNT = MERRORCNT + 1
ENDIF
IF EDISSIGN = "+" .OR. EDISSIGN = "-"
ELSE
    DISPLAY FIELDS SID NO, EDISSIGN OFF
    MERRORCNT = MERRORCNT + 1
ENDIF
IF LANEID = "R" .OR. LANEID = "L"
ELSE
    DISPLAY FIELDS SID NO, LANEID OFF
    MERRORCNT = MERRORCNT + 1
ENDIF
IF ENDMNT = 0 .AND. BEGMNT = 0
    DISPLAY FIELDS SID NO, BEGMNT, ENDMNT OFF
    MERRORCNT = MERRORCNT + 1
ENDIF
IF MPMIMO < 0 .OR. MPMIMO > 12
    DISPLAY FIELDS SID NO, MPMIMO OFF
    MERRORCNT = MERRORCNT + 1
END
ENDIF
IF NUMLANES < 1 .OR. NUMLANES > 22
   DISPLAY FIELDS SID NO, NUMLANES OFF
   MERRORCNT = MERRORCNT + 1
ENDIF
IF ACIVFLAG
   IF INACTYR <> 0
      DISPLAY FIELDS SID NO, INACTYR OFF
      ? " Inactive Flag and Inactive YEAR are inconsistent"
      ? " "
      ? " "
      MERRORCNT = MERRORCNT + 1
   ENDIF
   IF INACIMO <> 0
      DISPLAY FIELDS SID NO, INACIMO OFF
      ? " Inactive Flag and Inactive MONTH are inconsistent"
      ? " "
      ? " "
      MERRORCNT = MERRORCNT + 1
   ENDIF
   IF NEXTSID <> 0
      DISPLAY FIELDS SID NO, NEXTSID OFF
      ? " Cannot point to Another SID number when present SID number is active"
      ? " "
      ? " "
      MERRORCNT = MERRORCNT + 1
   ENDIF
ENDIF
IF PREVSID <> 0
   MPREVSID = PREVSID
   MBEGMPST = BEGMPST
   MEISSIGN = BDISSIGN
   MEMPSIDIS = EMPSTDIS
   MENMPST = ENDMPST
   MEISSIGN = EDISSIGN
   MEMPSIDIS = EMPSTDIS
   SELE 2
   SEEK MPREVSID
   IF .NOT. FOUND()
      SELE 1
      DISPLAY FIELD SID NO, PREVSID
      ? " SID Number NOT present in location file"
      ? " "
      ? " "
      MERRORCNT = MERRORCNT + 1
   ELSE
      IF MBEGMPST <> BEGMPST .OR. MEISSIGN <> BDISSIGN .OR. MEMPSIDIS <>
      EMPSTDIS
         SELE 1
         DISPLAY FIELD SID NO, PREVSID OFF
         ? " SID Number and PREVIOUS SID Beginning Mile Post do not match"
         ? " "
         ? " 
      ENDIF
   ENDIF
ENDIF
MERRORCNT = MERRORCNT + 1
ENDIF
SE1 2
IF MENDMPST <> ENDMPST .OR. MEDISSIGN <> EDISSIGN .OR. MEMPSTDIS <>

EMPSTDIS
SE1 1
DISPLAY FIELD SID_NO, PREVSID OFF
? " SID Number and PREVIOUS SID Beginning Mile Post do not match"
? ""
? ""
MERRORCNT = MERRORCNT + 1
ENDIF
ENDIF
SE1 1
ENDIF
IF CONTROL = 0
DISPLAY FIELDS SID NO, CONTROL OFF
MERRORCNT = MERRORCNT + 1
ENDIF
IF SECTION = 0
DISPLAY FIELDS SID NO, SECTION OFF
MERRORCNT = MERRORCNT + 1
ENDIF
IF FUCNCLAS < 1 .OR. FUCNCLAS > 7
DISPLAY FIELDS SID NO, FUCNCLAS OFF
MERRORCNT = MERRORCNT + 1
ENDIF
IF MENDMPST = 0
DISPLAY FIELDS SID NO, ENDMPST OFF
MERRORCNT = MERRORCNT + 1
ENDIF
IF EMPSTDIS < 0
DISPLAY FIELDS SID NO, EMPSTDIS OFF
MERRORCNT = MERRORCNT + 1
ENDIF
IF EMPSTDIS < 0
DISPLAY FIELDS SID NO, EMPSTDIS OFF
MERRORCNT = MERRORCNT + 1
ENDIF
IF MPRY < 20 .OR. MPRY > 99
DISPLAY FIELDS SID NO, HWYNUM OFF
MERRORCNT = MERRORCNT + 1
ENDIF
IF .NOT. ACTIVFLG
IF INACTYR < 72 .OR. INACTYR > 99
DISPLAY FIELDS SID NO, INACTYR OFF
MERRORCNT = MERRORCNT + 1
ENDIF
IF INACIMO < 0 .OR. INACIMO > 12
DISPLAY FIELDS SID NO, INACIMO OFF
MERRORCNT = MERRORCNT + 1
ENDIF
ENDIF
IF MERRORCNT <> 0

335
REPLACE ERRORCHAR WITH .T.
MERRORCNT = 0
ENDIF
SKIP
ENDDO
SELE 1
USE
SELE 2
USE
* reset parameters
CLEAR
?? CHR(12)
SET AITE OFF
RETURN
* SUBSYSTEM: EDIT & UPDATE
* PROGRAM NAME: LAYRCHK.PRG 06/08/88
* CALLED FROM: INV UPDT.PRG
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE SYSTEM
* TAMI/TTI
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: LIST OF PROCEDURES TO CHECK FOR ERRORS IN THE LAYER FILE
*

***************
* PROCEDURE LAYER DESCRIPTION
***************
PROC LAYDESC

* set parameters
SET TALK OFF
SET ECHO OFF
CLEAR

USE \PAVEDB\EDITUPDT\LAYR NEW INDEX \PAVEDB\EDITUPDT\LAYR NEW
M ERROR = "LAYER ID FILE - INCORRECT LAYER DESCRIPTION OR MATERIAL CLASSIFICATION"
M ERROR2 = "______________________________"
SET ALERT ON
? M ERROR
? M_ERROR2
* check layer description with layer material classification
DO WHILE .NOT. EOF()
    DO CASE
        CASE LAYDESC = 7
            IF LAYMATCL < 41 OR LAYMATCL > 45
                DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
                REPLACE ERRORCHAR WITH .T.
            ENDIF
        CASE LAYDESC = 6
            IF LAYMATCL < 21 OR LAYMATCL > 33
                IF LAYMATCL <> 17
                    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
                    REPLACE ERRORCHAR WITH .T.
                ENDIF
            ENDIF
        CASE LAYDESC = 5
            IF LAYMATCL < 21 OR LAYMATCL > 27
                IF LAYMATCL <> 17
                    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
                ENDIF
            ENDIF
    END CASE
END DO
REPLACE ERRORCHAR WITH .T.
ENDIF

CASE LAYDESC = 4
IF LAYMATCL < 1 .OR. LAYMATCL > 4
   DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
   REPLACE ERRORCHAR WITH .T.
ENDIF

CASE LAYDESC = 3
IF LAYMATCL < 1 .OR. LAYMATCL > 17
   DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
   REPLACE ERRORCHAR WITH .T.
ENDIF

CASE LAYDESC = 2
IF LAYMATCL <> 11
   DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
   REPLACE ERRORCHAR WITH .T.
ENDIF

CASE LAYDESC = 1
IF LAYMATCL < 1 .OR. LAYMATCL > 16
   DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
   REPLACE ERRORCHAR WITH .T.
ENDIF

CASE LAYDESC = 10
IF LAYMATCL < 5 .OR. LAYMATCL > 7
   DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
   REPLACE ERRORCHAR WITH .T.
ENDIF

CASE LAYDESC = 12
IF LAYMATCL < 12 .OR. LAYMATCL > 15
   DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
   REPLACE ERRORCHAR WITH .T.
ENDIF

CASE LAYDESC = 14
IF LAYMATCL <> 27
   DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
   REPLACE ERRORCHAR WITH .T.
ENDIF
ENDCASE
SKIP
ENDDO
USE
* reset parameters
? CHR(12)
CLEAR
SET ALITE OFF
RETURN
***************
* PROCEDURE VERIFY LAYER NUMBERS
***************
PROC IAYRSVER
*
* VERIFIES THAT THE LAYER NUMBERS ARE IN CONSECUTIVE
* ORDER IN THE TRANSACTION FILE AND THAT NONE ARE MISSING
*
CLEAR
SELE 1
USE \PAVEDB\FILES\LAYER INDEX \PAVEDB\INDEXES\LAYNDX
SELE 2
USE \PAVEDB\EDITUPDT\LAYR_NEW INDE \PAVEDB\EDITUPDT\LAYR_NEW

M_ERROR = "LAYER ID FILE - MISSING OR INCORRECT LAYER NUMBERS"
M_ERROR2 = "---------------------------------------------"

SET ALIKE ON
? M_ERROR
? M_ERROR2
DO WHILE .NOT. EOF()
   MSID = SID_NO

   SELE 1
   SEEK STR(MSID,4)
   IF FOUND()
      DO WHILE .NOT. EOF() .AND. SID_NO <> MSID
         SKIP
      ENDDO
      SKIP -1
      MLAYNUM = LAYNUM + 1
   ELSE
      MLAYNUM = 1
   ENDF

   * checks for consecutive sid numbers
   SELE 2
   DO WHILE .NOT. EOF() .AND. B->SID_NO = MSID
      IF LAYNUM <> MLAYNUM
         IF LAYDESC = 13 .OR. LAYDESC = 12
            SKIP
         STORE LAYNUM TO MLAYNUM
      IF MSID <> SID_NO
         SKIP -1
      ENDF
   ELSE
      DISPLAY SID_NO, STRUCNUM, LAYNUM OFF
      REPLACE ERRORCHAR WITH .T.
      DO WHILE .NOT. EOF() .AND. MSID = SID_NO
         SKIP
      REPLACE ERRORCHAR WITH .T.

339
ENDO
  SKIP -1
ENDIF
ENDIF
MLAYNUM = MLAYNUM + 1
IF .NOT. EOF()
  SKIP
ENDIF
ENDDO
USE
* resets parameters
N = 0
DO WHILE N < 6
  ? CHR(13)
  N = N + 1
ENDDO
SET ALIVE OFF
clear
RETURN

*******************************
* PROCEDURE LAYER OUT OF RANGE CHECK
*******************************
PROC LAYRCHK3
*
* TO LIST OUT OF RANGE DATA IN THE LAYER IDENTIFICATION
* TRANSACTION FILE
*
* set parameters
SET TALK OFF
SET ECHO OFF
USE \PAVEDB\EDITUPDT\LAYR_NEW INDEX \PAVEDB\EDITUPDT\LAYR_NEW
M_ERRORCK = 0
M_ERROR = "LAYER ID FILE - OUT OF RANGE DATA"
M_ERROR2 = "-----------------------------------"
SET ALIVE ON
? M_ERROR
? M_ERROR2
* check the data
DO WHILE .NOT. EOF()
  IF STRUCNUM < 1 .OR. STRUCNUM > 9
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDF
  IF LAYNUM < 1 .OR. LAYNUM > 13
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM OFF
    M_ERRORCK = M_ERRORCK + 1
ENDO

340
ENDIF
IF LAYDESC < 1 .OR. LAYDESC > 14
  DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC OFF
  M_ERRORCK = M_ERRORCK + 1
ENDIF
IF CENTTHK < .1 .OR. CENTTHK > 25
  IF LAYNUM <> 1 .AND. CENTTHK = 0
    ELSE
      DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, CENTTHK OFF
      M_ERRORCK = M_ERRORCK + 1
    ENDIF
  ENDIF
ENDIF
IF LAYMATCL < 1 .OR. LAYMATCL > 45
  DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYMATCL OFF
  M_ERRORCK = M_ERRORCK + 1
ENDIF
IF JOBCMPYR < 20 .OR. JOBCMPYR > 99
  DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, JOBCMPYR OFF
  M_ERRORCK = M_ERRORCK + 1
ENDIF
IF JOBCMPMO < 1 .OR. JOBCMPMO > 12
  DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, JOBCMPMO OFF
  M_ERRORCK = M_ERRORCK + 1
ENDIF
IF WIDENLYR < 28 .OR. WIDENLYR > 99
  IF WIDENLYR <> 0
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, WIDENLYR OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDIF
ENDIF
IF WIDENIMO < 1 .OR. WIDENIMO > 12
  IF WIDENIMO <> 0
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, WIDENIMO OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDIF
ENDIF
ENDIF
IF M_ERRORCK <> 0
  REPLACE ERRORCHAR WITH .T.
  M_ERRORCK = 0
ENDIF
SKIP
ENDDO

USE
* reset parameters
N = 0
DO WHILE N < 6
  ? CHR(13)
  N = N + 1
ENDDO
CLEAR
SET ALTE OFF
RETURN
**********************
* PROCEDURE CHECK SID NUMBER
**********************
PROC SIDCHEK1
*
* LIST OUT SID NUMBERS THAT ARE INCORRECT
*
* set parameters
set talk off
set echo off
clear

USE \PAVEDB\EDITUPD\LAYR_NEW INDEX \PAVEDB\EDITUPD\LAYR_NEW
M_ERROR = "LAYER ID FILE - INCORRECT SID NUMBERS"
M_ERROR2 = "----------------------------------"
SET AITE ON
? M_ERROR
? M_ERROR2

* initialize variables
STORE 0 TO VAR1, VAR2, VAR3, VAR4, COMPARE

* calculate correct Sid Number check digit
DO WHILE .NOT. EOF()
  VAR1 = INT(SID_NO/1000)
  VAR2 = MOD(INT(SID_NO/100),10)
  VAR2 = VAR2 * 2
  VAR3 = MOD(INT(SID_NO/10),10)
  VAR3 = VAR3 * 3
  VAR4 = MOD(VAR1,10)+VAR2+VAR3
  COMPARE = MOD(VAR4,10)

  * compare actual check digit with calculated check digit
  IF COMPARE <> MOD(SID_NO,10)
    DISPLAY SID_NO, STRUCNUM, LAYNUM OFF
    REPLACE ERRORCHAR WITH .T.
  ENDIF
  IF SID_NO = 0
    DISPLAY SID_NO, STRUCNUM, LAYNUM OFF
    REPLACE ERRORCHAR WITH .T.
  ENDIF
  SKIP
ENDDO
USE
* reset parameters
N = 0
DO WHILE N < 6
  ? CHR(13)
  N = N + 1
ENDDO
SET AITE OFF
USE
CLEAR
RETURN

***********************
* PROCEDURE LOCATION CHECK
***********************
PROC LOCHECK1
*
* LIST OUT LAYER RECORDS THAT DO NOT HAVE A LOCATION RECORD
*
* set parameters
set talk off
set echo off
clear

SELE 1
USE \PAVEDB\EDITUPDT\LAYR_NEW INDEX \PAVEDB\EDITUPDT\LAYR_NEW
SELE 2
USE \PAVEDB\FILES\LOCATION INDE \PAVEDB\INDEXES\LOGSID

M_ERROR = "LAYER ID FILE - ENTERED SID NUMBERS THAT DO NOT HAVE A LOCATION RECORD"
M_ERROR2 =
"_"

SET ALTIE ON
? M_ERROR
? M_ERROR2
SELE 1
DO WHILE .NOT. EOF()
    MSID_NO = A->SID_NO
    SELE 2
    SEEK MSID_NO
    IF .NOT. FOUND()
        SELE 1
        DISPLAY SID NO, STRUCNUM, LAYNUM OFF
        REPLACE ERRCHAR WITH .T.
    ENDIF
    SELE 1
    SKIP
ENDDO
SELE 1
USE
SELE 2
USE

* reset parameters
N = 0
DO WHILE N < 6
    ? CHR(13)
    N = N + 1
ENDDO
SET ALTE OFF
CLEAR
RETURN
* SUBSYSTEM: EDIT & UPDATE INVENTORY FILES
* PROGRAM NAME: LAYTCK. PRG     06/08/88
* CALLED FROM: INV UPDT. PRG
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO LIST ERRORS INT THE LAYER THICKNESS ACROSS THE ROAD FILE
*

*********************
* PROCEDURE CHECK LAYER THICKNESS FOR RECORDS NOT IN LAYER FILE
*********************
PROC LAYTCK1
* set parameters
Set talk off
set echo off
CLEAR

* print out page heading
SET ALTIE ON
MITTILE = "LAYER THICKNESS FILE - LIST OF NEW RECORDS ENTERED NOT PRESENT IN LAYER ID FILE"
MITTILE2 =
"---------------------------------------------------------------------"
? MITTILE
? MITTILE2

* assign files to different work areas
SELECT 1
   USE \PAVEDB\FILES\LAYER INDEX \PAVEDB\INDEXES\LAYNDX
SELECT 2
   USE \PAVEDB\EDITUPDT\LAYT_NEW INDEX \PAVEDB\EDITUPDT\LAYT_NEW

DO WHILE .NOT. EOF()
   MLAYSID = SID NO
   MLAYSRC = STRUCNUM
   MLAYNUM = LAYNUM

   * find the laythick file record in layer file
SELECT 1
   seek str(MLAYSID,4)+str(MLAYSRC,2)+str(MLAYNUM,2)
   IF .NOT. FOUND()
      SELECT 2
      DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM OFF
      REPLACE ERRORCHAR WITH .T.
   ENDIF
SELECT 2
   SKIP

345
ENDDO

* reset parameters
N = 0
DO WHILE N < 6
  ? CHR(13)
  N = N + 1
ENDDO
CLEAR
SET AITE OFF
SELE 1
USE
SELE 2
USE
RETURN

***********************
* PROCEDURE TO CHECK OUT OF RANGE DATA
***********************
PROC LAYTCHK3
*
* TO LIST ALL OUT OF RANGE DATA IN THE LAYER THICKNESS
* ACROSS THE ROAD FILE
*
* set parameters
SET TALK OFF
SET ECHO OFF
USE \PAVEDB\EDITUPDT\LAYT_NEW INDEX \PAVEDB\EDITUPDT\LAYT_NEW
GOTO TOP
M_ERROR = 0

* print out the page heading
M_TITLE = "LAYER THICKNESS FILES - OUT OF RANGE CHECK"
M_TITLE2 = "----------------------------------------------"
SET AITE ON
? M_TITLE
? M_TITLE2

* check the ranges
DO WHILE .NOT. EOF()
  IF STRUCNUM < 1 .OR. STRUCNUM > 9
    DISPLAY FIELDS SID_NO, STRUCNUM, LAYNUM, STRUCNUM OFF
    M_ERROR = M_ERROR + 1
ENDIF
  IF LAYNUM < 1 .OR. LAYNUM > 13
    DISPLAY FIELDS SID_NO, STRUCNUM, LAYNUM, LAYNUM OFF
    M_ERROR = M_ERROR + 1
ENDIF
  IF FC3THK < 0 .OR. FC3THK > 25
    DISPLAY FIELDS SID_NO, STRUCNUM, LAYNUM, FC3THK OFF
    M_ERROR = M_ERROR + 1
ENDIF
ENDWHILE

346
ENDIF
IF FC2THK < 0 .OR. FC2THK > 25
   DISPLAY SID_NO, STRUCNUM, LAYNUM, FC2THK OFF
   M_ERROR = M_ERROR + 1
ENDIF
IF FC1THK < .1 .OR. FC1THK > 25
   DISPLAY SID_NO, STRUCNUM, LAYNUM, FC1THK OFF
   M_ERROR = M_ERROR + 1
ENDIF
IF CENTTHK < .1 .OR. CENTTHK > 25
   DISPLAY SID_NO, STRUCNUM, LAYNUM, CENTTHK OFF
   M_ERROR = M_ERROR + 1
ENDIF
IF FC1DIS < .01 .OR. FC1DIS > 99
   DISPLAY SID_NO, STRUCNUM, LAYNUM, FC1DIS OFF
   M_ERROR = M_ERROR + 1
ENDIF
IF FC2DIS < FC1DIS
   IF FC2DIS <> 0
      DISPLAY SID_NO, STRUCNUM, LAYNUM, FC2DIS OFF
      M_ERROR = M_ERROR + 1
   ENDIF
ENDIF
IF FC3DIS < FC2DIS
   IF FC3DIS <> 0
      DISPLAY SID_NO, STRUCNUM, LAYNUM, FC3DIS OFF
      M_ERROR = M_ERROR + 1
   ENDIF
ENDIF
IF M_ERROR <> 0
   REPL ERRORCHAR WITH .T.
   M_ERROR = 0
ENDIF
SKIP
ENDDO

* reset parameters
N = 0
DO WHILE N < 6
   ? CHR(13)
   N = N + 1
ENDDO
CLEAR
SET ALITE OFF
USE
RETURN

***********************
* PROCEDURE TO CHECK THE SID NUMBER
***********************
PROC SIDCHEXK2
*

347
* TO CHECK FOR INVALID SID NUMBERS
*

* set parameters
set talk off
set echo off
CLEAR

USE \PAVEDB\EDITUPDT\LAYT_NEW INDEX \PAVEDB\EDITUPDT\LAYT_NEW

* set up page heading
M_TITLE1 = "LAYERTHICKNESS FILE - SID NUMBER CHECK"
M_TITLE2 = "----------------------------------"
SET ALINE ON
? M_TITLE1
? M_TITLE2
STORE 0 TO VAR1, VAR2, VAR3, VAR4, COMPARE

* calculate check digit
DO WHILE .NOT. EOF()
    VAR1 = INT(SID_NO/1000)
    VAR2 = MOD(INT(SID_NO/100),10)
    VAR2 = VAR2 * 2
    VAR3 = MOD(INT(SID_NO/10),10)
    VAR3 = VAR3 * 3
    VAR4 = MOD(VAR1,10)+VAR2+VAR3
    COMPARE = MOD(VAR4,10)

    * compare calculated check digit with actual check digit
    IF COMPARE <> MOD(SID_NO,10)
        DISPLAY SID NO OFF
        REPL ERRORCHAR WITH .T.
        MSID = SID NO
        DO WHILE SID_NO = MSID .OR. (.NOT. EOF())
            SKIP
        ENDDO
        SKIP -1
    ENDF
    IF SID NO = 0
        DISPLAY SID NO, STRUCNUM, LAYNUM OFF
        REPL ERRORCHAR WITH .T.
    ENDF
    SKIP
ENDDO

* reset parameters
? CHR(12)
CLEAR
SET ALINE OFF
USE
RETURN

***************
* PROCEDURE LAYER THICKNESS CHECK 2
***************
PROC LAYTHCK2
*
* TO LIST LAYER IDENTIFICATION FILE RECORDS THAT ARE NOT
* PRESENT IN LAYER THICKNESS ACROSS THE ROAD FILE
*
* set parameters
SET TALK OFF
SET ECHO OFF
CLEAR

M_TITLE = "LAYER THICKNESS FILE - LIST OF ALL RECORDS IN LAYER ID FILE THAT ARE NOT"
M_TITLE2 = "PRESENT IN THE LAYER THICKNESS FILE"
M_TITLE3 = ""__________________________"

SET ALTE ON
? M_TITLE
? M_TITLE2
? M_TITLE3

* assign files to different work areas
SELECT 1
  USE \PAVEDB\FILES\LAYER INDEX \PAVEDB\INDEXES\LAYINDEX
SELECT 2
  USE \PAVEDB\FILES\LAYTHICK INDEX \PAVEDB\INDEXES\LAYINDEX

SELECT 1
DO WHILE .NOT. EOF()
  MSID = SID_NO
  MSTR = STRUCNUM
  MLAY = LAYNUM

* if layer number is 1, skip to the next record in layer file
IF LAYNUM = 1
  SKIP
  LOOP
ENDIF

* find the layer record in laythick file
SELECT 2
SEEK SSTR(MSID,4)+SSTR(MSTR,2)+SSTR(MLAY,2)
IF .NOT. FOUND()
  SELECT 1
  DISPLAY FIELDS SID_NO, STRUCNUM, LAYNUM OFF
  REPLACE ERRORCHAR WITH .T.
ENDIF
SELECT 1
SKIP
ENDDO

* reset parameters
N = 0
DO WHILE N < 6
    ? CHR(13)
    N = N + 1
ENDDO
CLEAR
SET ALTIE OFF
SELE 1
USE
SELE 2
USE
RETURN
* PROGRAM LISTING

* SUBSYSTEM: EDIT & UPDATE INVENTORY FILES
* PROGRAM NAME: GEOSCHK1.RG 06/07/88
* CALLED FROM: INV UPDT.RG
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: PROCEDURES TO LIST NEWLY ENTERED RECORDS THAT ARE
* IN ERROR

********************
* PROCEDURE GEOSCHK1
********************
PROC GEOSCHK1

* set parameters
set talk off
set echo off
CLEAR

M_TITLE = "GEOMETRIC & SHOULDER FILE - LIST OF NEWLY ENTERED RECORDS NOT IN
LAYER ID FILE"
M_TITLE2 = "----------------------------------------------------------------------------------"

SET ALl REasel
? M_TITLE
? M_TITLE2

* assign database files to different work areas
SELECT 1
   USE \PAVEDB\FILES\LAYER INDEX \PAVEDB\INDEXES\LAYNDX
SELECT 2
   USE \PAVEDB\EDITUPDT\GEOS_NEW INDEX \PAVEDB\EDITUPDT\GEOS_NEW
SELE 2
DO WHILE .NOT. EOF()
   * store to memory variables
   MSID = SID_NO
   MSTRUC = STRUCNUM

SELECT 1
   * find the record in layer that equals the sid number and structure
   * number
   SEEK STR(MSID,4)+STR(MSTRUC,2)

   * if record is not found, print the sid number
   IF .NOT. FOUND()
      SELE 2

351
DISPLAY FIELD SID_NO, STRUCNUM OFF
REPLACE ERRCHAR WITH .T.
ENDIF
SELE 2
SKIP
ENDDO

* reset parameters
N = 0
DO WHILE N < 6
    ? CHR(13)
    N = N + 1
ENDDO
CLEAR
SET ALINE OFF
SELE 1
USE
SELE 2
USE
return

**************************
* PROCEDURE GEOSCHK3
**************************
PROC GEOSCHK3

*
* TO LIST ALL OUT OF RANGE DATA FOR THE GEOMETRIC AND
* INFORMATION FILE
*

* set parameters
SET TALK OFF
SET ECHO OFF

USE \PAVEDB\EDITUPDT\GEOS_NEW INDEX \PAVEDB\EDITUPDT\GEOS_NEW
M_ERROR = 0

* set up page heading
M_TTITLE  = "GEOMETRIC & SHOULDER FILE - OUT OF RANGE CHECK"
M_TTITLE2 = "----------------------------------------"
SET ALINE ON
? M_TTITLE
? M_TTITLE2

* check data range
DO WHILE .NOT. EOF()
    IF STRUCNUM < 1 .OR. STRUCNUM > 9
        DISPLAY FIELDS SID_NO, STRUCNUM OFF
        M_ERROR = M_ERROR + 1
    ENDF
    IF PAVETYP < 1 .OR. PAVETYP > 27
        DISPLAY FIELDS SID_NO, STRUCNUM, PAVETYP OFF
M_ERROR = M_ERROR + 1
ENDIF
IF LANEWID < 8 .OR. LANEWID > 15
  IF LANEWID <> 0
    DISPLAY FIELDS SID NO, STRUCNUM, LANEWID OFF
    M_ERROR = M_ERROR + 1
  ENDIF
ENDIF
IF OUTSHOWD < 0 .OR. OUTSHOWD > 12
  DISPLAY FIELDS SID NO, STRUCNUM, OUTSHOWD OFF
  M_ERROR = M_ERROR + 1
ENDIF
IF SHOSFTYP < 1 .OR. SHOSFTYP > 6
  DISPLAY FIELDS SID NO, STRUCNUM, SHOSFTYP OFF
  M_ERROR = M_ERROR + 1
ENDIF
IF SHOBSTYP < 21 .OR. SHOBSTYP > 45
  IF SHOBSTYP <> 0
    DISPLAY FIELDS SID NO, STRUCNUM, SHOBSTYP OFF
    M_ERROR = M_ERROR + 1
  ENDIF
ENDIF
IF WIDENFLG < 0 .OR. WIDENFLG > 2
  DISPLAY FIELDS SID NO, STRUCNUM, WIDENFLG OFF
  M_ERROR = M_ERROR + 1
ENDIF
IF SHOSFTHK < 0 .OR. SHOSFTHK > 10.1
  DISPLAY FIELDS SID NO, STRUCNUM, SHOSFTHK OFF
  M_ERROR = M_ERROR + 1
ENDIF
IF M_ERROR <> 0
  REPLACE ERRORCHAR WITH .T.
  M_ERROR = 0
ENDIF
SKIP
ENDDO

* reset parameters
N = 0
DO WHILE N < 6
  ? CHR(13)
  N = N + 1
ENDDO
CLEAR
SET ALIVE OFF
USE
RETURN

************
* PROCEDURE GEOSID
************
PROC SIDCHECK3
* TO LIST OUT ALL INCORRECT SID NUMBERS FOR GEOMETRIC & SHOULDER *

* SET PARAMETERS
set talk off
set echo off

USE \PAVEDB\EDITUPDT\GEOS_NEW INDEX \PAVEDB\EDITUPDT\GEOS_NEW

* SET PAGE HEADING
M_TTITLE = "GEOMETRIC & SHOULDER FILE - LIST OF INCORRECT SID NUMBERS"
M_TTITLE2 = "-------------------------------------------------------------"
SET ALTE ON
? M_TTITLE
? M_TTITLE2

STORE 0 TO VAR1, VAR2, VAR3, VAR4, COMPARE
DO WHILE .NOT. EOF()
  * calculate the correct check digit for the Sid number
  VAR1 = INT(SID_NO/1000)
  VAR2 = MOD(INT(SID_NO/100),10)
  VAR2 = VAR2 * 2
  VAR3 = MOD(INT(SID_NO/10),10)
  VAR3 = VAR3 * 3
  VAR4 = MOD(VAR1,10)+VAR2+VAR3
  COMPARE = MOD(VAR4,10)

  * compare the entered check digit with the calculated check digit
  IF COMPARE <> MOD(SID_NO,10)
    DISPLAY SID NO OFF
    REPLACE ERRORCHAR WITH .T.
    MSID = SID NO
    DO WHILE SID_NO = MSID .OR. (.NOT. EOF() )
      SKIP
    ENDDO
    SKIP -1
  ENDF
  IF SID NO = 0
    DISPLAY SID NO, STRUCTNUM OFF
    REPLACE ERRORCHAR WITH .T.
  ENDF
  SKIP
ENDDO

* reset parameters
? CHR(12)
CLEAR
SET ALTE OFF
USE
RETURN
PROGRAM LISTING

*  * SUBSYSTEM: EDIT & UPDATE INVENTORY FILES
*  * PROGRAM NAME: SURFCHK.PRG  06/08/88
*  * CALLED FROM: INV_UPDT.PRG
*  * PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
*  * TAMU/TTI
*  * AUTHOR: TREVOR X. PEREIRA
*  * PURPOSE: LISTS THE SURFACE FILE RECORDS THAT HAVE ERRORS
*

***************
* PROCEDURE TO CHECK SURFACE RECORDS IN LAYER FILE
***************
PROC SURFCHK1

* LISTS THE SURFACE FILE RECORDS THAT ARE NOT PRESENT IN
* THE MASTER LAYER IDENTIFICATION FILE
* set parameters
set talk off
set echo off
CLEAR

M_TTITLE  = "SURFACE FILE - LIST OF NEWLY ENTERED RECORDS NOT PRESENT IN LAYER ID FILE"
M_TTITLE2 = "_______________________________________________________________________________"
SET ALT ON
? M_TTITLE
? M_TTITLE2

* assign files to different work areas
SELECT 1
  USE \PAVEDB\FILES\LAYER INDEX \PAVEDB\INDEXES\LAYNDX

SELECT 2
  USE \PAVEDB\EDITUPDT\SURF_NEW INDEX \PAVEDB\EDITUPDT\SURF_NEW

DO WHILE .NOT. EOF()
  * store surface record in memory
  MSURFSID = SID_NO
  MSURFSTR = STRUCNUM
  MSURFIAY = IAYNUM

  * find the layer ID record that matches surface record
  SELECT 1
  SEEK STR(MSURFSID,4)+STR(MSURFSTR,2)+STR(MSURFIAY,2)
  IF .NOT. FOUND()
    SELECT 2
    DISPLAY FIELDS SID_NO, STRUCNUM, IAYNUM OFF

355
REPLACE ERRORCHAR WITH .T.
ELSE
SELE 2
ENDIF
SKIP
ENDDO

* reset parameters
N = 0
DO WHILE N < 6
? CHR(13)
N = N + 1
ENDDO
CLEAR
SET ALTE OFF
SELE 1
USE
SELE 2
USE
RETURN

**********************************************
* PROCEDURE TO CHECK OUT OF RANGE DATA
**********************************************
PROC SURFCHK3
*
* TO LIST ALL OUT OF RANGE DATA IN THE SURFACE FILE
*
* set parameters
SET TALK OFF
SET ECHO OFF

* print out page headings
M_TITLE = "SURFACE FILE - OUT OF RANGE RECORDS"
M_TITL2 = "-----------------------------------------"
M_ERROR = 0
SET ALTE ON
? M_TITLE
? M_TITL2

USE \PAVEDB\EDITUPT\SURF_NEW INDEX \PAVEDB\EDITUPT\SURF_NEW

* check data with ranges
DO WHILE .NOT. EOF()
IF STRUCNUM < 1 .OR. STRUCNUM > 9
   DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, STRUCNUM OFF
   M_ERROR = M_ERROR + 1
ENDIF
IF LAYNUM < 1 .OR. LAYNUM > 13
   DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYNUM OFF
   M_ERROR = M_ERROR + 1
ENDIF

356
IF AGAPPLRT < 30 .OR. AGAPPLRT > 200
    IF AGAPPLRT <> 0
      DISPLAY SID_NO, STRUCNUM, LAYNUM, AGAPPLRT OFF
      M_ERROR = M_ERROR + 1
    ENDIF
ENDIF
IF ALMXPER < 0 .OR. ALMXPER > 10.2
    DISPLAY SID_NO, STRUCNUM, LAYNUM, ALMXPER OFF
    M_ERROR = M_ERROR + 1
ENDIF
IF ASAPPLRT < .1 .OR. ASAPPLRT > .6
    IF ASAPPLRT <> 0
      DISPLAY SID_NO, STRUCNUM, LAYNUM, ASAPPLRT OFF
      M_ERROR = M_ERROR + 1
    ENDIF
ENDIF
IF M_ERROR <> 0
    REPLACE ERRORCHAR WITH .T.
    M_ERROR = 0
ENDIF
SKIP
ENDDO

* reset parameters
N = 0
DO WHILE N < 6
    ? CHR(13)
    N = N + 1
ENDDO
CLEAR
SET ALIVE OFF
USE
RETURN

*************************
* PROCEDURE TO CHECK SID NUMBER
*************************
PROC SIDCHEK4
*
* TO LIST ALL INCORRECT SID NUMBERS FOR SURFACE FILE
*
*
* set parameters
set talk off
set echo off
CLEAR
USE \PAVEDB\EDITUPDT\SURF_NEW INDEX \PAVEDB\EDITUPDT\SURF_NEW
*
print out page heading
M_TITLE   = "SURFACE FILE - INCORRECT SID NUMBER"
M_TITLE2  = "-----------------------------------"
SET ALTE ON
? M_TITLE
? M_TITLE2

* initialize variables
STORE 0 TO VAR1, VAR2, VAR3, VAR4, COMPARE

* calculate check digit of Sid number
DO WHILE .NOT. EOF()
  VAR1 = INT(SID_NO/1000)
  VAR2 = MOD(INT(SID_NO/100),10)
  VAR2 = VAR2 * 2
  VAR3 = MOD(INT(SID_NO/10),10)
  VAR3 = VAR3 * 3
  VAR4 = MOD(VAR1,10)+VAR2+VAR3
  COMPARE = MOD(VAR4,10)

  * compare calculated check digit with actual check digit
  IF COMPARE <> MOD(SID_NO,10)
    DISPLAY SID_NO, STRUCNUM, IAYNUM OFF
    REPLACE ERRORCHAR WITH .T.
  ENDF
  IF SID NO = 0
    DISPLAY SID NO, STRUCNUM, IAYNUM OFF
    REPLACE ERRORCHAR WITH .T.
  ENDF
  SKIP
ENDDO

* reset parameters
? CHR(12)
CLEAR
SET ALTE OFF
USE
RETURN
* SUBSYSTEM: EDIT & UPDATE INVENTORY FILES
* PROGRAM NAME: SUBGCHK.PRG 06/08/88
* CALLED FROM: INV UPDT.PRG
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TII
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: PROCEDURES TO LIST THE SUBGRADE FILE RECORDS THAT HAVE ERRORS
*

********************************************************************
* PROCEDURE SUBGCHK1
********************************************************************
PROCEDURE SUBGCHK1.
*
* TO LIST THE SUBGRADE FILE RECORDS THAT ARE NOT PRESENT
* IN THE MASTER LAYER IDENTIFICATION FILE
*
* set parameters
set talk off
set echo off
CLEAR

M_TITLE = " SUBGRADE FILE - LIST OF NEWLY ENTERED RECORDS NOT PRESENT IN LAYER ID FILE"
M_TITLE2 = "_____________________________________________________________________
SET ALIVE ON
? M_TITLE
? M_TITLE2

* assign files to different work areas
SELECT 1
  USE \PAVEDB\FILES\LAYER INDEX \PAVEDB\INDEXES\LAYNDX
SELECT 2
  USE \PAVEDB\EDITUPDT\SUBG_NEW INDEX \PAVEDB\EDITUPDT\SUBG_NEW

DO WHILE .NOT. EOF()
  MSUBGSID = SID_NO
  MSUBGLAY = " 1"
  MSUBGSTIR = " 1"

  * find the subgrade sid number in layer identification file
  SELECT 1
  seek STR(MSUBGSID,4)+MSUBGLAY+MSUBGSTIR
  IF .NOT. FOUND()
    SELECT 2
    DISPLAY FIELD SID_NO, STRUCNUM, LAYNUM OFF

359
REPLACE ERRORCHAR WITH .T.
ENDIF
SELECT 2
SKIP
ENDDO

* reset parameters
N = 0
DO WHILE N < 6
   ? CHR(13)
   N = N + 1
ENDDO
SET ALTE OFF
clear
SELE 1
USE
SELE 2
USE
RETURN

***************
* PROCEDURE TO CHECK OUT OF RANGE DATA
***************
PROC SUBGCHK3
*
* TO LIST OUT ALL DATA THAT IS OUT OF RANGE IN THE SUBGRADE FILE
*
*
* set parameters
SET TALK OFF
SET ECHO OFF
*
* print out page heading
M_TITLE = "SUBGRADE FILE - OUT OF RANGE RECORDS"
M_TITLE2 = "----------------------------------------"
M_ERROR = 0
SET ALTE ON
? M_TITLE
? M_TITLE2
USE \PAVEDB\EDITUPDT\SUBG_NEW INDEX \PAVEDB\EDITUPDT\SUBG_NEW
*
do a range check for all fields
DO WHILE .NOT. EOF()
   IF PLASTIX < 0 .OR. PLASTIX > 70
      IF PLASTIX <> 0
         DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, PLASTIX OFF
         M_ERROR = M_ERROR + 1
      ENDFIELD
   ENDFIELD
      IF LIQLIM < 10 .OR. LIQLIM > 80
         IF LIQLIM <> 0
            DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LIQLIM OFF
360
M_ERROR = M_ERROR + 1
ENDIF
ENDIF
IF TXTRIAXL < 1 .OR. TXTRIAXL > 6
   IF TXTRIAXL <> 0
      DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, TXTRIAXL OFF
      M_ERROR = M_ERROR + 1
   ENDIF
ENDIF
ENDIF
IF PERMIX < 0 .OR. PERMIX > 10
   DISPLAY SID NO, STRUCNUM, LAYNUM, PERMIX OFF
   M_ERROR = M_ERROR + 1
ENDIF
IF M_ERROR <> 0
   REPLACE ERRORCHAR WITH .T.
   M_ERROR = 0
ENDIF
SKIP
ENDDO
* reset parameters
N = 0
DO WHILE N < 6
   ? CHR(13)
   N = N + 1
ENDDO
SET ALTE OFF
clear
SELE 1
USE
SELE 2
USE
RETURN

***************
* PROCEDURE TO CHECK SID NUMBERS
***************
PROC SIDCHEK5
*
* LIST OF INCORRECT SID NUMBERS FOR THE SUBGRADE FILE
*
* set parameters
set talk off
set echo off
M_TTITLE = "SUBGRADE FILE - INCORRECT SID NUMBERS"
M_TTITLE2 = "------------------------------------------"
SET ALTE ON
? M_TTITLE
? M_TTITLE2
* initialize variables
STORE 0 TO VAR1, VAR2, VAR3, VAR4, COMPARE

* calculate correct Sid number check digit
DO WHILE .NOT. EOF()
    VAR1 = INT(SID_NO/1000)
    VAR2 = MOD(INT(SID_NO/100),10)
    VAR3 = VAR2 * 2
    VAR3 = MOD(INT(SID_NO/10),10)
    VAR4 = MOD(VAR1,10)+VAR2+VAR3
    COMPARE = MOD(VAR4,10)

    * compare calculated check digit with actual check digit
    IF COMPARE <> MOD(SID_NO,10)
        DISPLAY SID_NO, STRUCNUM, LAYNUM OFF
        REPLACE ERRORCHAR WITH .T.
    ENDIF
    IF SID_NO = 0
        DISPLAY SID_NO, STRUCNUM, LAYNUM OFF
        REPLACE ERRORCHAR WITH .T.
    ENDIF
    SKIP
ENDDO

* reset parameters
? CHR(12)
SET ALIKE OFF
clear
USE
RETURN
# METRIC (SI*) CONVERSION FACTORS

## APPROXIMATE CONVERSIONS TO SI UNITS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>When You Know</th>
<th>Multiply By</th>
<th>To Find</th>
</tr>
</thead>
<tbody>
<tr>
<td>in</td>
<td>inches</td>
<td>2.54</td>
<td>cm</td>
</tr>
<tr>
<td>ft</td>
<td>feet</td>
<td>0.3048</td>
<td>m</td>
</tr>
<tr>
<td>yd</td>
<td>yards</td>
<td>0.914</td>
<td>m</td>
</tr>
<tr>
<td>mi</td>
<td>miles</td>
<td>1.61</td>
<td>km</td>
</tr>
</tbody>
</table>

### LENGTH

<table>
<thead>
<tr>
<th>Symbol</th>
<th>When You Know</th>
<th>Multiply By</th>
<th>To Find</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm</td>
<td>millimetres</td>
<td>0.039</td>
<td>inches</td>
</tr>
<tr>
<td>m</td>
<td>metres</td>
<td>3.28</td>
<td>feet</td>
</tr>
<tr>
<td>m</td>
<td>metres</td>
<td>1.09</td>
<td>yards</td>
</tr>
<tr>
<td>km</td>
<td>kilometres</td>
<td>0.621</td>
<td>miles</td>
</tr>
</tbody>
</table>

### AREA

<table>
<thead>
<tr>
<th>Symbol</th>
<th>When You Know</th>
<th>Multiply By</th>
<th>To Find</th>
</tr>
</thead>
<tbody>
<tr>
<td>in²</td>
<td>square inches</td>
<td>645.2</td>
<td>cm²</td>
</tr>
<tr>
<td>ft²</td>
<td>square feet</td>
<td>0.0929</td>
<td>m²</td>
</tr>
<tr>
<td>yd²</td>
<td>square yards</td>
<td>0.836</td>
<td>m²</td>
</tr>
<tr>
<td>mi²</td>
<td>square miles</td>
<td>2.59</td>
<td>km²</td>
</tr>
</tbody>
</table>

### MASS (weight)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>When You Know</th>
<th>Multiply By</th>
<th>To Find</th>
</tr>
</thead>
<tbody>
<tr>
<td>oz</td>
<td>ounces</td>
<td>28.35</td>
<td>g</td>
</tr>
<tr>
<td>lb</td>
<td>pounds</td>
<td>0.454</td>
<td>kg</td>
</tr>
<tr>
<td>T</td>
<td>short tons (2000 lb)</td>
<td>0.907</td>
<td>Mg</td>
</tr>
</tbody>
</table>

### VOLUME

<table>
<thead>
<tr>
<th>Symbol</th>
<th>When You Know</th>
<th>Multiply By</th>
<th>To Find</th>
</tr>
</thead>
<tbody>
<tr>
<td>fl oz</td>
<td>fluid ounces</td>
<td>29.57</td>
<td>mL</td>
</tr>
<tr>
<td>gal</td>
<td>gallons</td>
<td>3.785</td>
<td>L</td>
</tr>
<tr>
<td>ft³</td>
<td>cubic feet</td>
<td>0.0328</td>
<td>m³</td>
</tr>
<tr>
<td>yd³</td>
<td>cubic yards</td>
<td>0.0765</td>
<td>m³</td>
</tr>
</tbody>
</table>

**NOTE:** Volumes greater than 1000 L shall be shown in m³.

### TEMPERATURE (exact)

<table>
<thead>
<tr>
<th>°C</th>
<th>°F</th>
<th>°F (after subtracting 32)</th>
<th>°C</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>81</td>
<td>5 (then add 32)</td>
<td>14</td>
</tr>
<tr>
<td>28</td>
<td>82</td>
<td>6 (then add 32)</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>°F</th>
<th>°C</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>0</td>
</tr>
</tbody>
</table>

*SI is the symbol for the International System of Measurements
CONTINUATION OF REPORT 456-1F Vol. II
Narrative on Changing Inventory Data

The Edit & Update program (EDITUPDT.PRG) calls CHNGLAYR.PRG to change the Inventory data. The inventory files include Location, Layer Identification, Layer Thickness Across the Road, Geometric and Shoulder, Surface and Subgrade. The user must make a backup of the master files first. In order to do this, CHNGLAYR.PRG calls the backup program CHNGBKUP.PRG.

The backup program backs up the master files to floppy disk as well as to the hard disk. It also creates 6 temporary dBASE files for the user to make changes to (LOCNCHNG.DBF, LAYRCCHNG.DBF, LAYTCCHNG.DBF, GEOSCHNG.DBF, SURFCHNG.DBF and SUBGCHNG.DBF).

After the changes have been completed, the Edit/Check programs must be run to flag any errors. The Edit/Check programs include CHEKLOCN.PRG, CHEKLAYR.PRG, CHEKLTAYT.PRG, CHEKGEOS.PRG, CHEKSURF.PRG and CHEKSUBG.PRG. If there are any errors present in the files, an error listing is printed out and the user can edit the data again to remove the errors. If no errors are present, the temporary files become the master files with the new changes in them. The original master files are deleted.

The programs and the temporary dBASE files for this section (CHANGE Inventory Data) are in the subdirectory \PAVEDB\EDITUPDT.

The inventory change process is illustrated in figures 16 through 18. Figure 16 depicts the change process on a global level, Figure 17 illustrates the high level program flow logic, and figure 18 charts the programs, procedures, and input and output files used in the inventory change process.
Inventory Data - Change Process

Inventory Changes

Create or Modify the Temp files

Master T F P D Files

Update Master DB or Create Error Rept

Temporary Inventory Files

Error Report

FIGURE 16
Inventory Data - Change Process
Program Flow Chart

Inventory Changes

Add Data Exists?
Y  Stop
N

Temp Change Files Exist?
Y  Create Temp Change Files
N

Change Data

Data With Errors?
Y  Error Reports
N

Update Master Database

Stop

FIGURE 17
PROGRAM SPECIFICATION

Program Name: CHNGLAYR.PRG

Purpose: To display the CHANGE Inventory data menu, Change the Data and to call the programs that Edit & Check the data.

Procedures\Edits:

The following are the procedures:
1) Backup the master files using CHNGBKUP.PRG.
2) Run the Edit/Check procedures to flag errors in the Changed data (CHEKLOCN.PRG, CHEKLAYR.PRG, CHEKLAYT.PRG, CHEKGEOS.PRG, CHEKSRF.PRG and CHEKSUBG.PRG).

Input\Output Files:

<table>
<thead>
<tr>
<th>Files</th>
<th>Indices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporary files -</td>
<td></td>
</tr>
<tr>
<td>LOCNLCHNG.DBF</td>
<td>LOCNCHNG.NDX</td>
</tr>
<tr>
<td>LAYRLCHNG.DBF</td>
<td>LAYRLCHNG.NDX</td>
</tr>
<tr>
<td>LAYTLCHNG.DBF</td>
<td>LAYTLCHNG.NDX</td>
</tr>
<tr>
<td>GEOSLCHNG.DBF</td>
<td>GEOSLCHNG.NDX</td>
</tr>
<tr>
<td>SURFPCCHNG.DBF</td>
<td>SURFPCCHNG.NDX</td>
</tr>
<tr>
<td>SUBGSCCHNG.DBF</td>
<td>SUBGSCCHNG.NDX</td>
</tr>
</tbody>
</table>

| Master Files -         |                  |
| LOCATION.DBF           | LOCST.D.NDX      |
| LAYER.DBF              | LAYNDX.NDX       |
| LAYTHICK.DBF           | LAYTNDX.NDX      |
| GEOSHD.DBF             | GEONDX.NDX       |
| SURFACE.DBF            | SURFNDX.NDX      |
| SUBGRADE.DBF           | SUBGNDX.NDX      |
Programs Called (See Program Flow Diagram):

dBASE programs
- CHNGKUP.PRG
- CHEKGEOS.PRG
- CHEKLAYR.PRG
- CHEKLAYT.PRG
- CHEKLOCN.PRG
- CHEKSUBG.PRG
- CHEKSURF.PRG

Screen Format Programs
- GEOSCHNG.FMT
- LAYRCHNG.FMT
- LAYTCHNG.FMT
- LOCNCHNG.FMT
- SUBGCHNG.FMT
- SURFCHNG.FMT
CHANGE INVENTORY SCREENS

The following screens are produced by CHNGLAVR.PRG and the screen format files except for the next menu (Edit & Update menu 3.0) which is produced by EDITUPDT.PRG.

<table>
<thead>
<tr>
<th>TEXAS FLEXIBLE PAVEMENT DATABASE</th>
<th>3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit &amp; Update</td>
<td></td>
</tr>
<tr>
<td>1 - Pavement Condition Data</td>
<td></td>
</tr>
<tr>
<td>2 - Inventory Data</td>
<td></td>
</tr>
<tr>
<td>3 - Traffic Data</td>
<td></td>
</tr>
<tr>
<td>4 - Tables</td>
<td></td>
</tr>
</tbody>
</table>

OPTION ➔

Choice 2 Inventory Data asks the user if he wants to ADD or CHANGE data to the inventory files. If he wants to CHANGE, CHNGLAVR.PRG program is run which produces the next screen (CHANGE Inventory Data 3.2.C).

<table>
<thead>
<tr>
<th>TEXAS FLEXIBLE PAVEMENT DATABASE</th>
<th>3.2.C</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDIT &amp; UPDATE</td>
<td></td>
</tr>
<tr>
<td>CHANGE Inventory Data</td>
<td></td>
</tr>
<tr>
<td>1 - Location</td>
<td></td>
</tr>
<tr>
<td>2 - Layer ID</td>
<td></td>
</tr>
<tr>
<td>3 - Geometric &amp; Shoulder</td>
<td></td>
</tr>
<tr>
<td>4 - Surface</td>
<td></td>
</tr>
<tr>
<td>5 - Subgrade</td>
<td></td>
</tr>
<tr>
<td>6 - Layer Thickness Across The Road</td>
<td></td>
</tr>
<tr>
<td>K - Check the Data Changed</td>
<td></td>
</tr>
</tbody>
</table>

OPTION ➔
Choice 1: Location displays the first screen on this page to let the user change Location data.
Choice 2: Layer ID displays the second screen on this page to let the user change add Layer data.

### Texas Flexible Pavement Database
#### Edit & Update - Change Inventory
##### Location File

<table>
<thead>
<tr>
<th>SID Number</th>
<th>District</th>
<th>County</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1</td>
<td>92</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Highway Ident.</th>
<th>Control/Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>US 82</td>
<td>45/4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mile Post</th>
<th>Lane Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.00 TO 24.00</td>
<td>R</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mile Point</th>
<th>HPMS Section Subdivision</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.000 TO 24.000</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HPMS Sample Number</th>
<th>Number of Lanes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Active</th>
<th>Inactive Date</th>
<th>Previous SID</th>
<th>Next SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>0/0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Functional Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comment</th>
</tr>
</thead>
</table>

To exit & keep changes, press **CTRL + End** keys simultaneously

### Texas Flexible Pavement Database
#### Edit & Update - Change Inventory
##### Layer Identification

<table>
<thead>
<tr>
<th>Structure Number</th>
<th>Layr No.</th>
<th>Layer Desc.</th>
<th>Center Thick</th>
<th>Material Type Class.</th>
<th>Job Compltd Mnth</th>
<th>Year</th>
<th>Widened Mnth Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>7</td>
<td>0.00</td>
<td>44</td>
<td>6</td>
<td>31</td>
<td>0 0</td>
</tr>
</tbody>
</table>

To exit & keep changes, press **CTRL + End** keys simultaneously
Choice 3  Geometric & Shoulder displays the first screen on this page to modify data.
Choice 4  Surface displays the second screen on this page to modify data.

**TEXAS FLEXIBLE PAVEMENT DATABASE**
**EDIT & UPDATE - CHANGE Inventory**
**Geometric & Shoulder Information**

<table>
<thead>
<tr>
<th>Sid Number</th>
<th>Structure Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1</td>
</tr>
</tbody>
</table>

Type of Pavement (See TTI Codes) 34
Lane Width (Feet) 12
Outside Shoulder Width (Feet) 0
Shoulder Surface Type 1
Shoulder Base Type (See Base Type Code, Table A.6) 0
Shoulder Surface Thickness (Inches) 0.0
Shoulder Base Thickness (Inches) 0.00
Widened Flag (0-2) 1

To exit & keep changes, press CTRL + End keys simultaneously

**TEXAS FLEXIBLE PAVEMENT DATABASE**
**EDIT & UPDATE - CHANGE Inventory**
**Surface Layer**

<table>
<thead>
<tr>
<th>Structure Number</th>
<th>Layer Number</th>
<th>Aggregate Application Rate</th>
<th>Type</th>
<th>Percent Admixture (Mean Asphalt Content)</th>
<th>Asphalt Application Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0 AC (S.Y./C.Y.)</td>
<td>5.70</td>
<td>0.00 (GAL/S.Y.)</td>
<td></td>
</tr>
</tbody>
</table>

To exit & keep changes, press CTRL + End keys simultaneously
Choice 5 Subgrade displays the first screen on this page to enter data into.
Choice 6 Layer Thickness Across The Road displays the second screen on this page to enter data into.

TEXAS FLEXIBLE PAVEMENT DATABASE
EDIT & UPDATE - CHANGE Inventory
Subgrade File

SID NUMBER 13

Percent Passing No. 200 Sieve 88.8
Texas Triaxial Class 5.3
Liquid Limit 64.5
Plasticity Index 40.4
Permeability Index 0.23

To exit & keep changes, press CTRL + End keys simultaneously

TEXAS FLEXIBLE PAVEMENT DATABASE
EDIT & UPDATE - CHANGE Inventory
Layer Thickness Across The Road

SID NUMBER 13

<table>
<thead>
<tr>
<th>Structure Number</th>
<th>Layer Number</th>
<th>Thickness - From Center</th>
<th>Distance From Center</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3rd Pos 2nd Pos 1st Pos</td>
<td>3rd Pos 2nd Pos 1st Pos</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In Inches</td>
<td>(In Feet)</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>10.00 10.00 6.00 6.00</td>
<td>12.0 9.0 5.0</td>
</tr>
</tbody>
</table>

To exit & keep changes, press CTRL + End keys simultaneously
PROGRAM LISTING

* GEOMETRIC & SHOULDER INFORMATION SCREEN FORMAT FILE - GEOSCHNG.FMT

@ 0, 0 CLEAR
@ 4, 22 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 5, 22 SAY "EDIT & UPDATE - CHANGE Inventory"
@ 6, 22 SAY "Geometric & Shoulder Information"
@ 8, 56 SAY "Sid Number "
@ 8, 67 SAY GEOSCHNG->SID_NO
@ 9, 50 SAY "Structure Number"
@ 9, 69 SAY GEOSCHNG->STRUCNUM PICTURE "99"
@ 11, 8 SAY "Type of Pavement (See TTI Codes)"
@ 11, 61 GET GEOSCHNG->PAVETYP PICTURE "99" RANGE 1,27
@ 12, 8 SAY "Lane Width (Feet)"
@ 12, 61 GET GEOSCHNG->LANEWID PICTURE "99" RANGE 8,15
@ 13, 8 SAY "Outside Shoulder Width (Feet)"
@ 13, 61 GET GEOSCHNG->OUTSHOWD PICTURE "99" RANGE 0,15
@ 14, 8 SAY "Shoulder Surface Type"
@ 14, 61 GET GEOSCHNG->SHOFTYP PICTURE "99" RANGE 1,6
@ 15, 8 SAY "Shoulder Base Type (See Base Type Code, Table A.6)"
@ 15, 61 GET GEOSCHNG->SHOBSTYP PICTURE "99" RANGE 21,45
@ 16, 8 SAY "Shoulder Surface Thickness (Inches)"
@ 16, 59 GET GEOSCHNG->SHOSFTHK PICTURE "99.9" RANGE 0.0,10.1
@ 17, 8 SAY "Shoulder Base Thickness (Inches)"
@ 17, 58 GET GEOSCHNG->SHOBSTHK PICTURE "99.99"
@ 18, 8 SAY "Wide Flag (0-2)"
@ 18, 62 GET GEOSCHNG->WIDENFLG PICTURE "9" RANGE 0,2
@ 3, 3 TO 20, 74 DOUBLE
@ 22, 10 SAY "To exit & keep changes, press CTRL + End keys simultaneously"
PROGRAM LISTING

* LAYER IDENTIFICATION SCREEN FORMAT FILE - LAYRCHNG.FMT

@ 0, 0 CLEAR
@ 3, 23 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 4, 23 SAY "EDIT & UPDATE - CHANGE Inventory"
@ 5, 23 SAY "Layer Identification"
@ 7, 55 SAY "Sid Number"
@ 7, 66 SAY LAYRCHNG->SID_NO PICTURE "9999"
@ 10, 54 SAY "Date"
@ 11, 4 SAY "
@ 12, 4 SAY "Structure Layr Layer Center Type Material Job Compltd Widened"
@ 13, 4 SAY "Number No. Desc. Thick Class. Mnth Year Mnth Year"
@ 14, 4 SAY "--------- ---- ----- ------ ------- ---- ---- ---
@ 15, 4 SAY "",

@ 16, 8 SAY LAYRCHNG->STRUCNUM PICTURE "9"
@ 16, 17 SAY LAYRCHNG->LAYNUM PICTURE "99"
@ 16, 24 GET LAYRCHNG->LAYDESC PICTURE "99" RANGE 1,14
@ 16, 30 GET LAYRCHNG->CENTTHK PICTURE "99.99" RANGE 0,25.0
@ 16, 40 GET LAYRCHNG->LAYMATCL PICTURE "99" RANGE 01,45
@ 16, 48 GET LAYRCHNG->JOBCMPMO PICTURE "99" RANGE 1,12
@ 16, 55 GET LAYRCHNG->JOBCMMPYR PICTURE "99" RANGE 20,99
@ 16, 62 GET LAYRCHNG->WIDENLMO PICTURE "99" RANGE 0,12
@ 16, 67 GET LAYRCHNG->WIDENLYR PICTURE "99" RANGE 0,99
@ 2, 2 TO 18, 74 DOUBLE
@ 22, 10 SAY "To exit & keep changes, press CTRL + End keys simultaneously"
PROGRAM LISTING

* LAYER THICKNESS ACROSS THE ROAD SCREEN FORMAT FILE - LAYTHCHG.FMT

@ 0, 0  CLEAR
@ 4, 23  SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 5, 23  SAY "EDIT & UPDATE - CHANGE Inventory"
@ 6, 23  SAY "Layer Thickness Across The Road"
@ 8, 56  SAY "SID NUMBER"
@ 8, 67  SAY LAYTHCHNG->SID NO PICTURE "9999"
@ 10, 25 SAY "Thickness - From Center Distance From Center"
@ 11, 3  SAY "Structure Layer" __________________________
@ 12, 4  SAY "Number Number 3rd Pos 2nd Pos 1st Pos Center 3rd Pos 2nd Pos 1st Pos"
@ 13, 3  SAY "---------- ------ ------- ------ ------- ------ -------"
@ 14, 6  SAY LAYTHCHNG->STRUCNUM PICTURE "99"
@ 14, 16  SAY LAYTHCHNG->LAYNUM PICTURE "99"
@ 14, 24  GET LAYTHCHNG->FC3THK PICTURE "99.99" RANGE 0.0,25.0
@ 14, 32  GET LAYTHCHNG->FC2THK PICTURE "99.99" RANGE 0.0,25.0
@ 14, 40  GET LAYTHCHNG->FC1THK PICTURE "99.99" RANGE 0.1,25.0
@ 14, 47  GET LAYTHCHNG->CENTTHK PICTURE "99.99" RANGE 0.1,25.0
@ 14, 56  GET LAYTHCHNG->FC3DIS PICTURE "99.9" RANGE 0.0,99.0
@ 14, 64  GET LAYTHCHNG->FC2DIS PICTURE "99.9" RANGE 0.0,99.0
@ 14, 72  GET LAYTHCHNG->FC1DIS PICTURE "99.9" RANGE 0.0,99.0
@ 15, 34  SAY "(In Inches)"
@ 15, 62  SAY "(In Feet)"
@ 2, 0  TO 17, 79  DOUBLE
@ 22, 10  SAY "To exit & keep changes, press CTRL + End keys simultaneously"
* LOCATION INFORMATION SCREEN FORMAT FILE - LOCNCHG.FMT

@ 2, 23 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 3, 23 SAY "EDIT & UPDATE - CHANGE Inventory"
@ 4, 23 SAY "Location File"
@ 6, 6 SAY "SID Number"
@ 6, 18 SAY LOCNCHG->SID_NO PICTURE "9999"
@ 6, 44 SAY "District"
@ 6, 53 GET LOCNCHG->HWYDIST PICTURE "99" RANGE 0,99
@ 6, 59 SAY "County"
@ 6, 66 GET LOCNCHG->CNTYNUM PICTURE "999" RANGE 0,999
@ 8, 6 SAY "Highway Ident."
@ 8, 21 GET LOCNCHG->HWYPREFIX
@ 8, 24 GET LOCNCHG->HWYNUM PICTURE "9999"
@ 8, 29 GET LOCNCHG->HWYSUFFIX
@ 8, 44 SAY "Control/Section"
@ 8, 60 GET LOCNCHG->CONTROL PICTURE "9999"
@ 8, 64 SAY "/"
@ 8, 65 GET LOCNCHG->SECTION PICTURE "99"
@ 10, 6 SAY "Mile Post"
@ 10, 17 GET LOCNCHG->BEGMPST PICTURE "999"
@ 10, 21 GET LOCNCHG->BEDISSIGN
@ 10, 23 GET LOCNCHG->EMPSTDIS PICTURE "99"
@ 10, 26 SAY "TO"
@ 10, 29 GET LOCNCHG->ENDMST PICTURE "999"
@ 10, 33 GET LOCNCHG->EDISSIGN
@ 10, 35 GET LOCNCHG->EMPSTDIS PICTURE "99"
@ 10, 44 SAY "Lane Identification"
@ 10, 64 GET LOCNCHG->LANEID
@ 12, 6 SAY "Mile Point"
@ 12, 17 GET LOCNCHG->BEGMPNT PICTURE "99.999"
@ 12, 24 SAY "TO"
@ 12, 27 GET LOCNCHG->ENDMPT PICTURE "99.999"
@ 12, 44 SAY "Mile Point Date"
@ 12, 61 GET LOCNCHG->MPNIMO PICTURE "99" RANGE 0, 12
@ 12, 63 SAY "/"
@ 12, 64 GET LOCNCHG->MPNYR PICTURE "99"
@ 14, 6 SAY "HPMS Sample Number"
@ 14, 25 GET LOCNCHG->HPMSSAM
@ 14, 44 SAY "HPMS Section Subdivision"
@ 14, 69 GET LOCNCHG->HPMSSEC PICTURE "9"
@ 16, 6 SAY "Functional Classification"
@ 16, 31 GET LOCNCHG->FUNCIAS PICTURE "99" RANGE 1, 7
@ 16, 44 SAY "Number of Lanes"
@ 16, 61 GET LOCNCHG->NUMLANES PICTURE "99" RANGE 1, 22
@ 18, 6 SAY "Active?"
@ 18, 15 GET LOCNCHG->ACTIVFLAG PICTURE "I"
@ 18, 19 SAY "Inactive Date"
@ 18, 33 GET LOCNCHG->INACTIMO PICTURE "99" RANGE 0, 12
@ 18, 35 SAY "/"
@ 18, 36 GET LOCNXING->INACTYR PICTURE "99"
@ 18, 41 SAY "Previous SID"
@ 18, 54 GET LOCNXING->PREVSID PICTURE "9999"
@ 18, 61 SAY "Next SID"
@ 18, 70 GET LOCNXING->NEXTSID PICTURE "9999"
@ 20, 6 SAY "Comment"
@ 20, 15 GET LOCNXING->COMMENT PICTURE
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 1, 2 TO 21, 77 DOUBLE
@ 22, 10 SAY "To exit & keep changes, press CTRL + End keys simultaneously"
PROGRAM LISTING

* SUBGRADE SCREEN FORMAT FILE - SUBGCHNG.FMT

@ 0, 0 CLEAR
@ 4, 23 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 5, 23 SAY "EDIT & UPDATE - CHANGE Inventory"
@ 6, 23 SAY "Subgrade File"
@ 7, 45 SAY "SID NUMBER"
@ 7, 56 SAY SUBGCHNG->SID NO PICTURE "9999"
@ 8, 19 SAY "Percent Passing No. 200 Sieve"
@ 11, 51 GET SUBGCHNG->[FSV200] PICTURE "99.9"
@ 12, 19 SAY "Texas Triaxial Class"
@ 12, 52 GET SUBGCHNG->[TXTRIAXL] PICTURE "9.9" RANGE 0.0,6.0
@ 13, 19 SAY "Liquid Limit"
@ 13, 51 GET SUBGCHNG->[LIQLIM] PICTURE "99.9" RANGE 0.0,80.0
@ 14, 19 SAY "Plasticity Index"
@ 14, 51 GET SUBGCHNG->[PLASTIX] PICTURE "99.9" RANGE 0.0,70.0
@ 15, 19 SAY "Permeability Index"
@ 15, 50 GET SUBGCHNG->[PERMIX] PICTURE "99.99" RANGE 0.00,10.00
@ 3, 10 TO 17, 65 DOUBLE
@ 22, 10 SAY "To exit & keep changes, press CTRL + End keys simultaneously"
PROGRAM LISTING

* SURFACE SCREEN FORMAT FILE - SURFCHNG.FMT

@ 0, 0 CLEAR
@ 4, 20 SAY " TEXAS FLEXIBLE PAVEMENT DATABASE"
@ 5, 20 SAY " EDIT & UPDATE - CHANGE Inventory"
@ 6, 20 SAY " Surface Layer"
@ 8, 57 SAY "SID NUMBER "
@ 8, 68 SAY SURFCHNG->SID_NO PICTURE "9999"
@ 10, 3 SAY "" Percent"
@ 11, 3 SAY " Aggregate Admixture
Asphalt"
@ 12, 3 SAY "Structure Layer Application Type (Mean Asphalt Application"
@ 13, 3 SA
* First check to see if there is any data in the transaction file. If
* there is data, display error message and go back to editupdt program.

SET ESCAPE OFF
CLEAR
MOUNT = 0
USE \PAVEDB\EDITUPDT\LOCN NEW INDEX \PAVEDB\EDITUPDT\LOCN_NEW
MOUNT = MOUNT + RECOUNT()
USE \PAVEDB\EDITUPDT\LAYR NEW INDEX \PAVEDB\EDITUPDT\LAYR_NEW
MOUNT = MOUNT + RECOUNT()
USE \PAVEDB\EDITUPDT\LAYT NEW INDEX \PAVEDB\EDITUPDT\LAYT_NEW
MOUNT = MOUNT + RECOUNT()
USE \PAVEDB\EDITUPDT\GEOS NEW INDEX \PAVEDB\EDITUPDT\GEOS_NEW
MOUNT = MOUNT + RECOUNT()
USE \PAVEDB\EDITUPDT\SURF NEW INDEX \PAVEDB\EDITUPDT\SURF_NEW
MOUNT = MOUNT + RECOUNT()
USE \PAVEDB\EDITUPDT\SUBG NEW INDEX \PAVEDB\EDITUPDT\SUBG_NEW
MOUNT = MOUNT + RECOUNT()
MCONTINUE = .T.
IF MOUNT <> 0
   CLEAR
   @ 4, 7 SAY "Data is present in the Temporary Inventory files. You have"
   @ 5, 7 SAY "to remove these records before you can change the data"
   @ 6, 7 SAY "in the master files."
   @ 8, 7 SAY "To remove the records,"
   @ 9, 7 SAY "1) Choose choice '3' from the Edit & Update Menu."
   @ 10, 7 SAY "2) Type 'A' to the next question to Add Data"
   @ 11, 7 SAY "3) Type 'E' to Edit the data in the Temporary Files."
   @ 12, 7 SAY "4) Now you can either"
   @ 13, 7 SAY "A) Delete each record by pressing Ctrl U and PgDn"
   @ 14, 7 SAY "OR"
   @ 15, 7 SAY "B) Correct the Data and run the Edit/Check programs"
   @ 16, 7 SAY "i.e. Choice 'K'."
   @ 17, 7 SAY "Press any key to continue . . ."
   WAIT ""
   CLEAR
   RETURN
ENDIF
MOUNT2 = 0
IF FILE('PAVDB\EDITUPDT\LOCNCHNG.DBF')
    MCOUNT2 = MCOUNT2 + 1
ENDIF
IF FILE('PAVDB\EDITUPDT\LOCNCHNG.NDX')
    MCOUNT2 = MCOUNT2 + 1
ENDIF
IF FILE('PAVDB\EDITUPDT\LAYRCHNG.DBF')
    MCOUNT2 = MCOUNT2 + 1
ENDIF
IF FILE('PAVDB\EDITUPDT\LAYRCHNG.NDX')
    MCOUNT2 = MCOUNT2 + 1
ENDIF
IF FILE('PAVDB\EDITUPDT\LAYTCHNG.DBF')
    MCOUNT2 = MCOUNT2 + 1
ENDIF
IF FILE('PAVDB\EDITUPDT\LAYTCHNG.NDX')
    MCOUNT2 = MCOUNT2 + 1
ENDIF
IF FILE('PAVDB\EDITUPDT\GEOCHNG.DBF')
    MCOUNT2 = MCOUNT2 + 1
ENDIF
IF FILE('PAVDB\EDITUPDT\GEOCHNG.NDX')
    MCOUNT2 = MCOUNT2 + 1
ENDIF
IF FILE('PAVDB\EDITUPDT\SURFCHNG.DBF')
    MCOUNT2 = MCOUNT2 + 1
ENDIF
IF FILE('PAVDB\EDITUPDT\SURFCHNG.NDX')
    MCOUNT2 = MCOUNT2 + 1
ENDIF
IF FILE('PAVDB\EDITUPDT\SUBGCHNG.DBF')
    MCOUNT2 = MCOUNT2 + 1
ENDIF
IF FILE('PAVDB\EDITUPDT\SUBGCHNG.NDX')
    MCOUNT2 = MCOUNT2 + 1
ENDIF
IF MCOUNT2 = 0
    DO 'PAVDB\EDITUPDT\CHNGBKUP
ENDIF
M_OK = 0
CLEAR
* Display the Inventory Change data menu on the screen
DO WHILE MCONTINUE
    STORE "" TO CHNGCOLL
    DO WHILE .NOT. (CHNGCOLL $ '123456K')
        @ 3, 21 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE 3.3.C"
        @ 4, 21 SAY "EDIT & UPDATE"
        @ 5, 21 SAY "CHANGE Inventory Data"
        @ 8, 15 SAY "1 - Location"
        @ 9, 15 SAY "2 - Layer ID"
        @ 10, 15 SAY "3 - Geometric & Shoulder"
        @ 11, 15 SAY "4 - Surface"
        @ 12, 15 SAY "5 - Subgrade"
        @ 13, 15 SAY "6 - Layer Thickness Across The Road"
@ 15, 15 SAY "K - Check the Data Changed"
@ 17, 40 SAY "OPTION " GET CHNGCOLL
@ 2, 9 TO 20, 65 DOUBLE
READ
IF READKEY() = 12
    MCONTINUE = .F.
    SET PROG TO EXIT
ENDIF
IF (CHNGCOLL $ '123456')
    @ 19, 23 SAY "Please enter SID Number " GET MSID NO PICTURE "9999"
    READ
    CLEAR
ENDIF
ENDDO
IF MCONTINUE
    CLEAR
DO CASE
    CASE CHNGCOLL = "1"
        USE \PAVEDB\EDITUPDT\LOCNCHNG INDE \PAVEDB\EDITUPDT\LOCNCHNG
        SEEK MSID NO
        SET FORMAT TO \PAVEDB\EDITUPDT\LOCNCHNG
        CHANGE
        SET FORMAT TO
    CASE CHNGCOLL = "2"
        USE \PAVEDB\EDITUPDT\LAYRCNCHNG INDE \PAVEDB\EDITUPDT\LAYRCNCHNG
        SEEK STR(MSID NO,4)
        SET FORMAT TO \PAVEDB\EDITUPDT\LAYRCNCHNG
        CHANGE
        SET FORMAT TO
    CASE CHNGCOLL = "3"
        USE \PAVEDB\EDITUPDT\GEOSCNCHNG INDE \PAVEDB\EDITUPDT\GEOSCNCHNG
        SEEK STR(MSID NO,4)
        SET FORMAT TO \PAVEDB\EDITUPDT\GEOSCNCHNG
        CHANGE
        SET FORMAT TO
    CASE CHNGCOLL = "4"
        USE \PAVEDB\EDITUPDT\SURFSCNCHNG INDE \PAVEDB\EDITUPDT\SURFSCNCHNG
        SEEK STR(MSID NO,4)
        SET FORMAT TO \PAVEDB\EDITUPDT\SURFSCNCHNG
        CHANGE
        SET FORMAT TO
    CASE CHNGCOLL = "5"
        USE \PAVEDB\EDITUPDT\SUBGCNCHNG INDE \PAVEDB\EDITUPDT\SUBGCNCHNG
        SEEK STR(MSID NO,4)
        SET FORMAT TO \PAVEDB\EDITUPDT\SUBGCNCHNG
        CHANGE
        SET FORMAT TO
    CASE CHNGCOLL = "6"
        USE \PAVEDB\EDITUPDT\LAYTCNCHNG INDE \PAVEDB\EDITUPDT\LAYTCNCHNG
        SEEK STR(MSID NO,4)
        SET FORMAT TO \PAVEDB\EDITUPDT\LAYTCNCHNG
        CHANGE
        SET FORMAT TO

383
CASE CHNGCOLL = "K"
   DO \PAVEDB\EDITUPDT\CHEKLOCN
   DO \PAVEDB\EDITUPDT\CHEKIAYR
   DO \PAVEDB\EDITUPDT\CHEKIAYT
   DO \PAVEDB\EDITUPDT\CHEKGEOS
   DO \PAVEDB\EDITUPDT\CHERSURF
   DO \PAVEDB\EDITUPDT\CHEKSUBG
   DO \PAVEDB\EDITUPDT\COPYLAYR
ENDCASE
CLEAR
ENDIF
CLEAR
ENDDO
RETURN
PROGRAM LISTING

* * SUBSYSTEM: EDIT & UPDATE
* PROGRAM NAME: CHGEBKUP.PRG 06/21/88
* CALLED FROM: CHNGLAYR.PRG
* MODIFIED ON: 09/20/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TII
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: BACK UP MASTER INVENTORY FILES (ORIGINAL FILES)
*

SET TALK OFF
SET ECHO OFF
SET ESCAPE OFF
CLEAR
MBACKDRV = MDRIVE2 + ":"
CLEAR TYPE
SET TYPE TO 0
STORE 0 TO MSIZE, MNUMFIELDS, MHEADER, MIOFALSIZ, MOCOUNT
STORE "" TO MNULL
@ 5,5 say "The MASTER files have to be backed up first."
@ 9, 5 say "Place a FORMATTED DISKETTE in drive " + MBACKDRV + " and close the drive."
@ 12,5 say "If you DO NOT want to continue, press the ESC key."
@ 13,5 say "Otherwise press any other key to continue."
READ
IF READKEY() = 12
   CLEAR
   MCONTINUE = .F.
   RETURN
ENDIF

*
* Backing up Location Data
*
CLEAR
MOVERWRITE = ""
SET DEFAULT TO &MBACKDRV
DO WHILE FILE('LOCATION.DBF')
   ? "LOCATION file exists on drive " + MBACKDRV
   ACCEPT "Overwrite it (Y/N) " TO MOVERWRITE
   IF MOVERWRITE $ "Y"
      DELE FILE LOCATION.DBF
   ELSE
      CLEAR
      ? "Replace diskette in drive " + MBACKDRV + " with another diskette"
      ? "Then press any key to continue"
      WAIT ""
   ENDIF
ENDDO
SET DEFAULT TO &MDRIVE

SET TALK OFF
* Checking diskspace on backup drive and backing up LOCATION master files
USE \PAVEDB\FILES\LOCATION
STORE RECOUNT() * RECSIZE() TO MSIZE
MNULL = ""
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
   MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MITOTALSIZE = MSIZE + MHEADER + 20
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDRIVE
DO WHILE MDISKSPACE < MITOTALSIZE
   ? "Not enough space on diskette in drive " + MBACKDRV
   ? "Please replace with another diskette and press any key to continue"
   WAIT ""
   IF READKEY() = 12
      ? "PROCESS ABORTED"
      RETURN
   ENDDO
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDRIVE
ENDDO
USE
? "Please wait. Backing up Location data files . . ."
IF FILE("\PAVEDB\FILES\backup\LOCATION.DBF")
   DELE FILE \PAVEDB\FILES\backup\LOCATION.DBF
ENDIF
SET TALK ON
COPY FILE \PAVEDB\FILES\LOCATION.DBF TO &MBACKDRV\LOCATION.DBF
COPY FILE \PAVEDB\FILES\LOCATION.DBF TO \PAVEDB\FILES\backup\LOCATION.DBF
SET TALK OFF

*
* backing up layer id files
*
CLEAR
MOVERWRITE = " "
SET DEFAULT TO &MBACKDRV
DO WHILE FILE('LAYER.DBF')
   ? "LAYER FILE exists on drive " + MBACKDRV
   ACCEPT "Overwrite it (Y/N) " TO MOVERWRITE
   IF MOVERWRITE $ "y"
      DELE FILE LAYER.DBF
   ELSE
      CLEAR
      ? "Replace diskette in drive " + MBACKDRV + " with another diskette"
      ? "Then press any key to continue"
      WAIT ""
   ENDIF
ENDIF
ENDDO
CLEAR
SET DEFAULT TO &MDRIVE

* Checking diskspace on backup drive and backing up Layer ID master files
USE \PAVEDB\FILES\LAYER
STORE RECOUNT() * RECSIZE() TO MSIZE
MNULL = ""
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MTOTALSIZE = MSIZE + MHEADER + 20
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDRIVE
DO WHILE MDISKSPACE < MTOTALSIZE
    ? "Not enough space on diskette in drive " + MBACKDRV
    ? "Please replace with another diskette and press any key to continue"
    WAIT ""
    IF READKEY() = 12
        ? "PROCESS ABORTED"
        RETURN
    ENDF
    SET DEFAULT TO &MBACKDRV
    MDISKSPACE = DISKSPACE()
    SET DEFAULT TO &MDRIVE
ENDDO
CLEAR
? "Please wait. Backing up Layer ID . . ."
USE
IF FILE('\pavedb\files\backup\LAYER.DBF')
    DELETE FILE \PAVEDB\FILES\BACKUP\LAYER.DBF
ENDIF
SET TALK ON
COPY FILE \PAVEDB\FILES\LAYER.DBF TO &MBACKDRV\LAYER.DBF
COPY FILE \PAVEDB\FILES\LAYER.DBF TO \PAVEDB\FILES\BACKUP\LAYER.DBF
SET TALK OFF

*
* backing up LAYTHICK THICKNESS files
*
CLEAR
MOVERWRITE = " "
SET DEFAULT TO &MBACKDRV
DO WHILE FILE('LAYTHICK.DBF')
    ? "LAYER THICKNESS FILE exists on drive " + MBACKDRV
    ACCEPT "Overwrite it (Y/N) " TO MOVERWRITE
    IF MOVERWRITE $ "$Y"
        DELETE FILE LAYTHICK.DBF
    ELSE
        CLEAR
        ? "Replace diskette in drive " + MBACKDRV + " with another diskette"
        ? "Then press any key to continue"
    ENDIF
ENDDO
WAIT " "
ENDIF
ENDDO
SET DEFAULT TO &MDRIVE

* Checking diskspace on backup drive and backing up LAYTHICK ID master files
USE \\PAVEDB\FILES\LAYTHICK
STORE RECOUNT() * RECSIZE() TO MSIZE
MNULL = ""
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
   MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MTOTALSIZE = MSIZE + MHEADER + 20
SET DEFAULT TO &MBEDRVR
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDRIVE
DO WHILE MDISKSPACE < MTOTALSIZE
   ? "Not enough space on diskette in drive " + MBACKDRV
   ? "Please replace with another diskette and press any key to continue"
   WAIT " "
   IF READKEY() = 12
      ? "PROCESS ABORTED"
      RETURN
   ENDDO
SET DEFAULT TO &MBEDRVR
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDRIVE
ENDDO
? "Please wait. Backing up LAYER THICKNESS Across the Road files . . ."
USE
IF FILE('\\pavedb\files\backup\LAYTHICK.DBF')
   DELE FILE \\PAVEDB\FILES\BACKUP\LAYTHICK.DBF
ENDIF
SET TALK ON
COPY FILE \\PAVEDB\FILES\LAYTHICK.DBF TO &MBEDRVR\LAYTHICK.DBF
COPY FILE \\PAVEDB\FILES\LAYTHICK.DBF TO \\PAVEDB\FILES\BACKUP\LAYTHICK.DBF
SET TALK OFF

*
* backing up GEOMETRIC & SHOULDER files
*
CLEAR
MOVEDWRITE = " "
SET DEFAULT TO &MBEDRVR
DO WHILE FILE('GEOSH. DBF')
   ? "GEOMETRIC & SHOULDER FILE exists on drive " + MBACKDRV
   ACCEPT "Overwrite it (Y/N) " TO MOVEDWRITE
   IF MOVEDWRITE $ "y"
      DELE FILE GEOSH. DBF
   ELSE
      CLEAR
      ? "Replace diskette in drive " + MBACKDRV + " with another diskette"
      ? "Then press any key to continue"
   ENDIF
ENDDO
WAIT " "
ENDIF
ENDDO
SET DEFAULT TO &MDRIVE

* Checking disk space on backup drive and backing up GEOMETRIC & SHOULDER master files
USE \PAVEDB\FILES\GEOSHO
STORE RECOUNT() * RECSIZE() TO MSIZE
MNULL = ""
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MIOTALSIZE = MSIZE + MHEADER + 20
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDRIVE
DO WHILE MDISKSPACE < MIOTALSIZE
    ? "Not enough space on diskette in drive " + MBACKDRV
    ? "Please replace with another diskette and press any key to continue"
    WAIT " "
    IF READKEY() = 12
        ? "PROCESS ABORTED"
        RETURN
    ENDIF
    SET DEFAULT TO &MBACKDRV
    MDISKSPACE = DISKSPACE()
    SET DEFAULT TO &MDRIVE
ENDDO
? "Please wait. Backing up GEOMETRIC & SHOULDER files . . ."
USE
IF FILE('\pavedb\files\backup\GEOSHO.DBF')
    DELE FILE \PAVEDB\FILES\BACKUP\GEOSHO.DBF
ENDIF
SET TALK ON
COPY FILE \PAVEDB\FILES\GEOSHO.DBF TO &MBACKDRV\GEOSHO.DBF
COPY FILE \PAVEDB\FILES\GEOSHO.DBF TO \PAVEDB\FILES\BACKUP\GEOSHO.DBF
SET TALK OFF

*
* backing up SURFACE THICKNESS files
*
CLEAR
MOVERNWRITE = " "
SET DEFAULT TO &MBACKDRV
DO WHILE FILE('SURFACE.DBF')
    ? "SURFACE FILE exists on drive " + MBACKDRV
    ACCEPT "Overwrite it (Y/N) " TO MOVERNWRITE
    IF MOVERNWRITE $ "Y"
        DELE FILE SURFACE.DBF
    ELSE
        CLEAR

389
? "Replace diskette in drive " + MBACKDRV + " with another diskette"
? "Then press any key to continue"
WAIT " "
ENDIF
ENDDO
CLEAR
SET DEFAULT TO &MDRIVE

* Checking disk space on backup drive and backing up SURFACE master files
USE \\PAVEDB\\FILES\\SURFACE
STORE RECCOUNT() * RECSIZE() TO MSIZE
MNULL = ""
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
   MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MTOTALSIZE = MSIZE + MHEADER + 20
SET DEFAULT TO &BACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDRIVE
DO WHILE MDISKSPACE < MTOTALSIZE
   ? "Not enough space on diskette in drive " + MBACKDRV
   ? "Please replace with another diskette and press any key to continue"
   WAIT " "
   IF READKEY() = 12
      ? "PROCESS ABORTED"
      RETURN
   ENDIF
SET DEFAULT TO &BACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDRIVE
ENDDO
USE
CLEAR
? "Please wait. Backing up SURFACE files . . ."
IF FILE('\\pavedb\\files\\backup\\SURFACE.DBF')
   DELE FILE \\PAVEDB\\FILES\\BACKUP\\SURFACE.DBF
ENDIF
SET TALK ON
COPY FILE \\PAVEDB\\FILES\\SURFACE.DBF TO &BACKDRV\\SURFACE.DBF
COPY FILE \\PAVEDB\\FILES\\SURFACE.DBF TO \\PAVEDB\\FILES\\BACKUP\\SURFACE.DBF
SET TALK OFF

*
* backing up SUBGRADE THICKNESS files
*
CLEAR
MOVERWRITE = " "
SET DEFAULT TO &BACKDRV
DO WHILE FILE('SUBGRADE.DBF')
   ? "SUBGRADE file exists on drive " + MBACKDRV
   ACCEPT "Overwrite it (Y/N) " TO MOVERWRITE
   IF MOVERWRITE $ "Y"
      DELE FILE SUBGRADE.DBF
   ENDIF
ENDIF
ELSE
  CLEAR
  ? "Replace diskette in drive " + MBACKDRV + " with another diskette"
  ? "Then press any key to continue"
  WAIT ""
ENDIF
ENDDO
CLEAR
SET TALK OFF
SET DEFAULT TO &MDRIVE

* Checking diskspase on backup drive and backing up SUBGRADE master files
USE \PAVEDB\FILES\SUBGRADE
STORE RECOOUNT() * RECSIZE() TO MSIZE
MNULL = ""
DO WHILE MNUL < FIELD(MNUMFIELDS + 1)
  MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MIOTALSIZE = MSIZE + MHEADER + 20
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
SET DEFAULT TO &MDRIVE
DO WHILE MDISKSPACE < MIOTALSIZE
  ? "Not enough space on diskette in drive " + MBACKDRV
  ? "Please replace with another diskette and press any key to continue"
  WAIT ""
  IF READKEY() = 12
    ? "PROCESS ABORTED"
    RETURN
  ENDDIF
  SET DEFAULT TO &MBACKDRV
  MDISKSPACE = DISKSPACE()
  SET DEFAULT TO &MDRIVE
ENDDO
USE
? "Please wait. Backing up SUBGRADE files . . ."
IF FILE('\pavedb\files\backup\SUBGRADE.DBF')
  DELE FILE \PAVEDB\FILES\BACKUP\SUBGRADE.DBF
ENDIF
SET TALK ON
COPY FILE \PAVEDB\FILES\SUBGRADE.DBF TO &MBACKDRV\SUBGRADE.DBF
COPY FILE \PAVEDB\FILES\SUBGRADE.DBF TO \PAVEDB\FILES\BACKUP\SUBGRADE.DBF
SET TALK OFF

CLEAR
? "Please wait. Setting up Files . . ."
SET TALK ON
SET SAFETY OFF
COPY FILE \PAVEDB\FILES\LOCATION.DBF TO \PAVEDB\EDITUPDT\LOCNCHNG.DBF
COPY FILE \PAVEDB\FILES\INDEXES\LCSTD.NDX TO \PAVEDB\EDITUPDT\LOCNCHNG.NDX
COPY FILE \PAVEDB\FILES\INDEXES\LAYER.DBF TO \PAVEDB\EDITUPDT\LAYNCHNG.DBF
COPY FILE \PAVEDB\FILES\INDEXES\LAYNDX.NDX TO \PAVEDB\EDITUPDT\LAYNCHNG.NDX
COPY FILE \PAVEDB\FILES\INDEXES\LAYTHICK.DBF TO \PAVEDB\EDITUPDT\LAYNCHNG.DBF
COPY FILE \PAVE\DB\INDEXES\LAYINDEX.NDX TO \PAVE\DB\EDITUPDT\LAYCHNG.NDX
COPY FILE \PAVE\DB\FILES\GEOSSHO.DBF TO \PAVE\DB\EDITUPDT\GEOSCHNG.DBF
COPY FILE \PAVE\DB\INDEXES\GEONDX.NDX TO \PAVE\DB\EDITUPDT\GEOSCHNG.NDX
COPY FILE \PAVE\DB\FILES\SURFACE.DBF TO \PAVE\DB\EDITUPDT\SURFCHNG.DBF
COPY FILE \PAVE\DB\INDEXES\SURFNDX.NDX TO \PAVE\DB\EDITUPDT\SURFCHNG.NDX
COPY FILE \PAVE\DB\FILES\SUBGRADE.DBF TO \PAVE\DB\EDITUPDT\SUBGCHNG.DBF
COPY FILE \PAVE\DB\INDEXES\SUBGNDX.NDX TO \PAVE\DB\EDITUPDT\SUBGCHNG.NDX

SET SAFETY ON
SET TALK OFF
SET ECHO OFF
SET TYPE TO 20
SET ESCAPE OFF
RETURN
* SUBSYSTEM: EDIT & UPDATE INVENTORY FILES
* PROGRAM NAME: CHEKGEOS.PRG  06/21/88
* CALLED FROM: CHNGLAYR.PRG
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TII
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: PROGRAM TO LIST RECORDS THAT ARE IN ERROR
*

* set parameters
set talk off
set echo off
CLEAR
M ERROR = 0
SET ALIVE TO GEOSERR

M TITLE = "GEOMETRIC & SHOULDER FILE - LIST OF RECORDS NOT IN LAYER ID FILE"
M TITLE2 = "-----------------------------------------------------------------------------"
SET ALIVE ON
? M TITLE
? M TITLE2

* assign database files to different work areas
SELECT 1
    USE \PAVEDB\EDITUPDT\LAYRCNG INDEX \PAVEDB\EDITUPDT\LAYRCNG

SELECT 2
    USE \PAVEDB\EDITUPDT\GEOSCHNG INDEX \PAVEDB\EDITUPDT\GEOSCHNG

SELE 2
DO WHILE .NOT. EOF()
    * store to memory variables
    MSID = SID NO
    MSTRUC = STRUCNUM

    SELECT 1
    * find the record in layer that equals the sid number and structure number
    SEEK STR(MSID,4)+STR(MSTRUC,2)

    * if record is not found, print the sid number
    IF .NOT. FOUND()
        SELE 2
        DISPLAY FIELD SID NO, STRUCNUM OFF
        M_ERROR = M_ERROR + 1
    ENDF  
    SELE 2
    SKIP
ENDDO
SELE 1
USE
SELE 2
USE
? " "
? " "
? " "
? " "
? " "
? " "

* TO LIST ALL OUT OF RANGE DATA FOR THE GEOMETRIC AND
* INFORMATION FILE
*

USE \PAVEDB\EDITUPDT\GBOSCHNG INDEX \PAVEDB\EDITUPDT\GBOSCHNG

* set up page heading
M TITLE = "GEOMETRIC & SHOULDER FILE - OUT OF RANGE RECORDS"
M TITL2 = "-----------------------------------------------"
? M TITLE
? M TITL2

* check data range
DO WHILE .NOT. EOF()
   IF STRUCNUM < 1 .OR. STRUCNUM > 9
      DISPLAY FIELDS SID NO, STRUCNUM OFF
      M ERROR = M ERROR + 1
   ENDIF
   IF PAVETYP < 1 .OR. PAVETYP > 37
      DISPLAY FIELDS SID NO, STRUCNUM, PAVETYP OFF
      M ERROR = M ERROR + 1
   ENDIF
   IF IANEWID < 8 .OR. IANEWID > 15
      IF IANEWID <> 0
         DISPLAY FIELDS SID NO, STRUCNUM, IANEWID OFF
         M ERROR = M ERROR + 1
      ENDIF
   ENDIF
   IF OUTSHOWD < 0 .OR. OUTSHOWD > 15
      DISPLAY FIELDS SID NO, STRUCNUM, OUTSHOWD OFF
      M ERROR = M ERROR + 1
   ENDIF
   IF SHOSFTYP < 1 .OR. SHOSFTYP > 6
      DISPLAY FIELDS SID NO, STRUCNUM, SHOSFTYP OFF
      M ERROR = M ERROR + 1
   ENDIF
   IF SHOBTYP < 21 .OR. SHOBTYP > 49
      IF SHOBTYP <> 0
         DISPLAY FIELDS SID NO, STRUCNUM, SHOBTYP OFF
         M ERROR = M ERROR + 1
      ENDIF
   ENDIF
ENDIF
IF WIDENFLG < 0 .OR. WIDENFLG > 2
   DISPLAY FIELDS SID NO, STRUCNUM, WIDENFLG OFF
   M_ERROR = M_ERROR + 1
ENDIF
IF SHOPTHK < 0 .OR. SHOPTHK > 10.1
   DISPLAY FIELDS SID NO, STRUCNUM, SHOPTHK OFF
   M_ERROR = M_ERROR + 1
ENDIF
SKIP
ENDDO
CLEAR
? " "
? " "
? " "
? " "
? " "
? " "

* TO LIST OUT ALL INCORRECT SID NUMBERS FOR GEOMETRIC & SHOULDER *

* SET PARAMETERS
set talk off
set echo off

* SET PAGE HEADING
M_TITLE = "GEOMETRIC & SHOULDER FILE - LIST OF INCORRECT SID NUMBERS"
M_TITLE2 = "---------------------------------------------------------------------"
? M_TITLE
? M_TITLE2

STORE 0 TO VAR1, VAR2, VAR3, VAR4, COMPARE
DO WHILE .NOT. EOF()
   * calculate the correct check digit for the Sid number
   VAR1 = INT(SID NO/1000)
   VAR2 = MOD(INT(SID_NO/100),10)
   VAR2 = VAR2 * 2
   VAR3 = MOD(INT(SID_NO/10),10)
   VAR3 = VAR3 * 3
   VAR4 = MOD(VAR1,10)+VAR2+VAR3
   COMPARE = MOD(VAR4,10)
   * compare the entered check digit with the calculated check digit
   IF COMPARE <> MOD(SID_NO,10)
      DISPLAY SID NO, STRUCNUM OFF
      M_ERROR = M_ERROR + 1
   ENDIF
   IF SID_NO = 0
      DISPLAY SID NO, STRUCNUM OFF
      M_ERROR = M_ERROR + 1
   ENDIF
   SKIP
ENDDO
* reset parameters
CLEAR
? ""
? ""
? ""
? ""
? ""
? ""

* TO LIST THE LAYER IDENTIFICATION RECORDS THAT ARE NOT
* PRESENT IN GEOMETRIC AND SHOULDER
*

* set parameters
CLEAR
M_ERROR1 = "GEOMETRIC & SHOULDER FILE - LIST OF ALL LAYER ID RECORDS"
M_ERROR2 = " NOT PRESENT IN THE GEOMETRIC & SHOULDER FILE"
M_ERROR3 = "----------------------------------"
? M_ERROR1
? M_ERROR2
? M_ERROR3

* assign files to different work areas
SELECT 1
  USE \PAVEDB\EDITUPDT\LAYRCHNG INDEX \PAVEDB\EDITUPDT\LAYRCHNG
SELECT 2
  USE \PAVEDB\EDITUPDT\GEOSCHNG INDEX \PAVEDB\EDITUPDT\GEOSCHNG

SELECT 1
DO WHILE .NOT. EOF()
  MSID = SID NO
  MSTRUC = STRUCNUM
  STRC_COMP = STRUCNUM

  * find the 1st layer identification record in geometric and shoulder
  * file
  SELECT 2
  Seek str(MSID,4)+str(MSTRUC,2)
  IF .NOT. FOUND()
    SELE 1
    DISPLAY FIELDS SID_NO, STRUCNUM OFF
    M_ERROR = M_ERROR + 1
  ENDF
  SELE 1
  SKIP

  * skip the rest of the records for the same sid number
  DO WHILE MSID = SID_NO .AND. STRC_COMP = STRUCNUM
    SKIP
  ENDDO
ENDDO
SELE 1
use
SELE 2
USE
USE \PAVEDB\EDITUPDT\GEOSCHNG INDEX \PAVEDB\EDITUPDT\GEOSCHNG
PACK
USE
CLEAR
? CHR(12)
SET ALTE OFF
CLOSE ALTE
IF M_ERROR <> 0
  M_OK = M_OK + 1
ENDIF
RETURN
* SUBSYSTEM: EDIT & UPDATE INVENTORY FILES
* PROGRAM NAME: CHEKILAYR.PRG  06/21/88
* CALLED FROM: CHNGILAYR.PRG
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE SYSTEM
* TAMU/TTI
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: CHECKS FOR ERRORS IN THE LAYER FILE
*

* set parameters
SET TALK OFF
SET ECHO OFF
CLEAR
SET ALTE TO LAYERRR
USE \PAVEDB\EDITUPDT\LAYRCHING INDEX \PAVEDB\EDITUPDT\LAYRCHING
M_ERROR = "LAYER ID FILE - INCORRECT LAYER DESCRIPTION OR MATERIAL CLASSIFICATION"
M_ERROR2 =

 SET ALTE ON
? M_ERROR
? M_ERROR2
M_ERRORCK = 0
* check layer description with layer material classification
DO WHILE .NOT. EOF()
   DO CASE
      CASE LAYDESC = 7
         IF LAYMATCL < 41 .OR. LAYMATCL > 45
            DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
            M_ERRORCK = M_ERRORCK + 1
         ENDF
      CASE LAYDESC = 6
         IF LAYMATCL < 21 .OR. LAYMATCL > 33
            IF LAYMATCL <> 17
               DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
               M_ERRORCK = M_ERRORCK + 1
            ENDF
         ENDF
      CASE LAYDESC = 5
         IF LAYMATCL < 21 .OR. LAYMATCL > 27
            IF LAYMATCL <> 17
               DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
               M_ERRORCK = M_ERRORCK + 1
            ENDF
         ENDF
      CASE LAYDESC = 4
         IF LAYMATCL < 1 .OR. LAYMATCL > 4
            DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
   ENDF
ENDDO
M_ERRORCK = M_ERRORCK + 1
ENDIF
CASE LAYDESC = 3
  IF LAYMATCL < 1 .OR. LAYMATCL > 17
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDIF
CASE LAYDESC = 2
  IF LAYMATCL <> 11
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDIF
CASE LAYDESC = 1
  IF LAYMATCL < 1 .OR. LAYMATCL > 16
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDIF
CASE LAYDESC = 10
  IF LAYMATCL < 5 .OR. LAYMATCL > 7
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDIF
CASE LAYDESC = 12
  IF LAYMATCL < 12 .OR. LAYMATCL > 15
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDIF
CASE LAYDESC = 14
  IF LAYMATCL <> 27
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC, LAYMATCL OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDIF
ENDCASE
ENDDO
CLEAR
? " "
? " "
? " "
? " "
? " "
? " "

* * VERIFIES THAT THE LAYER NUMBERS ARE IN CONSECUTIVE
* ORDER IN THE TRANSACTION FILE AND THAT NONE ARE MISSING
* GO TO TOP
M_ERROR = "LAYER ID FILE - MISSING OR INCORRECT LAYER NUMBERS"
M_ERROR2 = "-------------------------------------------------------------------"
? M_ERROR
? M_ERROR2
MSID NO = SID NO
MLAYNUM = 1
DO WHILE .NOT. EOF()
  * checks for consecutive sid numbers
  DO WHILE .NOT. EOF() .AND. MSID_NO = SID_NO
    IF LAYNUM <> MLAYNUM
      IF LAYDESC = 13 .OR. LAYDESC = 12
        SKIP
        STORE LAYNUM TO MLAYNUM
        IF MSID_NO <> SID_NO
          SKIP -1
        ENDF
      ELDIF
    DISPLAY SID NO, STRUCNUM, LAYNUM OFF
    M_ERRORCK = M_ERRORCK + 1
    STORE LAYNUM TO MLAYNUM
  ENDF
  MIAYNUM = MLAYNUM + 1
  SKIP
ENDDO
MSID_NO = SID_NO
MLAYNUM = 1
ENDDO
CLEAR
? " "
? " "
? " "
? " "
? " "
? " "
? " 

* TO LIST OUT OF RANGE DATA IN THE LAYER IDENTIFICATION
* TRANSACTION FILE
* GOTO TOP
M_ERROR = "LAYER ID FILE - OUT OF RANGE DATA"
M_ERROR2 = "-----------------------------------------"
? M_ERROR
? M_ERROR2

* check the data
DO WHILE .NOT. EOF()
  IF STRUCNUM < 1 .OR. STRUCNUM > 9
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDF
  IF LAYNUM < 1 .OR. LAYNUM > 13
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDF
  IF LAYDESC < 1 .OR. LAYDESC > 14
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYDESC OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDF
IF CENTHK < .1 .OR. CENTHK > 25
  IF LAYNUM <> 1 .AND. CENTHK = 0
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, CENTHK OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDF
ENDIF
ENDIF
IF LAYMATCL < 1 .OR. LAYMATCL > 45
  DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYMATCL OFF
  M_ERRORCK = M_ERRORCK + 1
ENDIF
IF JOBCMPYR < 20 .OR. JOBCMPYR > 99
  DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, JOBCMPYR OFF
  M_ERRORCK = M_ERRORCK + 1
ENDIF
IF JOBCMPMO < 1 .OR. JOBCMPMO > 12
  DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, JOBCMPMO OFF
  M_ERRORCK = M_ERRORCK + 1
ENDIF
IF WIDENLYR < 28 .OR. WIDENLYR > 99
  IF WIDENLYR <> 0
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, WIDENLYR OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDF
ENDIF
ENDIF
IF WIDENIMO < 1 .OR. WIDENIMO > 12
  IF WIDENIMO <> 0
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, WIDENIMO OFF
    M_ERRORCK = M_ERRORCK + 1
  ENDF
ENDIF
ENDDO
SKIP
CLEAR
? ""
? ""
? ""
? ""
? ""
? ""

*
* LIST OUT SID NUMBERS THAT ARE INCORRECT
*

clear
GOTO TOP
M_ERROR = "LAYER ID FILE - INCORRECT SID NUMBERS"
M_ERROR2 = "---------------------------------------------"
? M_ERROR
? M_ERROR2

* initialize variables
STORE 0 TO VAR1, VAR2, VAR3, VAR4, COMPARE
* calculate correct Sid Number check digit
DO WHILE .NOT. EOF()
   VAR1 = INT(SID_NO/1000)
   VAR2 = MOD(INT(SID_NO/100),10)
   VAR2 = VAR2 * 2
   VAR3 = MOD(INT(SID_NO/10),10)
   VAR3 = VAR3 * 3
   VAR4 = MOD(VAR1,10)+VAR2+VAR3
   COMPARE = MOD(VAR4,10)
   * compare actual check digit with calculated check digit
   IF COMPARE <> MOD(SID NO,10)
      DISPLAY SID NO, STRUCNUM, LAYNUM OFF
      M_ERRORCK = M_ERRORCK + 1
   ENDIF
   IF SID NO = 0
      DISPLAY SID NO, STRUCNUM, LAYNUM OFF
      M_ERRORCK = M_ERRORCK + 1
   ENDIF
   SKIP
ENDDO
?
?
?
?
?
?
?
?
?
?

* LIST OUT LAYER RECORDS THAT DO NOT HAVE A LOCATION RECORD
*
clear
GOTO TOP
SELE 2
USE \PAVEDB\EDITUPDT\LOCNCHG INDE \PAVEDB\EDITUPDT\LOCNCHG
M_ERROR = "LAYER ID FILE - SID NUMBERS THAT DO NOT HAVE A LOCATION RECORD"
M_ERROR2 = "---------------------------------------------------------------------"
SELE 1
? M_ERROR
? M_ERROR2
DO WHILE .NOT. EOF()
   MSID_NO = A->SID_NO
   SELE 2
   SEEK MSID NO
   IF .NOT. FOUND()
      SELE 1
      DISPLAY SID_NO, STRUCNUM, LAYNUM OFF
      M_ERRORCK = M_ERRORCK + 1
   ENDFOR
   SELE 1
   DO WHILE A->SID_NO = MSID_NO .OR. (.NOT. EOF())
      SKIP
   ENDDO
ENDDO
sele 1
use
sele 2
use
USE \PAVEDB\EDITUPDT\LAYRCHNG INDEX \PAVEDB\EDITUPDT\LAYRCHNG
PACK
USE
? CHR(12)
SET ALITE OFF
CLOSE ALITE
IF M_ERRORCK <> 0
   M_OK = M_OK + 1
ENDIF
CLEAR
RETURN
* SUBSYSTEM: EDIT & UPDATE INVENTORY FILES
* PROGRAM NAME: CHEKIATY.PRG 06/21/88
* CALLED FROM: CHINGLAY.R.PRG
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TTI
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO LIST ERRORS IN THE LAYER THICKNESS ACROSS THE ROAD FILE
*

* set parameters
Set talk off
set echo off
CLEAR

* print out page heading
SET ALINE TO LAYTERR
SET ALINE ON
MITTLE = "LAYER THICKNESS FILE - LIST OF RECORDS NOT PRESENT IN LAYER ID FILE"
MITTLE2 = "---------------------------------------------"
? MITTLE
? MITTLE2
* assign files to different work areas
SELECT 1
   USE \PAVEDB\EDITUPDT\LAYRCHNG INDEX \PAVEDB\EDITUPDT\LAYRCHNG
SELECT 2
   USE \PAVEDB\EDITUPDT\LAYCHNG INDEX \PAVEDB\EDITUPDT\LAYCHNG
M ERROR = 0
DO WHILE .NOT. EOF()
   MLAYSID = SID NO
   MLAYSTRC = STURCNUM
   MLAYNUM = LAYNUM

   * find the laythick file record in layer file
   SELECT 1
   seek str(MLAYSID,4)+str(MLAYSTRC,2)+str(MLAYNUM,2)
   IF .NOT. FOUND()
      SELECT 2
      DISPLAY FIELDS SID NO, STURCNUM, LAYNUM OFF
      M ERROR = M ERROR + 1
   ENDIF
   SELE 2
   SKIP
ENDDO
SELE 1
USE
SELE 2
USE
CLEAR
? " "
? " "
? " "
? " "
? " "

* TO LIST ALL OUT OF RANGE DATA IN THE LAYER THICKNESS
* ACROSS THE ROAD FILE
*

USE \PAVEDB\EDITUPDT\LAYTCHNG INDEX \PAVEDB\EDITUPDT\LAYTCHNG
GOTO TOP

* print out the page heading
M_TTITLE = "LAYER THICKNESS FILES - OUT OF RANGE RECORDS"
M_TTITLE2 = "----------------------------------"
? M_TTITLE
? M_TTITLE2

* check the ranges
DO WHILE .NOT. EOF()
    IF STRUCNUN < 1 .OR. STRUCNUN > 9
        DISPLAY FIELDS SID NO, STRUCNUN, LAYNUM, STRUCNUN OFF
        M_ERROR = M_ERROR + 1
    ENDIF
    IF IAYNUM < 1 .OR. IAYNUM > 13
        DISPLAY FIELDS SID NO, STRUCNUN, LAYNUM, IAYNUM OFF
        M_ERROR = M_ERROR + 1
    ENDIF
    IF FC3THK < 0 .OR. FC3THK > 25
        DISPLAY FIELDS SID NO, STRUCNUN, LAYNUM, FC3THK OFF
        M_ERROR = M_ERROR + 1
    ENDIF
    IF FC2THK < 0 .OR. FC2THK > 25
        DISPLAY FIELDS SID NO, STRUCNUN, LAYNUM, FC2THK OFF
        M_ERROR = M_ERROR + 1
    ENDIF
    IF FC1THK < 1 .OR. FC1THK > 25
        DISPLAY FIELDS SID NO, STRUCNUN, LAYNUM, FC1THK OFF
        M_ERROR = M_ERROR + 1
    ENDIF
    IF CEN3THK < 1 .OR. CEN3THK > 25
        DISPLAY FIELDS SID NO, STRUCNUN, LAYNUM, CEN3THK OFF
        M_ERROR = M_ERROR + 1
    ENDIF
    IF FC1DIS < .01 .OR. FC1DIS > 99
        DISPLAY FIELDS SID NO, STRUCNUN, LAYNUM, FC1DIS OFF
        M_ERROR = M_ERROR + 1
    ENDIF
    IF FC2DIS < FC1DIS
        IF FC2DIS <> 0
            DISPLAY FIELDS SID NO, STRUCNUN, LAYNUM, FC2DIS OFF

405
M_ERROR = M_ERROR + 1
ENDIF
ENDIF
IF FC3DIS < FC2DIS
  IF FC3DIS <> 0
    DISPLAY SID NO, STRUCNUM, LAYNUM, FC3DIS OFF
    M_ERROR = M_ERROR + 1
  ENDF
ENDIF
SKIP
ENDDO
CLEAR
?
?
?
?
?
?
?
?
?

*  
* TO CHECK FOR INVALID SID NUMBERS  
* 
GO TO TOP
* set up page heading
MITLE1 = "LAYER THICKNESS FILE - INCORRECT SID NUMBERS"
MITLE2 = "---------------------------------------------"
? MITLE1
? MITLE2
STORE 0 TO VAR1, VAR2, VAR3, VAR4, COMPARE

* calculate check digit
DO WHILE .NOT. EOF()
  VAR1 = INT(SID_NO/1000)
  VAR2 = MOD(INT(SID_NO/100),10)
  VAR2 = VAR2 * 2
  VAR3 = MOD(INT(SID_NO/10),10)
  VAR3 = VAR3 * 3
  VAR4 = MOD(VAR1,10)+VAR2+VAR3
  COMPARE = MOD(VAR4,10)

  * compare calculated check digit with actual check digit
  IF COMPARE <> MOD(SID_NO,10)
    DISPLAY SID NO, STRUCNUM, LAYNUM OFF
    M_ERROR = M_ERROR + 1
  ENDF
  IF SID NO = 0
    DISPLAY SID NO, STRUCNUM, LAYNUM OFF
    M_ERROR = M_ERROR + 1
  ENDF
  SKIP
ENDDO
* reset parameters
CLEAR
?
""
* TO LIST LAYER IDENTIFICATION FILE RECORDS THAT ARE NOT
* PRESENT IN LAYER THICKNESS ACROSS THE ROAD FILE
*
M.Title  = "LAYER THICKNESS FILE - LIST OF ALL RECORDS IN MASTER LAYER ID FILE"
M.Title2 = "THAT ARE NOT PRESENT IN THE LAYER THICKNESS FILE"
M.Title3 = "-----------------------------------------------------------------------------------"
?
?
?
?
?
?
*
assign files to different work areas
SELECT 1
  USE \PAVDB\EDITUPDT\LAYrchng INDEX \PAVDB\EDITUPDT\LAYrchng
SELECT 2
  USE \PAVDB\EDITUPDT\LAYrchng INDEX \PAVDB\EDITUPDT\LAYrchng
SELECT 1
DO WHILE NOT EOF()
  MSID = SID_NO
  MSTR = STRUCNUM
  MLAY = LAYNUM

  * if layer number is 1, skip to the next record in layer file
  IF LAYNUM = 1
    SKIP
  LOOP
ENDIF

  * find the layer record in laythick file
  SELECT 2
  SEEK STR(MSID,4)+STR(MSTR,2)+STR(MLAY,2)
  IF NOT. FOUND()
    SELECT 1
    DISPLAY FIELDS SID_NO, STRUCNUM, LAYNUM OFF
    M_ERROR = M_ERROR + 1
  ENDIF
  SELECT 1
  SKIP
ENDDO

CLEAR
? CHR(12)
SET ALTE OFF
SELE 1
use
SELE 2
USE
USE \PAVEDB\EDITUPDT\LAYTCHNG INDEX \PAVEDB\EDITUPDT\LAYTCHNG
PACK
USE
CLOSE ALTE
IF M_ERROR <> 0
   M_OK = M_OK + 1
ENDIF
RETURN
SET TALK OFF
CLEAR
USE \PAVEDB\EDITUPDT\LOCNCHNG INDE \PAVEDB\EDITUPDT\LOCNCHNG
M ERROR = "LOCATION FILE - INCORRECT DATA ENTERED"
M ERROR2 = "-----------------------------"
SET ALIKE TO LOCNERR
SET ALIKE ON
MERRORCNT = 0
? M ERROR
? M ERROR2
DO WHILE .NOT. EOF()
   MSID NO = SID NO
   * calculates the correct Sid Number
   STORE 0 TO VAR1, VAR2, VAR3, COMPARE
   VAR1 = INT(MSID NO/1000)
   VAR2 = MOD(INT(MSID NO/100),10)
   VAR2 = VAR2 * 2
   VAR3 = MOD(INT(MSID NO/10),10)
   VAR3 = VAR3 * 3
   VAR4 = MOD(VAR1,10)+VAR2+VAR3
   COMPARE = MOD(VAR4,10)
   * compares calculated Sid number with Sid number entered
   IF COMPARE <> MOD(MSID NO,10)
      MERRORCNT = MERRORCNT + 1
      DISPLAY FIELD SID NO OFF
      ? "SID NUMBER IS NOT VALID"
      ? ""
      ? ""
   ENDFI
   * verifies that Sid number has been entered
   IF MSID NO = 0
      MERRORCNT = MERRORCNT + 1
      DISPLAY FIELD SID NO OFF
      ? "SID NUMBER IS NOT VALID"
      ? ""
      ? ""
   ENDFI
IF HWYDIST < 0 .OR. HWYDIST > 27
  DISPLAY FIELDS SID NO, HWYDIST OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
IF CNTYNUM < 0 .OR. CNTYNUM > 270
  DISPLAY FIELDS SID NO, CNTYNUM OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
DO CASE
  CASE HWYPREFIX = "FM"
  CASE HWYPREFIX = "SH"
  CASE HWYPREFIX = "IH"
  CASE HWYPREFIX = "US"
  CASE HWYPREFIX = "SP"
  CASE HWYPREFIX = "LE"
  CASE HWYPREFIX = "PR"
  OTHERWISE
    DISPLAY FIELDS SID NO, HWYPREFIX OFF
    MERRORCNT = MERRORCNT + 1
  ENDCASE
IF HWNUM = 0
  DISPLAY FIELDS SID NO, HWNUM OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
IF EDISSIGN = "+" .OR. EDISSIGN = "-"
ELSE
  DISPLAY FIELDS SID NO, EDISSIGN OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
IF EDISSIGN = "+" .OR. EDISSIGN = "-"
ELSE
  DISPLAY FIELDS SID NO, EDISSIGN OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
IF LANEID = "R" .OR. LANEID = "L"
ELSE
  DISPLAY FIELDS SID NO, LANEID OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
IF BEGMENT = 0 .AND. ENDMNT = 0
  DISPLAY FIELDS SID NO, BEGMENT, ENDMNT OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
IF MPRIMO < 0 .OR. MPRIMO > 12
  DISPLAY FIELDS SID NO, MPRIMO OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
IF NUMLANES < 1 .OR. NUMLANES > 22
  DISPLAY FIELDS SID NO, NUMLANES OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
IF PREVSID <> 0
  MPREVSID = PREVSID
  MBEGMPST = BEGMPST
MBDISSIGN = BDISSIGN
MBPSTDIS = BMPSTDIS
MENDMST = ENDMST
MEDISSIGN = EDISSIGN
MEMPSTDIS = EMPSTDIS
SEEK MPREVSID
IF .NOT. FOUND()
    SEEK MSID NO
    DISPLAY FIELD SID NO, PREVSID OFF
    ? " SID Number NOT present in location file"
    ? " "
    ? " "
    MERRORCNT = MERRORCNT + 1
ELSE
    IF MEBGMPST <> EBGMPST .OR. MBDISSIGN <> BDISSIGN .OR. MEPSTDIS <>
    EEMPSTDIS
        SEEK MSID NO
        DISPLAY FIELD SID NO, PREVSID OFF
        ? " SID Number and PREVIOUS SID Beginning Mile Post do not match"
        ? " "
        ? " "
        MERRORCNT = MERRORCNT + 1
    ENDIF
    SEEK MPREVSID
    IF MENDMST <> EENDMST .OR. MEDISSIGN <> EDISSIGN .OR. MEPSTDIS <>
    EEMPSTDIS
        SEEK MSID NO
        DISPLAY FIELD SID NO, PREVSID OFF
        ? " SID Number and PREVIOUS SID Beginning Mile Post do not match"
        ? " "
        ? " "
        MERRORCNT = MERRORCNT + 1
    ENDIF
    ENDIF
    SEEK MSID NO
    ENDIF
    IF NEXSID <> 0
        MNEXTSID = NEXSID
        SEEK MNEXTSID
        IF .NOT. FOUND()
            SEEK MSID NO
            DISPLAY FIELD SID NO, NEXSID OFF
            ? " SID Number NOT present in location file"
            ? " "
            ? " "
            MERRORCNT = MERRORCNT + 1
        ENDIF
        SEEK MSID NO
        ENDIF
    IF CONTROL = 0
        DISPLAY FIELDS SID NO, CONTROL OFF
        MERRORCNT = MERRORCNT + 1
    ENDIF
    IF SECTION = 0
DISPLAY FIELDS SID NO, SECTION OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
IF FUNCAS < 0 .OR. FUNCAS > 7
  DISPLAY FIELDS SID NO, FUNCAS OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
IF ENDMPST = 0
  DISPLAY FIELDS SID NO, ENDMPT OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
IF EMPSTDIS < 0
  DISPLAY FIELDS SID NO, EMPSTDIS OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
IF EMPSTDIS < 0
  DISPLAY FIELDS SID NO, EMPSTDIS OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
IF MNTYR < 20 .OR. MNTYR > 99
  DISPLAY FIELDS SID NO, HWYNUM OFF
  MERRORCNT = MERRORCNT + 1
ENDIF
IF .NOT. ACTIVFLAG
  IF INACTYR < 50 .OR. INACTYR > 99
    DISPLAY FIELDS SID NO, INACTYR OFF
    MERRORCNT = MERRORCNT + 1
  ENDIF
  IF INACIMO < 0 .OR. INACIMO > 12
    DISPLAY FIELDS SID NO, INACIMO OFF
    MERRORCNT = MERRORCNT + 1
  ENDIF
ENDIF
IF ACTIVFLAG
  IF INACTYR <> 0
    DISPLAY FIELDS SID NO, INACTYR OFF
    ? " " Inactive Flag and Inactive YEAR are inconsistent"
    ? " "
    ? " "
    MERRORCNT = MERRORCNT + 1
  ENDIF
  IF INACIMO <> 0
    DISPLAY FIELDS SID NO, INACIMO OFF
    ? " " Inactive Flag and Inactive MONTH are inconsistent"
    ? " "
    ? " "
    MERRORCNT = MERRORCNT + 1
  ENDIF
  IF NEXTSID <> 0
    DISPLAY FIELDS SID NO, NEXTSID OFF
    ? " " Cannot point to Another SID number when present SID number is
    ? " "
    ? " "

412
MERRORCNT = MERRORCNT + 1
ENDIF
ENDIF
SKIP
ENDDO
use
USE \PAVEDB\EDITUPDT\LOCNCHNG INDE \PAVEDB\EDITUPDT\LOCNCHNG PACK
USE
* reset parameters
CLEAR
? CHR(12)
SET AITE OFF
CLOSE AITE
IF MERRORCNT <> 0
  M_OK = M_OK + 1
ENDIF
RETURN
* * SUBSYSTEM: EDIT & UPDATE INVENTORY FILES
* * PROGRAM NAME: CHEKSUBG.PRG  06/21/88
* * CALLED FROM: CHINGLAYR.PRG
* * PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* * TAMU/TTI
* * AUTHOR: TREVOR X. PEREIRA
* * PURPOSE: TO LIST THE SUBGRADE FILE RECORDS THAT HAVE ERRORS
*

* TO LIST THE SUBGRADE FILE RECORDS THAT ARE NOT PRESENT
* IN THE MASTER LAYER IDENTIFICATION FILE
* *
* set parameters
set talk off
set echo off
SET ALITE TO SUBGERR
CLEAR
M_ERROR = 0

M_TITLE = " SUBGRADE FILE - LIST OF RECORDS NOT PRESENT IN LAYER ID FILE"
M_TITLE2 = " "
SET ALITE ON
? M_TITLE
? M_TITLE2

* assign files to different work areas
SELECT 1
USE \PAVEDB\EDITUPDT\LAYRCNG INDEX \PAVEDB\EDITUPDT\LAYRCNG

SELECT 2
USE \PAVEDB\EDITUPDT\SUBGCNG INDEX \PAVEDB\EDITUPDT\SUBGCNG

DO WHILE .NOT. EOF()
  MSUBGSID = SID NO
  MSUBGLAY = " 1"
  MSUBGSTR = " 1"

  * find the subgrade sid number in layer identification file
  SELECT 1
  seek str(MSUBGSID,4)+MSUBGLAY+MSUBGSTR
  IF .NOT. FOUND()
    SELECT 2
    DISPLAY FIELD SID NO, STRUCNUM, LAYNUM OFF
    M_ERROR = M_ERROR + 1
  ENDF
  SELECT 2
  SKIP
ENDDO
SELE 1
USE
SELE 2
USE
? " "
? " "
? " "
? " "
? " "
? " "

* TO LIST OUT ALL DATA THAT IS OUT OF RANGE IN THE SUBGRADE FILE
*

* print out page heading
M_TITLE = "SUBGRADE FILE - OUT OF RANGE RECORDS"
M_TITLE2 = "-----------------------------"
? M_TITLE
? M_TITLE2

USE \PAVEDB\EDITUPDT\SUBGCHNG INDEX \PAVEDB\EDITUPDT\SUBGCHNG

* do a range check for all fields
DO WHILE .NOT. EOF()
   IF PLASTIX < 0 .OR. PLASTIX > 70
      IF PLASTIX <> 0
         DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, PLASTIX OFF
         M_ERROR = M_ERROR + 1
         ENDIF
      ENDF
   ENDF
   IF LIQLIM < 10 .OR. LIQLIM > 80
      IF LIQLIM <> 0
         DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LIQLIM OFF
         M_ERROR = M_ERROR + 1
         ENDIF
      ENDF
   ENDF
   IF TXTRIAXL < 1 .OR. TXTRIAXL > 6
      IF TXTRIAXL <> 0
         DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, TXTRIAXL OFF
         M_ERROR = M_ERROR + 1
         ENDIF
      ENDF
   ENDF
   IF PERMIX < 0 .OR. PERMIX > 10
      DISPLAY SID NO, STRUCNUM, LAYNUM, PERMIX OFF
      M_ERROR = M_ERROR + 1
      ENDF
   SKIP
ENDDO
CLEAR
? " "
? " "
? " "
? " "
? " "

415
* LIST OF INCORRECT SID NUMBERS FOR THE SUBGRADE FILE

* M_TITLE = "SUBGRADE FILE - INCORRECT SID NUMBERS"
M_TITLE2 = "-----------------------------"
? M_TITLE
? M_TITLE2

USE \PAVEDB\EDITUPDT\SUBGCHNG INDEX \PAVEDB\EDITUPDT\SUBGCHNG

* initialize variables
STORE 0 TO VAR1, VAR2, VAR3, VAR4, COMPARE

* calculate correct Sid number check digit
DO WHILE .NOT. EOF()
    VAR1 = INT(SID_NO/1000)
    VAR2 = MOD(INT(SID_NO/100),10)
    VAR2 = VAR2 * 2
    VAR3 = MOD(INT(SID_NO/10),10)
    VAR3 = VAR3 * 3
    VAR4 = MOD(VAR1,10)+VAR2+VAR3
    COMPARE = MOD(VAR4,10)

    * compare calculated check digit with actual check digit
    IF COMPARE <> MOD(SID_NO,10)
        DISPLAY SID_NO, STRUCNUM, LAYNUM OFF
        M_ERROR = M_ERROR + 1
    ENDF
    IF SID_NO = 0
        DISPLAY SID_NO, STRUCNUM, LAYNUM OFF
        M_ERROR = M_ERROR + 1
    ENDF
    SKIP
ENDDO
USE
* reset parameters
CLEAR
?

* TO LIST ALL LAYER IDENTIFICATION RECORDS NOT PRESENT IN THE SUBGRADE FILE

* CLEAR
* print out the page heading
M_TITLE = "SUBGRADE FILE - LIST OF ALL LAYER ID RECORDS NOT"
M_TITLE2 = "PRESENT IN THE SUBGRADE FILE"

416
M TITLE3 = "------------------------------------------------------"
? M TITLE
? M TITLE2
? M TITLE3

* assign files to different work areas
SELECT 1
  USE \PAVEDB\EDITUPDT\LAYRCHNG INDEX \PAVEDB\EDITUPDT\LAYRCHNG
SELECT 2
  USE \PAVEDB\EDITUPDT\SUBGCHNG INDEX \PAVEDB\EDITUPDT\SUBGCHNG

SELECT 1
MSID = SID_NO
MSTRUC = "'1"
MLAYNUM = "'1"
DO WHILE .NOT. EOF()
  SELECT 2
    seek str(MSID,4)+MSTRUC+MLAYNUM
    IF .NOT. FOUND()
      SELECT 1
        DISPLAY FIELD SID_NO, STRUCNUM, LAYNUM OFF
        M_ERROR = M_ERROR + 1
      ENDIF
      SELECT 1
      DO WHILE .NOT. EOF() .AND. SID_NO = MSID
        SKIP
      ENDDO
      MSID = SID_NO
    ENDDO
    USE
SELE 1
use
SELE 2
USE
USE \PAVEDB\EDITUPDT\SUBGCHNG INDEX \PAVEDB\EDITUPDT\SUBGCHNG
PACK
USE
CLEAR
? CHR(12)
SET ALTE OFF
CLOSE ALTE
IF M_ERROR <> 0
  M_OK = M_OK + 1
ENDIF
RETURN

417
* * SUBSYSTEM: EDIT & UPDATE INVENTORY FILES
* * PROGRAM NAME: CHEKSURF.PRG 06/23/88
* * CALLED FROM: CHGLAYR.PRG
* * PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* * TAMU/TTI
* * AUTHOR: TREVOR X. PEREIRA
* * PURPOSE: LISTS THE SURFACE FILE RECORDS THAT HAVE ERRORS
*
*
* LISTS THE SURFACE FILE RECORDS THAT ARE NOT PRESENT IN
* THE MASTER LAYER IDENTIFICATION FILE
*
* set parameters
set talk off
set echo off
SET ALTER TO SURFERR
CLEAR
M_ERROR = 0

M_TITLE = "SURFACE FILE - LIST OF RECORDS NOT PRESENT IN LAYER ID FILE"
M_TITLE2 = "--------------------------------------------------------------------------------
SET ALTER ON
? M TITLE
? M_TITLE2

* assign files to different work areas
SELECT 1
 USE \PAVEDB\EDITUPDT\LAYRCNHNG INDEX \PAVEDB\EDITUPDT\LAYRCNHNG

SELECT 2
 USE \PAVEDB\EDITUPDT\SURFCHNG INDEX \PAVEDB\EDITUPDT\SURFCHNG

DO WHILE .NOT. EOF()
* store surface record in memory
MSURFSID = SID_NO
MSURFSIR = STRUCNUM
MSURFAY = LAYNUM

* find the layer ID record that matches surface record
SELECT 1
SEEK SIR(MSURFSID,4)+STR(MSURFSIR,2)+STR(MSURFAY,2)
IF .NOT. FOUND()
    SELE 2
    DISPLAY FIELDS SID_NO, STRUCNUM, LAYNUM OFF
    M_ERROR = M_ERROR + 1
ELSE
    SELE 2
ENDIF
SKIP
ENDDO
SELE 1
USE
SELE 2
USE
? " "
? " "
? " "
? " "
? " "
? " "

* TO LIST ALL OUT OF RANGE DATA IN THE SURFACE FILE
*

* print out page headings
M_TITLE = "SURFACE FILE - OUT OF RANGE RECORDS"
M_TITLE2 = "---------------------"
? M TITLE
? M TITLE2

USE \PAVEDB\EDITUPDT\SURFCHNG INDEX \PAVEDB\EDITUPDT\SURFCHNG

* check data with ranges
DO WHILE NOT. EOF()
  IF STRUCNUM < 1 .OR. STRUCNUM > 9
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, STRUCNUM OFF
    M_ERROR = M_ERROR + 1
  ENDIF
  IF LAYNUM < 1 .OR. LAYNUM > 13
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM, LAYNUM OFF
    M_ERROR = M_ERROR + 1
  ENDIF
  IF AGAPPLRT < 30 .OR. AGAPPLRT > 200
    IF AGAPPLRT <> 0
      DISPLAY SID NO, STRUCNUM, LAYNUM, AGAPPLRT OFF
      M_ERROR = M_ERROR + 1
    ENDIF
  ENDIF
  IF ADMXPER < 0 .OR. ADMXPER > 10.2
    DISPLAY SID NO, STRUCNUM, LAYNUM, ADMXPER OFF
    M_ERROR = M_ERROR + 1
  ENDIF
  IF ASAPPLRT < .1 .OR. ASAPPLRT > .6
    IF ASAPPLRT <> 0
      DISPLAY SID NO, STRUCNUM, LAYNUM, ASAPPLRT OFF
      M_ERROR = M_ERROR + 1
    ENDIF
  ENDIF
  SKIP
ENDDO
? " "

419
TO LIST ALL INCORRECT SID NUMBERS FOR SURFACE FILE

* print out page heading
M TITLXE = "SURFACE FILE - INCORRECT SID NUMBERS"
M TITLXE2 = "--------------------------------------"
? M TITLXE
? M TITLXE2

* initialize variables
STORE 0 TO VAR1, VAR2, VAR3, VAR4, COMPARE

* calculate check digit of Sid number
DO WHILE .NOT. EOF()
   VAR1 = INT(SID_NO/1000)
   VAR2 = MOD(INT(SID_NO/100),10)
   VAR2 = VAR2 * 2
   VAR3 = MOD(INT(SID_NO/10),10)
   VAR3 = VAR3 * 3
   VAR4 = MOD(VAR1,10)+VAR2+VAR3
   COMPARE = MOD(VAR4,10)

   * compare calculated check digit with actual check digit
   IF COMPARE <> MOD(SID_NO,10)
      DISPLAY SID NO, STRUCNUM, LAYNUM OFF
      M_ERROR = M_ERROR + 1
   ENDIF
   IF SID NO = 0
      DISPLAY SID NO, STRUCNUM, LAYNUM OFF
      M_ERROR = M_ERROR + 1
   ENDIF
   SKIP
ENDDO
USE

TO LIST ALL LAYER IDENTIFICATION RECORDS THAT ARE NOT
* PRESENT IN THE SURFACE FILE
*
CLEAR
* print out page heading
M_TITLE  = "SURFACE FILE - LIST OF ALL LAYER ID RECORDS NOT"
M_TITLE2 = "PRESENT IN THE SURFACE FILE"
M_TITLE3 = "---------------------------------------------"

? M_TITLE
? M_TITLE2
? M_TITLE3

* assign files to different work areas
SELECT 1
  USE \PAVEDB\EDITUPDT\LAYRCNG INDEX \PAVEDB\EDITUPDT\LAYRCNG
SELECT 2
  USE \PAVEDB\EDITUPDT\SURFCNG INDEX \PAVEDB\EDITUPDT\SURFCNG

SELECT 1
DO WHILE .NOT. EOF()
  STORE .T. TO GOTT

  * find the surface layers in Layer ID file
  DO WHILE GOTT
    MSID = SID NO
    MSIR = STRUCNUM
    MLAY = LAYNUM
    IF LAYMATCL < 17
      GOTT = .F.
      MDESC = LAYMATCL
    ELSE
      SKIP
    ENDIF
  ENDDO

  * find the Layer ID record in Surface file
  SELECT 2
  SEEK STR(MSID,4)+STR(MSIR,2)+STR(MLAY,2)
  IF .NOT. FOUND()
    SELECT 1
    DISPLAY FIELDS SID NO, STRUCNUM, LAYNUM OFF
    M_ERROR = M_ERROR + 1
  ENDIF
  SELECT 1
  SKIP
ENDDO
SELECT 1
use
SELECT 2
USE \PAVEDB\EDITUPDT\SURFCNG INDEX \PAVEDB\EDITUPDT\SURFCNG
PACK
USE
CLEAR
? CHR(12)
SET ALTE OFF
CLOSE ALTE
IF M_ERROR <> 0
  M_OK = M_OK + 1
Section 4: Traffic Data Update
Traffic Tape to Disk

PROGRAM NARRATIVE

This program is for use with a 6250 BPI Cipher GRC CacheTape tape drive system manufactured by Overland Data Incorporated (ODI). It utilizes the ODI DEPOT2 program to copy a tape file to a disk file. The DEPOT2 program is documented in Chapter 4 of the ODI manual. STLOG.PAS (a subprogram of TRAFUPD.PRG) uses the disk file as an input file. The diskfile is stored in the subdirectory \PAVEDB\FILES.
PROGRAM SPECIFICATION

PROGRAM NAME: READIRAF.BAT

PURPOSE: To copy the Traffic data from a tape to a disk file using the ODI DEPOT2 program.

INPUT FILE: The annual Roadway Inventory Tape File (RIPTLE) obtained from the Texas State Department of Highways and Public Transportation.

OUTPUT FILE: TLOG.DAT - Annual Roadway Inventory Disk File.

PROGRAMS CALLED:

DEPOT2
READIRAF.CMD - command file for DEPOT2
rem readtraf.bat issues the depot2 command which calls
rem the cmd file to read the annual Roadway Inventory tape obtained
rem from SDHPT
DEPOT2 /c READTRAFCMD
PROGRAM LISTING

; README.F, a command file for the ODI program DEPOT2.
; Use as: DEPOT2 /c README.F
; Instructions on making out this command file are in Chapter 4
; of the ODI manual pages 4-26 to 4-28.
; Change D:\PAVEDB\FILES\TLOG.DAT to the name of your output disk file,
; if needed.
; Change "/r nn" record length if needed.
; Change "/s mnnn" blocksize if needed.
; "/m l" skips past the tape label.
; "/tvM" translates from EBCDIC, Verbose explanations, reads till End-
; Of-File mark at the end of the data.

/n D:\PAVEDB\FILES\TLOG.DAT /r 263 /s 263 /m 1 /tvM
Traffic Update

PROGRAM NARRATIVE

The traffic update program allows the user to update the Traffic File in the Texas Flexible Pavement Database. The source of new data for this process is the annual Roadway Inventory Data File (RIFILE) which is produced by D-10 of the Texas State Department of Highways and Public Transportation. The RIFILE data is copied from a tape to a disk file by a stand alone batch program (READTRAF.BAT). The program will extract relevant traffic information from the RIFILE file and through some intermediate data processing update the Traffic Database. Due to the normally large amount of data, this update procedure is time-consuming. It would thus make sense to perform it only once a year. The traffic update programs are stored in the subdirectory \PAVEDB\EDITUPDT\TRAFFIC. Other than the master traffic file, the files used by traffic update are stored in \PAVEDB\EDITUPDT\TRAFFIC.
PROGRAM SPECIFICATION

PROGRAM NAME: TRAFUPD.PRG

PURPOSE: To use annual Roadway Information File (RIFILE) data to update the Texas Flexible Pavement Database (TRAFFIC.DBF) Traffic File.

PROCEDURES/EDITS:

1. The traffic data is received on a tape. It is copied from the tape to a disk file and converted from EBCDIC to ASCII format. The data is stored as a record per line in the new disk file. This process is performed by READTRAFF.BAT and is not part of the menu system.

2. The Traffic File contains yearly information for each section identification number. The fields include annual average daily traffic (one-way), annual 18KIP, and percentage of trucks.

INPUT FILES:

1. Annual Roadway Inventory Disk File (TLOGxx.DAT where xx stands for the last 2 digits of the RIFILE year)

I/O FILES:

1. Traffic database file (TRAFFIC.DBF)

PROGRAMS CALLED:

STLOG.PAS
STDLOG.PRG
LOGITRAF.PRG
NEWITRAF.PRG

NOTE:

1) Where TLOG is used in the following documentation it is referring to the RIFILE.

2) Program Flow Diagrams, Program Specifications, related file layouts, and the program listings are provided in the remainder of this section for each of the called programs.
<table>
<thead>
<tr>
<th>LABEL</th>
<th>TYPE</th>
<th>LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUB FILE NUMBER</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>FILE NUMBER</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>THD DISTRICT</td>
<td>NUM</td>
<td>2</td>
</tr>
<tr>
<td>THD COUNTY</td>
<td>NUM</td>
<td>3</td>
</tr>
<tr>
<td>SHD CONTROL NUMBER</td>
<td>CHAR</td>
<td>4</td>
</tr>
<tr>
<td>SHD SECTION NUMBER</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>SHD CONTROL/SECTION NUMBER</td>
<td>CHAR</td>
<td>6</td>
</tr>
<tr>
<td>BEGINNING MILEPOINT</td>
<td>NUM</td>
<td>5.3</td>
</tr>
<tr>
<td>ENDING MILEPOINT</td>
<td>NUM</td>
<td>5.3</td>
</tr>
<tr>
<td>LENGTH OF SECTION</td>
<td>NUM</td>
<td>5.3</td>
</tr>
<tr>
<td>DESIGNATED HIGHWAY SYSTEM</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>DESIGNATED HIGHWAY NUMBER</td>
<td>CHAR</td>
<td>4</td>
</tr>
<tr>
<td>HIGHWAY STATUS</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>RIGHT-OF-WAY WIDTH (IN FEET)</td>
<td>CHAR</td>
<td>3</td>
</tr>
<tr>
<td>HIGHWAY DESIGN TYPE</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>ROAD BED WIDTH (IN FEET)</td>
<td>CHAR</td>
<td>3</td>
</tr>
<tr>
<td>BASE TYPE</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>SURFACE WIDTH (IN FEET)</td>
<td>CHAR</td>
<td>3</td>
</tr>
<tr>
<td>COMBINATION OF TWO OR MORE SURFACE TYPES</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>SURFACE TYPE</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>NUMBER OF LANES</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>SHOULDERS TYPE</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>DATE OF LAST MILEPOST CHANGE (YMM)</td>
<td>CHAR</td>
<td>4</td>
</tr>
<tr>
<td>YEAR OF LAST MILEPOST CHANGE</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>MONTH OF LAST MILEPOST CHANGE</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>ADMINISTRATIVE SYSTEM</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>TYPE RESERVATION SECTION LOCATED IN</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>DESIGNATED WAY FEDERAL AID CODE</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>1982 FUNCTIONAL CLASSIFICATION</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>URBAN-RURAL FUNCTIONAL CLASSIFICATION</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>CONNECTION LINK CODE (FUNCTIONAL CLASS)</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>MAINTENANCE SECTION NUMBER</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>MAINTENANCE CLASS CODE</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>RESTRICTED LOAD LIMIT</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>SPECIAL SYSTEMS</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>TRUCKS OR COMMERCIAL VEHICLES</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>CITY NUMBER</td>
<td>NUM</td>
<td>5</td>
</tr>
<tr>
<td>URBAN AREA NUMBER</td>
<td>NUM</td>
<td>5</td>
</tr>
<tr>
<td>HWY PERFORMANCE MONITORING SYST SECTION</td>
<td>CHAR</td>
<td>12</td>
</tr>
<tr>
<td>YEAR OF CURRENT AADT</td>
<td>NUM</td>
<td>2</td>
</tr>
<tr>
<td>ANNUAL AVG DAILY TRAFFIC, CURRENT YEAR</td>
<td>NUM</td>
<td>6</td>
</tr>
<tr>
<td>AADT FOR 1 YEAR PRIOR TO CURRENT YEAR</td>
<td>NUM</td>
<td>6</td>
</tr>
<tr>
<td>AADT FOR 2 YEARS PRIOR TO CURRENT YEAR</td>
<td>NUM</td>
<td>6</td>
</tr>
<tr>
<td>AADT FOR 3 YEARS PRIOR TO CURRENT YEAR</td>
<td>NUM</td>
<td>6</td>
</tr>
<tr>
<td>AADT FOR 4 YEARS PRIOR TO CURRENT YEAR</td>
<td>NUM</td>
<td>6</td>
</tr>
<tr>
<td>LABEL</td>
<td>TYPE</td>
<td>LENGTH</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>AADT FOR 5 YEARS PRIOR TO CURRENT YEAR</td>
<td>NUM</td>
<td>6</td>
</tr>
<tr>
<td>AADT FOR 6 YEARS PRIOR TO CURRENT YEAR</td>
<td>NUM</td>
<td>6</td>
</tr>
<tr>
<td>AADT FOR 7 YEARS PRIOR TO CURRENT YEAR</td>
<td>NUM</td>
<td>6</td>
</tr>
<tr>
<td>AADT FOR 8 YEARS PRIOR TO CURRENT YEAR</td>
<td>NUM</td>
<td>6</td>
</tr>
<tr>
<td>AADT FOR 9 YEARS PRIOR TO CURRENT YEAR</td>
<td>NUM</td>
<td>6</td>
</tr>
<tr>
<td>DAILY VEHICLE MILES FOR CURRENT YEAR</td>
<td>NUM</td>
<td>8</td>
</tr>
<tr>
<td>DESIGN YEAR</td>
<td>NUM</td>
<td>2</td>
</tr>
<tr>
<td>YEARLY ADT INCREASE FACTOR</td>
<td>NUM</td>
<td>4.2</td>
</tr>
<tr>
<td>ESTIMATED AADT FOR DESIGN YEAR</td>
<td>NUM</td>
<td>6</td>
</tr>
<tr>
<td>ESTIMATED DESIGN YEAR DAILY VEH. MILES</td>
<td>NUM</td>
<td>8</td>
</tr>
<tr>
<td>DESIGN HOUR'S % OF THE 24 HR AADT</td>
<td>NUM</td>
<td>3.1</td>
</tr>
<tr>
<td>DIRECTIONAL DISTRIBUTION IN DHV</td>
<td>NUM</td>
<td>2</td>
</tr>
<tr>
<td>% TRUCKS IN AADT</td>
<td>NUM</td>
<td>3.1</td>
</tr>
<tr>
<td>% TRUCKS IN DHV</td>
<td>NUM</td>
<td>3.1</td>
</tr>
<tr>
<td>AVG 10 HEAVIEST WHEEL LOADS (IN 100 LBS.)</td>
<td>NUM</td>
<td>3</td>
</tr>
<tr>
<td>% TANDEM AXLES IN AHWLD w FIELD</td>
<td>NUM</td>
<td>2</td>
</tr>
<tr>
<td># OF WHEEL LOADS 1/2 8000 LBS</td>
<td>NUM</td>
<td>5</td>
</tr>
<tr>
<td>18K FLEX. PAVEMENT ANALYSIS (IN 1000'S)</td>
<td>NUM</td>
<td>5</td>
</tr>
<tr>
<td>18K RIGID PAVEMENT ANALYSIS (IN 1000'S)</td>
<td>NUM</td>
<td>5</td>
</tr>
<tr>
<td>DESIGN HOUR VOLUME</td>
<td>NUM</td>
<td>5</td>
</tr>
<tr>
<td>CONTROL-SECTION LENGTH (NEAREST .001)</td>
<td>NUM</td>
<td>5.3</td>
</tr>
<tr>
<td>PRIORITY HIGHWAY SYSTEM</td>
<td>CHAR</td>
<td>2</td>
</tr>
<tr>
<td>PRIORITY HIGHWAY NUMBER</td>
<td>NUM</td>
<td>4</td>
</tr>
<tr>
<td>CONTROL-SECTION LENGTH (NEAREST .1)</td>
<td>NUM</td>
<td>3.1</td>
</tr>
<tr>
<td>CONTROL-SECTION VEHICLE MILES</td>
<td>NUM</td>
<td>16</td>
</tr>
</tbody>
</table>
* File name: TRAFUPD.FRG
* Written by: Victor Wong
* Created on: April 8, 1988
* Last updated: April 8, 1988
* Purpose: To use new RI2-TLOG data to update Texas Flexible Pavement Traffic Database.

* warn user of this time-consuming process
CLEAR
RES = " "
@ 10,16 SAY "This process will update the TRAFFIC database by"
@ 11,16 SAY "retrieving new data from RI2-TLOG file. Because"
@ 12,16 SAY "of the size of the data file, this process will"
@ 13,16 SAY "take a very long time (at least over night)!! "
@ 15,16 SAY "Do you still want to proceed? (Y/N) " GET RES
READ

IF RES $ "y"
* extract traffic information from RI2-TLOG data file
RUN \PAVEDB\EDITUPDT\TRAFFIC\STLOG

* order simplified RI2-TLOG data by section identification number
DO \PAVEDB\EDITUPDT\TRAFFIC\SIDTLOG

* calculate the Traffic information from the TLOG data
DO \PAVEDB\EDITUPDT\TRAFFIC\LOGIRAF

* update the old Traffic database by adding TLOG Traffic data
DO \PAVEDB\EDITUPDT\TRAFFIC\NEWIRAF
ENDIF

RETURN
Program Name: STLOG.PAS

Purpose: To simplify the TLOG data by extracting only fields that are required to calculate the values for the Traffic Database File.

Procedures/Edit:

1. The subprogram produces a simplified TLOG data file (STLOG.DLM) from the complete data set (TLOGxx.DAT). The extracted data is stored in a blank-delimited format.

Input Files(s):

1. TLOG data file (TLOG.DAT).

Output Files(s):

1. Simplified TLOG data file (STLOG.DLM).
## Traffic Update

**Subprogram 1: STLOG**

**STLOG Delimited File Record Layout**

**File Name:** STLOG.DIM

<table>
<thead>
<tr>
<th>Description</th>
<th>Size/Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Section number</td>
<td>4N</td>
</tr>
<tr>
<td>Beginning Mile Point</td>
<td>6.3N</td>
</tr>
<tr>
<td>Ending Mile Point</td>
<td>6.3N</td>
</tr>
<tr>
<td>Year of Current AADT</td>
<td>2N</td>
</tr>
<tr>
<td>Annual Average Daily Traffic, Current Year (AADT)</td>
<td>6N</td>
</tr>
<tr>
<td>AADT Increase Factor (x 0.01)</td>
<td>4N</td>
</tr>
<tr>
<td>Estimated AADT for Design Year</td>
<td>6N</td>
</tr>
<tr>
<td>Percent Trucks in AADT (x 0.1)</td>
<td>3N</td>
</tr>
<tr>
<td>Estimated 18K Flexible Pavement (x 1000)</td>
<td>5N</td>
</tr>
</tbody>
</table>
(* File name: STLOG.PAS
* Program name: stlog
* Project 2456: Texas Flexible Pavement Database Conversion
* TAMU/TTI
* Written by: Victor Wong
* Purpose: This program simplifies the data from TLOG file. The only data extracted is Control
*   Section number, Beginning Mile Point (EMP),
*   Ending Mile Point (EMP), Current year, Annual
*   Average Daily Traffic data (AADT), Design
*   year, Average Daily Traffic Increase Factor
*   (ADT_INCR_FACTOR), Estimated Average Daily
*   Traffic (EST ADT), Percentage of traffic is
*   Trucks (PERCENT_TRUCK), and Estimated 18KIP
*   (EST_18K).
* Input File: TLOG text data file
* Output File: STLOG.DEL
* NOTE: This procedure only works for TLOG data file AFTER 1985.
*)

{ PROGRAM STLOG <<<}
program stlog;

{ variables declarations <<<}
var
  in_file, out_file: text;  { input & output files }  
  skip_junk_1        : string[7];  { control section number }
  cntl_section_no    : string[6];  { beginning & ending }
  bmp_1, emp_1       : string[2];  { mile points }
  bmp_2, emp_2       : string[3];  { current yr of adt data }
  skip_junk_2        : string[96];  { adt data now & 9 yrs back }
  current_yr         : string[2];  { design year }
  adt_incr_factor    : array [1..10] of string[6];  { act increment factor }
  adt                 : string[6];  { estimated adt }
  skip_junk_4        : string[8];  
  design_yr          : string[2];  { percent truck }
  est_18K             : string[3];  { percent truck }
  skip_junk_5        : string[13];  
  percent_truck      : string[3];  { estimated 18k }
  est_18K             : string[13];  
  count               : integer;
  ok                  : boolean;
  krec, rec           : integer;

{ beginning of STLOG.PAS <<<}
begin
  Clrscr;
  gotoxy (20,10);
  write ('<< STLOG running ... >>');
  gotoxy (20,11);
  write (' Simplifying the TLOG data.');

  assign (in_file, '"pavedb\tlog.dat"');
  reset (in_file);

  assign (out_file, '"pavedb\editupdt\traffic\stlog.dlm"');
  rewrite (out_file);

  krec:= 0;
  rec:= 0;

  gotoxy (20,15);
  write ('Extracting data from TLOG data file...');
  gotoxy (20,16);
  write (krec:3,'',',rec:3,' records extracted');

  (--- do record by record until end of file for in_file---)
  while not eof (in_file) do
    begin
      (read in control section number)
      read (in_file, skip_junk_1, cntl_section_no);

      (read the rest of the info )
      if cntl_section_no is present
        if (cntl_section_no <> '000000')
          and (cntl_section_no <> ' 0')
          and (cntl_section_no <> ' ') then
            begin
              read (in_file, bmp_1, bmp_2);
              read (in_file, emp_1, emp_2);
              read (in_file, skip_junk_2, current_yr);

              (if current_yr is non-zero then continue)
              if current_yr <> '00' then
                begin
                  for count:= 1 to 10 do
                    read (in_file, aadt[count]);
                  read (in_file, skip_junk_3, design_yr);
                  read (in_file, adt_incr_factor, est_adt);
                  read (in_file, skip_junk_4, percent_truck);
                  readln (in_file, skip_junk_5, est_18K);

                  (write out info)
                  write (out_file, cntl_section_no, '"bmp_1,');
                  write (out_file, bmp_2, '"emp_1,');
                  if (design_yr <= '99') and (design_yr > '80') then
                    write (out_file, '19',design_yr)
                  else
          
      439
write (out_file, ' 20', design_yr);
if (current_yr <= '99') and (current_yr > '80') then
  write (out_file, ' 19', current_yr)
else
  write (out_file, ' 20', current_yr);
write (out_file, '', aadt[1]);
write (out_file, '', adt_incr_factor, '', est_adt);
writeln (out_file, '', percent_truck, '', est_18K);

if (rec < 999) then
  begin
    rec := rec + 1;
    gotoxy (24,16);
    write (rec:3);
  end
else
  begin
    krec := krec + 1;
    gotoxy (20,16);
    write (krec:3);
    rec := 0;
    gotoxy (24,16);
    write (rec:3);
  end;
end; (if)
end; (if)
end; (while)

{close input and output files}
close (in_file);
close (out_file);

gotoxy (20,18);
write ('<<<< STLOG done. >>');
end. (simplify_tlog_data)
SUBPROGRAM SIDTLOG.PRG – PROGRAM FLOW DIAGRAM

STLOG.DLM

simplified TLOG data

STLOG.DBF

SIDTLOG.PRG

LOCATION.DBF

SIDTLOG.DBF

FIGURE 21
Traffic Update
Subprogram 2: SIDTLOG

PROGRAM SPECIFICATIONS

Program Name: SIDTLOG.PRG

Purpose: To use the simplified TLOG data to create a dBASE file containing TLOG data for the section identification numbers found in the Location file.

Procedures/Edits:

1. The TLOG data key is the control section number.

2. The Location database contains information about the control section number and the beginning and ending milepoint of section identification numbers.

3. The program creates a temporary SIDTLOG dBASE file. This file contains the TLOG data for section identification numbers. This is done by joining the simplified TLOG and Location data by the control section number and mile-point range.

Input File(s):
1. Simplified TLOG data file (STLOG.DLM).
2. Location database file (LOCATION.DBF).

I/O File(s):
1. dBASE file for storing the simplified TLOG data (STLOG.DBF).

Output File(s):
1. dBASE file with TLOG data under section identification number (SIDTLOG.DBF)
### Traffic Update
#### Subprogram 2: SIDTLOG

**STLOG dBASE File Record Layout**

**File Name:** STLOG.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTLSEC</td>
<td>*</td>
<td>6N</td>
<td>Control Section Number</td>
</tr>
<tr>
<td>TLOGEMP</td>
<td>*</td>
<td>6.3N</td>
<td>Beginning Mile Point</td>
</tr>
<tr>
<td>TLOGEMP</td>
<td>*</td>
<td>6.3N</td>
<td>Ending Mile Point</td>
</tr>
<tr>
<td>CURYR</td>
<td></td>
<td>4N</td>
<td>Year of Current AADT</td>
</tr>
<tr>
<td>ADT</td>
<td></td>
<td>6N</td>
<td>Average Daily Traffic</td>
</tr>
<tr>
<td>ADTINCR</td>
<td></td>
<td>4N</td>
<td>ADT Increase Factor</td>
</tr>
<tr>
<td>ESTADT</td>
<td></td>
<td>6N</td>
<td>Estimated ADT for Design Year</td>
</tr>
<tr>
<td>PCNTIRK</td>
<td></td>
<td>3N</td>
<td>Percent Trucks in ADT</td>
</tr>
<tr>
<td>EST18K</td>
<td></td>
<td>5N</td>
<td>Estimated 18K Flexible Pavement</td>
</tr>
</tbody>
</table>
Traffic Update
Subprogram 2: SIDTLOG

SIDTLOG dBASE File Record Layout

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>BEGMPT</td>
<td></td>
<td>6.3N</td>
<td>Beginning Mile Point</td>
</tr>
<tr>
<td>ENDMPT</td>
<td></td>
<td>6.3N</td>
<td>Ending Mile Point</td>
</tr>
<tr>
<td>TLOGBMP</td>
<td></td>
<td>6.3N</td>
<td>TLOG Beginning Mile Point</td>
</tr>
<tr>
<td>TLOGEMP</td>
<td></td>
<td>6.3N</td>
<td>TLOG Ending Mile Point</td>
</tr>
<tr>
<td>CURYR</td>
<td></td>
<td>4N</td>
<td>Year of Current ADT</td>
</tr>
<tr>
<td>ADT</td>
<td></td>
<td>6N</td>
<td>Average Daily Traffic</td>
</tr>
<tr>
<td>ADTINCR</td>
<td></td>
<td>4N</td>
<td>ADT Increase Factor</td>
</tr>
<tr>
<td>ESTADT</td>
<td></td>
<td>6N</td>
<td>Estimated ADT for Design Year</td>
</tr>
<tr>
<td>PCNTRK</td>
<td></td>
<td>3N</td>
<td>Percent Trucks in ADT</td>
</tr>
<tr>
<td>EST18K</td>
<td></td>
<td>5N</td>
<td>Estimated 18K Flexible Pavement</td>
</tr>
</tbody>
</table>
PROGRAM LISTING

* File name: SIDTLOG.PRG
* Program: sidtlog
* Project 2456: Texas Flexible Pavement Database Conversion
* TAMU/TII
* Written by: Victor Wong
* Created on: April 13, 1988
* Last updated: August 8, 1988
* Purpose: DBASE III+ program to create the sidtlog.dbf file. The program will join the location.dbf and stlog.dbf files by the control section number. The goal is to have tlog information arranged in sid number instead of control section number.
* Input file(s): STLOG.DIM delimited TLOG data file
* LOCATION.DBF database with location data
* Output file(s): STLOG.NDX index file, deleted when done
* I/O file(s): STLOG.DBF database to store the STLOG data
* SIDTLOG.DBF database with TLOG data by SID #

* inform the use that the program is running
? "<< SIDTLOG.PRG running ... >>"
? " Arranging TLOG data by Section Identification Number."

* set flags
SET SAFETY OFF

* prepare the stlog database
USE \PAVEDB\EDITUPDT\TRAFFIC\STLOG
ZAP
APPEND FROM \PAVEDB\EDITUPDT\TRAFFIC\STLOG.DIM DELIMITED WITH BLANK INDEX ON CNTRLSEC TO \PAVEDB\EDITUPDT\TRAFFIC\STLOG
CLOSE ALL

* select files to be used
SELECT A
USE \PAVEDB\FILES\LOCATION
SELECT B
USE \PAVEDB\EDITUPDT\TRAFFIC\STLOG INDEX \PAVEDB\EDITUPDT\TRAFFIC\STLOG
SELECT C
USE \PAVEDB\EDITUPDT\TRAFFIC\SIDTLOG
ZAP

* for every location record find the tlog info
SELECT A
DO WHILE .NOT. EOF()
    IF A->ACTIVFLAG

      * find the same control section in stlog file
      SELECT B
IF (A->SECTION/10) >= 1
    ACNILSEC = INT(VAL( STR(A->CONTROL,4)+STR(A->SECTION,2)))
ELSE
    ACNILSEC = INT(VAL( STR(A->CONTROL,4) + "0"; + LTRIM(STR(A->SECTION,2)) ))
ENDIF
SEEK ACNILSEC

IF FOUND()
    * if found then for all the stlog record of that control *
    * section
    DO WHILE CNTLSEC=ACNILSEC
        * check to see if the stlog record's mile point range *
        * is within the range of location's
        IF ((TLOGEMP>=A->BEGMPNT).AND.((TLOGEMP<=A->ENDMPNT))
            .OR.((TLOGEMP>=A->BEGMPNT).AND.((TLOGEMP<=A->ENDMPNT))
            .OR.((TLOGEMP=A->BEGMPNT).AND.((TLOGEMP<=A->ENDMPNT))
            .OR.((TLOGEMP=A->BEGMPNT).AND.((TLOGEMP>=A->ENDMPNT))
        * if so, append a new sidtlog record with the info
        SELECT C
        APPEND BLANK
        REPLACE SID_NO WITH A-> SID_NO
        REPLACE BEGMPNT WITH A-> BEGMPNT
        REPLACE ENDMNT WITH A-> ENDMNT
        REPLACE TLOGEMP WITH B-> TLOGEMP
        REPLACE TLOGEMP WITH B-> TLOGEMP
        REPLACE DESYR WITH B-> DESYR
        REPLACE CURYR WITH B-> CURYR
        REPLACE ADT WITH B-> ADT
        REPLACE ADTINCR WITH B-> ADTINCR
        REPLACE ESTADT WITH B-> ESTADT
        REPLACE PCNTRK WITH B-> PCNTRK
        REPLACE EST18K WITH B-> EST18K
        ENDF
    * skip to next stlog record
        SELECT B
        SKIP
    ENDDO
ENDIF
ENDIF

* skip to next location record
SELECT A
SKIP
ENDDO

* close all the files
CLOSE ALL
* erase intermediate file
ERASE \PAVEDB\EDITUPDT\TRAFFIC\STLOG.NDX
* reset flags
SET SAFETY ON
* inform the user that the program is done
? "<< SIDTLOG.FRG done. >>"
Diagram 4. Program flow for subprogram LOGTRAF.
Traffic Update
Subprogram 3: LOGITRAF

PROGRAM SPECIFICATION

Program Name: LOGITRAF.PRG

Purpose: To calculate the annual average daily traffic, the percent of trucks, and the cumulative 18 keal for the Traffic dBASE file by processing the TLOG data of each section identification number.

Procedures/Edits:

1. The TLOG data for section identification numbers is found in SIDTLOG.DBF.

2. The program calculates traffic values needed to update the Traffic database. These values of a new year are stored in dBASE LOGITRAF.DBF.

Input File(s): 1. dBASE file containing TLOG data for each section identification number (SIDTLOG.DBF).

Output File(s):
1. dBASE file with TLOG data under section identification number (SIDTLOG.DBF)
**Traffic Update**  
Subprogram 3: LOGTRAF

LOGTRAF dBASE File Record Layout

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>YEAR</td>
<td>*</td>
<td>4N</td>
<td>Year</td>
</tr>
<tr>
<td>AADT1WAY</td>
<td></td>
<td>6N</td>
<td>Average Daily Traffic (one-way)</td>
</tr>
<tr>
<td>A18KEAL</td>
<td></td>
<td>9N</td>
<td>Annual Cumulative 18KEAL one</td>
</tr>
<tr>
<td>way</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCTTRK</td>
<td></td>
<td>4N</td>
<td>Percent Trucks in ADT</td>
</tr>
</tbody>
</table>

**NOTE:** 18 KEAL is 18000 lbs. equivalent axle load. This figure is calculated using the AASHTO Equivalency factors which convert any weighted truck axle to a number of equivalent of 18000 lbs. single axles. The basics for the equivalency is the observed pavement damage done by different axle loads at the AASHTO road Test (1959-1960).
PROGRAM LISTING

* File name: LOGTraf.PRG
* Created by: Victor Wong
* Written on: Apr 13, 1988
* Last updated: Apr 13, 1988
* Purpose: To calculate the adt, 18KIP and percent truck values
  * for test sections from sidtlog.dbf. The result will
  * be stored in logtraf.dbf
* I/O file(s): LOGTraf.DBF tlog traffic data base
* Input file(s): SIDTLOG.DBF tlog data base keyed on sid_no

* inform the user that the program is running
? "<< LOGTraf.PRG running ... >>"
? " Calculating the traffic data from the TLOG data."

* set flags
SET SAFETY OFF

* open files
SELECT A
USE \PAVEDB\EDITUPDT\TRAFFIC\LOGTraf
ZAP
SELECT B
USE \PAVEDB\EDITUPDT\TRAFFIC\SIDTLOG

* while not eof for sidtlog
SELECT B
DO WHILE .NOT. EOF()
  * variables initializations
  PREV_SID = SID_NO
  ACUM_PROP = 0.0
  ACUM_PCTNTRK = 0.0
  ACUM_TRAF = 0.0
  ACUM_18K = 0.0

  * while it is still the same sid number
  DO WHILE SID_NO = PREV_SID
    * prior adjustment to some field values
      ADT_INCR = ADTINCR/100.0
      PCNT_TRK = PCNTRK/10.0 * 0.01
      EST_18K = EST18K * 1000.0
      CUR_YR = CURyr

      IF EST_18K <> 0.0
        * calculate the 18KIP and adt values for that control section
        TRK_18K = ((ESTADT + ADT)/4) * 365.25 * (DESyr - CURyr) *;
        PCNT_TRK
        IF TRK_18K <> 0.0
          TRK_18K = EST_18K/TRK_18K
    ENDF

ENDWHILE

450
TRAF_ADT = (ADT/2) * 12 * 30.4375

* calculate the proportion of the control section to the test section
SEC_PROP =
(MIN(TLOGEMP,ENDMPTN) - MAX(TLOGEMP,BEGMPTN)) /(ENDMPTN-BEGMPTN)
ACUM_PROP = ACUM_PROP + SEC_PROP

* calculate the accumulated traffic, 18kip and percent truck values
* for the control sections
ACUM_TRAF = ACUM_TRAF + (SEC_PROP * TRAF_AD
ACUM_18K = ACUM_18K +(SEC_PROP *(TRAF_AD *PCNT_TRK)*TRK_18K)
ACUM_PCTRKO = ACUM_PCTRKO + (PCNT_TRK * 100 * SEC_PROP)

ENDIF

* skip to the next control section
SKIP
ENDDO

* if the acum_prop is not zero
IF ACUM_PROP <> 0.0

TLOG_AD = 0.5 + ACUM_TRAF/(ACUM_PROP * 365.25)
TLOG_18K = 0.5 + ACUM_18K/ACUM_PROP
TLOG_PCTRKO = ACUM_PCTRKO/ACUM_PROP
ELSE

TLOG_AD = 0.0
TLOG_18K = 0.0
TLOG_PCTRKO = 0.0
ENDIF

* find the location of the sid no in logtraf.dbf and store the result
SELECT A
APPEND BLANK
REPLACE SID NO WITH PREV_SID
REPLACE YEAR WITH CUR_YR
REPLACE AADTLWAY WITH TLOG_AD
REPLACE A18KFAAL WITH TLOG_18K
REPLACE PCNTTRKO WITH TLOG_PCTRKO

* get back with sidtlog.dbf
SELECT B
ENDDO

* close all files
CLOSE ALL

* reset flags
SET SAFETY ON

* inform the user that the program is done
? "<< LOGTRAFF.PRGR done. >>"
SUBPROGRAM NEWTRAF.PRG - PROGRAM FLOW DIAGRAM

FIGURE 23

LOGTRAF.DBF

NEWTRAF.PRG

TRAFFIC.DBF
Traffic Update
Subprogram 4: NEWTRAF

PROGRAM SPECIFICATIONS

Program Name: NEWTRAF.FRG

Purpose: To use the new traffic data in the LOGTRAF dBASE file to update the Traffic database file.

Procedures/Edits:

1. The updating is done by appending records from the LOGTRAF dBASE file to TRAFFIC.DBF.

2. The Traffic database file is sorted by section identification number and the year after the appending of new data.

Input File(s):

1. Database file containing traffic values for a new year (LOGTRAF.DBF).

Output File(s):

1. Traffic database file (TRAFFIC.DBF).
* File Name: NEWTRAF.PRG
* Written by: Victor Wong
* Created on: April 13, 1988
* Last updated: April 13, 1988
* Purpose: A DBASE III + program that updates the traffic.dbf
         (after 1985) file with the logtraf.dbf file.
* Input file(s): LOGTRAФ.DBF
* Output file(s): TRAFFIC.BK (backup for TRAFFIC.DBF)
* I/O file(s): TRAFFIC.DBF

* inform the user that newtraf is running
? "<< NEWTRAF.PRG running ... >>"
? " Updating the Traffic Database with data calculated from TLOG."

* set flags
SET SAFETY OFF

* make a backup of the old traffic database
COPY FILE \PAVEDB\FILES\TRAFFIC.DBF TO \PAVEDB\FILES\TRAFFIC.BK

* append the new traffic data
USE \PAVEDB\FILES\TRAFFIC
APPEND FROM \PAVEDB\EDITUPDT\TRAFFIC\LOGTRAФ

* sort the traffic database according to sid_no and year
SORT TO \PAVEDB\FILES\TEMPTRAФ ON SID_NO, YEAR
CLOSE ALL

* rename the new traffic database to traffic.dbf
ERASE \PAVEDB\FILES\TRAFFIC.DBF
RENAME \PAVEDB\FILES\TEMPTRAФ.DBF TO \PAVEDB\FILES\TRAFFIC.DBF

* erase intermediate files
ERASE \PAVEDB\FILES\TEMPTRAФ.DBF

* set flags
SET SAFETY ON

* inform the user that the program is done
? "<< NEWTRAF.PRG done ... >>"
CHAPTER 5

APPLICATIONS SUBSYSTEM
GENERAL NARRATIVE

The Applications Subsystem draws on the monitor an exponential decay curve for performance versus accumulated 18 KEAL where performance is indicated by PSI or area of distress for alligator cracking or rutting; creates a 'distress' file which contains the data needed to construct the curve; and builds a file which can be used to develop performance models.

The Distress File is a subset of data from the master files. It was created to substantially decrease the processing time for constructing the curves and for building the Model Files. The Distress File is also used to calculate values for the Model File. In addition to the values created from the Distress File, the Model File also includes a subset of data from the master files. It too was created to substantially reduce the processing time involved in developing performance models. The Model File can be directly processed by SAS. For additional information refer to Volume 1 of this report.

This chapter is divided into the following four sections:

- Application Driver
- Graph Accumulated 18 KIP vs Distress or PSI
- Building Model File
- Create Distress File

Each section contains the following information:

- Program Narrative
- Program Flow Diagram
- Program Specification
- Menu Screens
- Program Listings
Section 1: Application Driver
Applications Subsystem Driver Program Flow

FIGURE 24
The Application Subsystem main driver program (APPLICAT.PRG) initiates the programs which create the Distress & Model Files and the program which draws the 18 KIP vs Distress or PSI graphs on the screen.

In addition to driving the applications subsystem, APPLICAT.PRG also draws the menu for the application subsystem. When the user chooses an option on the applications menu, the appropriate program is called. If Option 1 (Graph Accumulated 18 KIP vs Distress or PSI) is chosen, program GRAPH18K is called. Option 2 (Build Model File) calls program MODEL.PRG while Option 3 (Build Distress File) calls program DISTRESS.PRG and DISTIVISL.PRG. The program APPLICAT.PRG is in the subdirectory \FAVEDB\APPLICAT.
PROGRAM SPECIFICATION

Program Name: APPLICATION.PRG

Purpose: To display the Application Menu Screen and to call the appropriate programs.

Procedures/Edits:

The following are the procedures:
1) Graph the 18 KIP vs Distress or PSI (GRAPH18K.PRG).
2) Build the Model File (MODEL.PRG).
3) Create the Distress file (DISTRESS.PRG & DISTVISL.PRG).

Input/Output Files:

None

Programs Called (See Program Flow Diagram):

- GRAPH18K.PRG
- MODEL.PRG
- DISTRESS.PRG
- DISTVISL.PRG
Above is Main Menu of the Texas Flexible Pavement System.

When Option 4 - Application is chosen, the application driver program APPLICAT.PRG is called. The applications driver then draws the next screen 4.0.

1) Graph Accumulated 18KIP vs Distress or PSI
2) Build Model File
3) Create Distress File
* File name: APPLICAT.PRG
* Program name: application
* Project 2456: Texas Flexible Pavement Database
* TAMU/TTI
* Written by: Victor Wong
* Created on: July 14, 1988
* Last updated: July 26, 1988
* Purpose: To call on application programs.

* set flag
SET SAFETY OFF

* initialize variables
REP = .T.

* repeat option menu until escape key is pressed
DO WHILE REP
OPTION = ""
* redisplay menu until valid option or escape is entered
DO WHILE .NOT. (OPTION $ "123") .AND. REP
  * option menu
  CLEAR
  @ 5,24 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
  @ 6,31 SAY "APPLICATIONS MENU"
  @ 9,15 SAY "1) Graph Accumulated 18KIP vs Distress or PSI"
  @ 10,15 SAY "2) Create Model File"
  @ 11,15 SAY "3) Create Distress File"
  @ 14,45 SAY "Option === " GET OPTION PICTURE "X"
  @ 4,9 TO 16,70 DOUBLE

  * read option
  READ

  * set repeat flag to false if escape key is pressed
  IF READKEY() = 12
    REP = .F.
  ENDIF

  * warning for invalid option entry
  IF .NOT. (OPTION $ "123") .AND. REP
    @20,10 SAY "Enter only 1, 2 or 3. "
    WAIT
    @20,10 SAY ""
  ENDIF
ENDDO

* carry out option
IF REP
  DO CASE
CASE OPTION = "1"
    DO \pavedb\application\graph18k\GRAPH18K
CASE OPTION = "2"
    DO \pavedb\application\model\MODEL
CASE OPTION = "3"
    CLEAR
    MRUN = " "
@ 10, 5 SAY "This program is going to create the Distress Database"
@ 11, 5 SAY " File. It will take the approximately 35 HOURS to run."
@ 13, 5 SAY "The DISTRESS file makes use of the monitoring data.
If"
@ 14, 5 SAY " the latest data is required, please run the
Pavement"
@ 15, 5 SAY " Condition Data Program (Option 2 on the Edit &
Update"
@ 16, 5 SAY " Menu) before this program."
@ 19, 5 SAY "Do you want to continue (Y/N) ? " GET MRUN
READ
IF MRUN = "Y"
    DO \PAVEDB\EDITUPD\DISTRESS\DISTRESS
    DO \PAVEDB\EDITUPD\DISTRESS\DISTVISL
    CLEA TYPE
ENDIF
RELEASE MRUN
ENDCASE
ENDIF
ENDDO

* reset flags
SET SAFETY ON

RETURN
Section 2: Accumulated 18KIP vs Alligator Cracking/Rutting/PSI
GENERAL NARRATIVE

These programs graph the area of alligator cracking versus accumulated 18KIP, area of rutting versus accumulated 18KIP, and PSI versus accumulated 18 KIP for a selected section (SID). The graph is displayed on the monitor. The GRAPH18K.PRG program prompts for the distress type and the section identification number. It then retrieves the 18KIP and distress information from DISTRESS.DBF calls GRAPH18K.PAS to calculate the rho and beta value and constructs the graph. The program specifications give more detailed information about the programs. The programs and the DAT files are stored in the subdirectory PAVEDB\EDIT\UPDT\APPLICAT\GRAPH18K.

The DBF files are stored in the subdirectory PAVEDB\FILES. The NDX files are stored in the subdirectory PAVEDB\FILES. The following are provided after this narrative:

- Program Flow Diagram
- Program Specification
  - GRAPH18K.PRG
- Graph Menu Screen
- File Layout - DISTTYP.DBF
- File Layout - DISTTYP.DAT
- File Layout - DISTRESS.DAT
- Program Listing - GRAPH18K.PRG
- Program Specification
  - GRAPH18K.PAS
- Program Listings
  (in the order mentioned in the Program Specifications)
Program Flow for Graphing $18k$ vs Area Distress or PSI

GRAPH18K.PRG

DISTRESS.DAT

Intermediate DISTRESS data file

Graph display

option menu

DISTRESS.DBF

DISTRESS database
Program Name: GRAPH18K.PRG

Purpose: To display a menu containing options to graph accumulated 18KIP versus the area of distress for alligator cracking or rutting, or to graph accumulated 18KIP versus PSI. This program will find and pass to a file the distress and 18KIP information needed for the chosen section identification number. With the retrieved data, a program is called to perform the processing and graphing.

Procedure/Edits:

1. The graphing program will be done using Turbo Pascal. Thus, the information needed to calculate the rho and beta values and to fit the curve is passed to the Pascal program (GRAPH18K.PAS) through an intermediate data file (DISTRESS.DAT). The distress type is passed in the file DISTTYP.DAT.

2. The data needed for program GRAPH18K.PAS is retrieved by locating the data of the chosen section from the DISTRESS dBASE file. The Location File is checked to determine if the requested SID is a valid number.

Input Files: 1. Distress dBASE file and index (DISTRESS.DBF) (DISTRESS.NDX).

2. Location dBASE file and index (LOCATION.DBF) (LOCATION.NDX).

3. Distress Type dBASE file (DISTTYP.DBF).

Intermediate Files:

1. Intermediate distress data file (DISTRESS.DAT).

2. Intermediate distress type file (DISTTYP.DAT).

Programs Called: GRAPH18K.PAS
Accumulated 18KIP vs Alligator Cracking/Rutting/PSI

Program Specification 1
Graph Menu Screen

TEXAS FLEXIBLE PAVEMENT DATABASE
GRAPH 18K VS DISTRESS OR PSI

1) 18KIP vs Alligator Cracking
2) 18KIP vs Rutting
3) 18KIP vs PSI

Option ==> 
SID Number ==> ___
**Accumulated 18 KIP vs Alligator Cracking/Rutting/PSI**

**Program Specification 1**

**File Layout**

<table>
<thead>
<tr>
<th>Field</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISTTYP</td>
<td>1C</td>
<td>Graph Option</td>
</tr>
</tbody>
</table>

File Name: DISTTYP.DBF
**Accumulated 18 KIP vs Alligator Cracking/Rutting/PSI**

**Program Specification 1**

**File Layout**

**File Name:** DISTTYP.DAT

<table>
<thead>
<tr>
<th>Column</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Graph Option</td>
</tr>
</tbody>
</table>
## File Name: DISTRESS.DAT

<table>
<thead>
<tr>
<th>Columns</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 4</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>5 - 6</td>
<td>Year</td>
</tr>
<tr>
<td>7 - 17</td>
<td>Annual Cumulative 18KIP Equivalent Axle Load</td>
</tr>
<tr>
<td>18</td>
<td>Patching Area Distress Value</td>
</tr>
<tr>
<td>19</td>
<td>Rutting Area Distress Value</td>
</tr>
<tr>
<td>20</td>
<td>Alligator Cracking Area Distress Value</td>
</tr>
<tr>
<td>21 - 27</td>
<td>Serviceability Index Mean Value</td>
</tr>
</tbody>
</table>
Accumulated 18KIP vs Alligator Cracking/Rutting/PSI

Program Specification 1
Program Listing

* File name: GRAPH18K.PRG
* Program name: graph18k
* Project 2456: Texas Flexible Pavement Database
* TAMU/TTI
* Written by: Victor Wong
* Created on: July 25, 1988
* Last updated: August 1, 1988
* Purpose: To set up menu for graphing 18KIP vs certain
type of area distress or PSI.

* set up databases
SELECT A
USE \pavedb\files\LOCATION INDEX \pavedb\indexes\LOCATION
SELECT B
USE \pavedb\files\DISTRESS INDEX \pavedb\indexes\DISTRESS SELECT C
USE \pavedb\files\DISTTYP

* initialize variables
PRIVATE REP
PRIVATE OPTION
REP = .T.

* repeat option menu until escape key is pressed
DO WHILE REP
  OPTION = " "
  SID = A->SID NO
  * redisplay menu until valid option/SID or escape key
  DO WHILE .NOT. (OPTION $ "123") .AND. REP
    * option menu
    CLEAR
    @ 5,23 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
    @ 7,23 SAY " GRAPH 18KIP VS DISTRESS OR PSI"
    @ 10,25 SAY "1) 18KIP vs Alligator Cracking"
    @ 11,25 SAY "2) 18KIP vs Rutting"
    @ 12,25 SAY "3) 18KIP vs PSI"
    @ 14,25 SAY "          Option ==> ";
      GET OPTION PICTURE "X"
    @ 15,25 SAY "          SID Number ==> ";
      GET PICTURE "$9999"
    @ 3,9 TO 18,70 DOUBLE
  * read option
  READ

  * set repeat flag to false if escape key is pressed
  IF READKEY() = 12
REP = .F.
ENDIF

* warning for invalid option
IF .NOT. (OPTION "$ "123") .AND. REP
  @20,10 SAY "Enter only 1, 2 or 3."
  WAIT
  @20,10 SAY "  "
ENDIF

ENDDO

* carry out option
IF REP
  SELECT A
  SEEK SID
  IF .NOT. FOUND()
    GO TO 1
    @20,10 SAY "Invalid SID Number."
    WAIT
    @20,10 SAY "  "
  ELSE
    * copy the distress information of the sid number to
    * a temporary file
    SELECT B
    SEEK STR(SID,4)
    IF FOUND()
      COPY TO \pavedb\applicat\graph18k\DISTRESS.DAT ;
      FIELDS SID NO,YEAR,A18KEAL,PATCH,UITT,ALLGCR,SIMEAN;
      WHILE SID NO=SID TYPE SDF
      SELECT C
      GO TO 1
      REPLACE DISTYP WITH OPTION
      COPY TO \pavedb\applicat\graph18k\DISTYP.DAT;
      RECORD 1 TYPE SDF
    ELSE
      @20,10 SAY "Cannot find SID number in DISTRESS file."
      WAIT
      @20,10 SAY "  "
    ENDFI
    * perform the graphing
    RUN \pavedb\applicat\graph18k\GRAPH18K
    ENDFI
  ENDI
ENDDO

* close databases
CLOSE ALL
RETURN
Program Name: GRAPH18K.PAS

Purpose: To graph the area of distress curve for alligator cracking/rutting or the PSI curve for the section identification number the user chooses. The curve is plotted after the program finds the rho and beta values of the curve.

Edit/Procedure Information:

1. This graphing program is written in Turbo Pascal. The information needed to calculate the rho and beta values, and to fit the curve is passed from the program GRAPH18K.PRG through an intermediate data file (DISTRESS.DAT). The distress type to be graphed is passed in DISTTYF.DAT.

2. The equation for the area distress curve of alligator cracking and rutting is

\[ \% \text{ area distress} = \exp\left(-\rho \left( \frac{\text{accumulated 18KIP}}{\text{beta}} \right) \right). \]

Refer to the program comments for additional information about solving for rho and beta.

3. The equation for the PSI curve is

\[ \text{PSI} = \text{initial PSI} - (\text{initial PSI} - \text{final PSI}) \times \exp\left(-\rho \left( \frac{\text{accumulated 18KIP}}{\text{beta}} \right) \right). \]

Refer to the program comments for additional information about solving for rho and beta.

4. The area of distress data for alligator cracking and rutting is collected using five discrete values representing the percentage of area of distress. The value "9" stands for missing data, and "0" stands for no area of distress. Otherwise, the following shows how the other three values are assigned:
The program attempts to compensate for an irregular shape area of distress curve for alligator cracking rutting where the irregularity is due to patching. The percentage area value for patching is added to the alligator cracking/rutting value. This eliminates the dropping portion of the curve due to decreased area distress resulting from patching. The coding assignments for patching are as shown below.

<table>
<thead>
<tr>
<th>Code</th>
<th>Distress</th>
<th>Plotting Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1 - 10</td>
<td>5.5</td>
</tr>
<tr>
<td>2</td>
<td>11 - 50</td>
<td>30.5</td>
</tr>
<tr>
<td>3</td>
<td>&gt; 50</td>
<td>50.0</td>
</tr>
</tbody>
</table>

If the areas of distress still decreases after the patching has been added to the alligator cracking or rutting value, the point is dropped from the curve.

As in item 5, when the PSI improves, the point is dropped from the curve.

**Input File(s):**

1. Intermediate distress data file (DISTRESS.DAT)
2. Intermediate distress type file (DISTTYP.DAT)

**PASCAL Procedures Called:**

- DECLARE.PAS
- GRAPH.P
- MYGRAPH.PAS
- RHOBETA.PAS
- FITCURVE.PAS
- PLOIGRPH.PAS
File Layouts:

1. DISTYP.DAT (Refer to File Layout 2 in Program Specification 1).

2. DISTRESS.DAT (Refer to File Layout 3 in Program Specification 1).
Accumulated 18KIP vs Alligator Cracking/Rutting/PSI

Program Specification 2
Program Listing

(* File name: GRAPH18K.PAS
 * Program name: graph18k
 * Project 2456: Texas Flexible Pavement Database Conversion
 * TAMU/TTI
 * Written by: Victor Wong
 * Created on: Feb 19, 1988
 * Last updated: June 27, 1988
 * Purpose: This program creates an onscreen menu
 * inquiring user for the type of distress and
 * sid number. It then call a procedure to find
 * the distress information of the sid number
 * from the distress database file. With the
 * information, the porcedure will process and
 * plot the value on an onscreen graph.
 *)

{>>> PROGRAM GRAPH18K <<<}
program graph18k;

{>>> constant declaration <<<}
const
  color = 15;
  blank = ' ';
  psi_final = 1.5;
  debug_flag = true;

{>>> type declaration <<<}
($I \pavedb\applicat\graph18k\declare)

{>>> variable declaration <<<}
var
  (--- variables for menu ---)
distress_type: integer;
response_ok,
continue: boolean;
response: char;

  (--- variables for data processing ---)
data_file: text;

(input data format)
sid_no: string[4]; (sid number 4N)
year: string[2]; (year 2N)
al8keal: string[11]; (18kip 11N)
patching: string[1]; (patching 1N)
rutting: string[1]; (rutting 1N)
allgcruk: string[1];  {alligator crack 1N}
simean: string[7];  {simean 75N}

:variables needed for conversion of data from strings into 
{numbers and variables needed for any numeric manipulation}
(index)
char at index: string[1];
err code
err code1:
err code2:
psi value
patch value
rutting value
allgcruk value:
patch area
rutting area
allgcruk area:
a18k1, a18k2,
a18k3, a18k4,
a18k1 carry,
a18k2 carry,
a18k3 carry,
a18k4 carry,
accum 18k1,
accum 18k2,
accum 18k3,
accum 18k4:
a18k1 zero,
a18k2 zero,
a18k3 zero,
a18k4 zero:
a18keal1:
a18keal2,
a18keal3,
a18keal4:

:variables needed for graph
psi max,
psi initial:
psi initial str:
psi,
y:
y array type;
{x value (accum. 18k value)}
x1, x2, x3, x4:
x array type;
{x value by digit group}
y factor,
x factor:
{the factor label for axis}
y min, y max,
y interval:
x min, x max,
x interval:
title:
title type;
x label,
y_label: label_type;
id_string: id_type;

{--- other variables in the program ---}
c: char;
i: integer;
found: boolean;
count: integer;

{>>> include files <<<}
($I \text{pavedb}\text{\textbackslash applicant}\text{\textbackslash graph18k}\text{\textbackslash graph.p})
{file with turbo graphics procedure}
($I \text{pavedb}\text{\textbackslash applicant}\text{\textbackslash graph18k}\text{\textbackslash mygraph})
{file with my graphics procedures}
($I \text{pavedb}\text{\textbackslash applicant}\text{\textbackslash graph18k}\text{\textbackslash rhobeta})
{file with procedure to calculate}
{rho and beta}
($I \text{pavedb}\text{\textbackslash applicant}\text{\textbackslash graph18k}\text{\textbackslash fitcurve})
{file with procedure to fit curve}
($I \text{pavedb}\text{\textbackslash applicant}\text{\textbackslash graph18k}\text{\textbackslash plotgraph})
{file with procedure to plot graph}

{>>> beginning of program <<<}
beg

{--- clear the screen ---}
ClrScr;

{--- initialize some variables---}
accum_18k1 := 0;
accum_18k2 := 0;
accum_18k3 := 0;
accum_18k4 := 0;

{find the distress type}
assign (data_file, '\text{pavedb}\text{\textbackslash applicant}\text{\textbackslash graph18k}\text{\textbackslash disttyp.dat}');
reset (data_file);
readln (data_file, distress_type);
close (data_file);

{set up input file for data}
assign (data_file, '\text{pavedb}\text{\textbackslash applicant}\text{\textbackslash graph18k}\text{\textbackslash distress.dat}');
reset (data_file);

count := 0; {initialize the count of valid data to 0}
psi max := 0.0; {initialize the maximum psi value to 0.0}
patch_area := 0.005; {initialize the patch area to 0.0}

{while not eof}
{read data, accumulate 18kip value }
{and store valid distress or psi value}
while not eof(data_file)
do begin
readln (data_file, sid_no, year, a18keal, patching, rutting, allgcrk, simean);
{convert the a81keal string value to 3 integer values}
{so that the concatenation of a18k1, a18k2 and a18k3  }
{will give the whole 18keal value
}
{if the err_code is non-zero, it means that digits
{are missing at those positions and it is the same as
{having zeros there
}
{the positions of leading zeros are indicated by the
{a18kx_zero flag
}

{split a18keal value string into 4 sections }
{{xx,xxx,xxx,xxx}}
 a18keal1:= copy(a18keal,1,2);
 a18keal2:= copy(a18keal,3,3);
 a18keal3:= copy(a18keal,6,3);
 a18keal4:= copy(a18keal,9,3);

{delete leading blank spaces of a18kealx string}
{and store convert the string to integer a18kx values}
index:= 1;
 char_at_index:= copy (a18keal1,index,1);
while (char_at_index = blank) and (index < 2) do
 begin
  index:= index + 1;
  char_at_index:= copy (a18keal1,index,1);
 end;
 if index < 2 then
  val (copy(a18keal1,index,3-index),a18k1,err_code)
 else if char_at_index <> blank then
  val (copy (a18keal1,index,3-index),a18k1,err_code)
 else
  begin
   a18k1:= 0;
   a18k1_zero:= true;
  end;

 index:= 1;
 char_at_index:= copy (a18keal2,index,1);
while (char_at_index = blank) and (index < 3) do
 begin
  index:= index + 1;
  char_at_index:= copy (a18keal2,index,1);
 end;
 if index < 3 then
  val (copy(a18keal2,index,4-index),a18k2,err_code)
 else if char_at_index <> blank then
  val (copy (a18keal2,index,4-index),a18k2,err_code)
 else
  begin
   a18k2:= 0;
   a18k2_zero:= true;
  end;

 index:= 1;
char_at_index := copy (a18keal3, index, 1);
while (char_at_index = blank) and (index < 3) do
begin
  index := index + 1;
  char_at_index := copy (a18keal3, index, 1);
end;
if index < 3 then
  val (copy(a18keal3, index, 4-index), a18k3, err_code)
else if char_at_index <> blank then
  val (copy (a18keal3, index, 4-index), a18k3, err_code)
else
begin
  a18k2 := 0;
  a18k3_zero := true;
end;

index := 1;
char_at_index := copy (a18keal4, index, 1);
while (char_at_index = blank) and (index < 3) do
begin
  index := index + 1;
  char_at_index := copy (a18keal4, index, 1);
end;
if index < 3 then
  val (copy(a18keal4, index, 4-index), a18k4, err_code)
else if char_at_index <> blank then
  val (copy (a18keal4, index, 4-index), a18k4, err_code)
else
begin
  a18k4 := 0;
  a18k4_zero := true;
end;

(if the a18keal value is not zero continue processing)
if not (a18k1_zero and a18k2_zero and a18k3_zero and a18k4_zero) then
begin
  (increment the accumulated 18kip value)
  accm 18k4 := accm 18k4 + a18k4;
  a18k4_carry := accm 18k4 div 1000;
  accm 18k4 := accm 18k4 mod 1000;
  accm 18k3 := accm 18k3 + a18k3 + a18k4_carry;
  a18k3_carry := accm 18k3 div 1000;
  accm 18k3 := accm 18k3 mod 1000;
  accm 18k2 := accm 18k2 + a18k2 + a18k3_carry;
  a18k2_carry := accm 18k2 div 1000;
  accm 18k2 := accm 18k2 mod 1000;
  accm 18k1 := accm 18k1 + a18k1 + a18k2_carry;

  (--- check which distress type we are using ---)
  (--- and do the necessary processing ---)
  case distress type of
    1: {alligator cracking}
begin
  (convert alligator crack data from string to integer)
  val (allgcrk,allgcrk_value,err_code1);

  (convert patching data from string to integer)
  val (patching,patch_value,err_code2);

  (if the conversion is successful)
  if (err_code1 = 0) and (err_code2 = 0) then

  (check if the value is valid i.e. <> 9)
  if (allgcrk_value <> 9) then
    begin
      (increment valid data count)
      count := count + 1;

      (determine the percentage of alligator)
      (cracking area)
      case allgcrk_value of
        0: allgcrk_area := 0.005;
        1: allgcrk_area := 0.055;
        2: allgcrk_area := 0.305;
        3: allgcrk_area := 0.50;
      end;

      (determine the percentage of patching)
      (area)
      if patch_value <> 9 then
        case patch_value of
          0: patch_area := 0.005;
          1: patch_area := 0.055;
          2: patch_area := 0.305;
          3: patch_area := 0.50;
        end;

      (take the sum of the two area %)
      (as the value to be plotted)
      y[count] := (allgcrk_area + patch_area);
      if y[count] >= 1.0 then
        y[count] := 0.985;

      (store the new accumulated 18kip value)
      (in x arrays)
      x1[count] := accum_18k1;
      x2[count] := accum_18k2;
      x3[count] := accum_18k3;
      x4[count] := accum_18k4;
    end
  else
    begin
      if err_code1 <> 0 then
        begin
          gotoxy (22,14);
        end
  end
write ('***WARNING***');
write (' bad data encountered');
gotox (22,15);
write (');
write ('INVALID CODE FOR ALLIGATOR ');
write ('CRACKING ');
gotox (22,17);
write ('Still processing. ');
write (' Please wait ...');
end;
if err_code2 <> 0 then
begin
    gotox (22,14);
    write ('***WARNING***');
    write (' bad data encountered');
    gotox (22,15);
    write ('');
    write ('INVALID CODE FOR PATCHING');
    write (');
    gotox (22,17);
    write ('Still processing.');
    write (' Please wait ...');
end;
end;
end;

2: (rutting)
begin
    (convert rutting data from string to integer)
val (rutting,rutting_value,err_code1);

    (convert patching data from string to integer)
val (patching,patch_value,err_code2);

    (if the conversion is successful)
if (err_code1 = 0) and (err_code2 = 0) then

    (check if the value is valid i.e. <> 9)
if (rutting_value <> 9) then
begin
    (increment valid data count)
count := count + 1;

    (determine the percentage of rutting)
(area)
case rutting_value of
    0: rutting_area := 0.005;
    1: rutting_area := 0.13;
    2: rutting_area := 0.38;
    3: rutting_area := 0.50;
end;

    (determine the percentage of patching)
(area)
if patch_value <> 9 then
  case patch_value of
    0: patch_area:= 0.005;
    1: patch_area:= 0.055;
    2: patch_area:= 0.305;
    3: patch_area:= 0.5;
  end;

  {take the sum of the two area %}
  {as the value to be plotted        }
  y[count]:= (rutting_area + patch_area);
  if y[count] >= 1.0 then
    y[count]:= 0.985;

  {store the new accumulated 18kip value}
  {in x arrays}
  x1[count]:= accum_18kl;
  x2[count]:= accum_18k2;
  x3[count]:= accum_18k3;
  x4[count]:= accum_18k4;
end
else
  begin
    if err_code1 <> 0 then
      begin
        gotoxy (22,14);
        write ('***WARNING*** ');
        write ('bad data encountered');
        gotoxy (22,15);
        write ('           ');
        write ('INVALID CODE FOR RUTTING   ');
        gotoxy (22,17);
        write ('Still processing.    ');
        write ('Please wait ...');
      end;
    if err_code2 <> 0 then
      begin
        gotoxy (22,14);
        write ('***WARNING*** ');
        write ('bad data encountered');
        gotoxy (22,15);
        write ('           ');
        write ('INVALID CODE FOR PATCHING  ');
        gotoxy (22,17);
        write ('Still processing.    ');
        write ('Please wait ...');
      end;
  end;
end;

3: (psi)
begin
  (convert psi data from string to real)
  val (simean,psi_value,err_code);

488
{if the conversion is successful}
if (err_code = 0) then

if psi_value <> 9.99999 then
    begin
    count:= count + 1;

    {store the psi value in the psi array}
    if psi_value = 0.0 then
        psi[count]:= 0.0005
    else
        psi[count]:= psi_value;

    {check for the maximum psi value}
    if psi_value > psi_max then
        psi_max:= psi_value;

    {store the new accumulated 18kip value}
    {in x arrays}
    x1[count]:= accum_18k1;
    x2[count]:= accum_18k2;
    x3[count]:= accum_18k3;
    x4[count]:= accum_18k4;
    end
else
    if err_code <> 0 then
        begin
        gotoxy (22,14);
        write ('***WARNING*** ');
        write ('bad data encountered');
        gotoxy (22,15);
        write ('');
        write ('INVALID PSI VALUE ');
        gotoxy (22,17);
        write ('Still processing. ');
        write ('Please wait ...');
        end;
    end; (case)
end; (if)
end;

close (input file)
close (data_file);

{ask user for an estimated initial psi value}
{the input value has to be larger than or }{equal to the existing maximum psi value }
response_ok:= false;
psi_initial:= 4.5;
if distress_type = 3 then
    begin
    {ask for the estimated initial psi value}
clrscr; {clear the screen}
HiRes; {set high resolution graphics}
DrawBox (80,56,560,144);
gotoxy (21,10);
write ('The program needs to know the estimated');
gotoxy (21,11);
write ('initial psi value. (> ',psi_max:4:2,', ')');
gotoxy (21,13);
write ('Initial PSI value (RETURN = 4.5) -> ');

{keep prompting for estimated initial psi value}
{if the input is smaller than the maximum psi value}
repeat
  psi_initial:= 4.5;
gotoxy (57,13);
write ('');
gotoxy (57,13);
readln (psi_initial_str);
val (psi_initial_str,psi_initial,err_code);
if (err_code <> 0) then
  begin
    gotoxy (21,14);
    write ('Input must be numeric!');
  end
else
  begin
    if psi_initial >= psi_max then
      response_ok:= true
    else
      begin
        gotoxy (21,14);
        write ('Value too small! Please reenter.');
      end;
  end;
until response_ok;

{transform the psi value to a distress value}
for i:= 1 to count do
  y[count]:= ((psi_initial-psi[count]) /
    (psi_initial-psi_final)) /10;

TextMode; {clear screen and back to text mode}
end; {if}

{find a suitable accumulated 18kip factor for the y axis on the graph}
case distress_type of
  1, 2: y_factor:= 'x 1';
  3: y_factor:= 'x 1';
end; {case}

{find a suitable accumulated 18kip factor for the x axis on the graph}
(NOTE: 18 kip values are taken as the 3 most significant)  
{ digits i.e. a number 1,234,567 will be expressed as 123 x 10\^4 }
if accum_18k3=0 then
  begin
    x factor:= 'x 1';
    for i:= 1 to count do x[i]:= x4[i];
  end
else if accum_18k2 = 0 then
  begin
    if (a18k3 div 100) > 0 then
      begin
        x_factor:= 'x 10\^3';
        for i:= 1 to count do x[i]:= x3[i];
      end
    else if (a18k3 div 10) > 0 then
      begin
        x_factor:= 'x 10\^2';
        for i:= 1 to count do
          x[i]:= x3[i]*10 + x4[i] div 100;
      end
    else
      begin
        x_factor:= 'x 10';
        for i:= 1 to count do
          x[i]:= x3[i]*100 + x4[i] div 10;
      end;
  end
else if accum_18k1 = 0 then
  begin
    if (a18k2 div 100) > 0 then
      begin
        x_factor:= 'x 10\^6';
        for i:= 1 to count do x[i]:= x2[i];
      end
    else if (a18k2 div 10 ) > 0 then
      begin
        x_factor:= 'x 10\^5';
        for i:= 1 to count do
          x[i]:= x2[i]*10 + x3[i] div 100;
      end
    else
      begin
        x_factor:= 'x 10\^4';
        for i:= 1 to count do
          x[i]:= x2[i]*100 + x3[i] div 10;
      end;
  end
else
  begin
    if (a18k1 div 10) > 0 then
      begin
        x_factor:= 'x 10\^8';
        for i:= 1 to count do
      end
end

491
x[i] := x1[i] * 10 + x2[i] div 10;
    end
else
    begin
    x_factor := 'x 10^7';
    for i := 1 to count do
      x[i] := x1[i] * 100 + x2[i] div 10;
    end;
end;

{set axis parameter before plotting the graph}
{set the appropriate values for y axis scale}
case distress_type of
  1, 2 : begin
    y_min := 0.0;
    y_max := 1.0;
    y_interval := 0.25;
  end;
  3 : begin
    y_min := 0.0;
    y_max := 5.0;
    y_interval := 1.0;
  end;
end;

{set the appropriate values for x axis scale}
if (x[count] div 10) = 0 then
  begin
    x_min := 10;
    x_max := 0;
    x_interval := 1;
  end
else
  begin
    x_min := 0;
    {NOTE: the value 100 in the following equation for}
    { x max makes the curve more centered }
    { changing 100 to 10 will make the curve to }
    { shift more to the right and the range of }
    { the x scale to be smaller }
    x_max := ((x[count] div 100) + 1) * 100;
    x_interval := x_max div 10;
  end;
end;

{set title, id and label}
case distress_type of
  1 : begin
    title := 'ALLIGATOR CRACK vs ACCUMULATED 18KIP';
    id_string := 'SID #: ' + sid_no;
    x_label := 'accumulated 18kip';
    y_label := '% area distress';
  end;
  2 : begin
    title := 'RUTTING vs ACCUMULATED 18KIP';
  end;
end;
id_string := 'SID #: ' + sid_no;
x_label := 'accumulated 18kip';
y_label := '% area distress';
end;

3: begin
    title := 'PSI vs ACCUMULATED 18KIP';
    id_string := 'SID #: ' + sid_no;
    x_label := 'accumulated 18kip';
    y_label := 'psi';
    end;
end; {case}

{plot the graph and points}
case distress_type of
    1, 2: PlotGraph (title, id_string, x_label, y_label,
    x_factor, y_factor, x, y, psi, psi_initial,
    count, x_min, x_max, x_interval,
    y_min, y_max, y_interval, d);
    3: PlotGraph (title, id_string, x_label, y_label,
    x_factor, y_factor, x, y, psi, psi_initial,
    count, x_min, x_max, x_interval,
    y_min, y_max, y_interval, p);
end; {case}

{>>> ending of program <<<}
end.
(* File Name: DECLARE.PAS)

type (type declarations needed for PlotGraph parameters)
  title_type = string [60];
  id_type = string [12];
  label_type = string [30];
  factor_type = string [10];
  x_type = integer;
  y_type = real;
  x_array_type = array [1..72] of x_type;
  y_array_type = array [1..72] of y_type;
  graph_type = (d, p);
PROGRAM LISTING


{>>> PROCEDURE: DRAWBOX (X1, Y1, X2, Y2) <<<
>>> This procedure draws a box using pixel coordinates (X1,Y1) as <<<
>>> the top left corner and (X2, Y2) as the bottom right corner. <<<

procedure DrawBox (x1, y1, x2, y2: integer);

begin
    Draw (x1, y1, x1, y2, 1);
    Draw (x1, y2, x2, y2, 1);
    Draw (x2, y2, x2, y1, 1);
    Draw (x2, y1, x1, y1, 1);
end;

{>>>>> PROCEDURE: PLOTPoint (X, Y, OPTION) <<<
>>>>> This procedure plots a point on the screen as one of the point <<<
>>>>> options available using the given x and y coordinates. <<<

procedure PlotPoint (x, y, option: integer);

begin
    case option of
        1: begin (cross)
            Draw (x-2, y-2, x+2, y+2, 1);
            Draw (x-2, y+2, x+2, y-2, 1);
        end;
        2: begin (plus)
            Draw (x-2, y, x+2, y, 1);
            Draw (x, y-2, x, y+2, 1);
        end;
        3: begin (asterisk)
            Draw (x-2, y, x+2, y, 1);
            Draw (x-2, y-2, x+2, y+2, 1);
            Draw (x-2, y+2, x+2, y-2, 1);
        end;
        4: begin (square)
            Draw (x-1,y-1,x+1, y-1, 1);
            Draw (x-1, y ,x+1, y , 1);
            Draw (x-1,y+1,x+1, y+1, 1);
        end;
    end;
end;

"Z

495
Accumulated 18KIP vs Alligator Cracking/Rutting/PSI

Program Specification 2
Program Listing

File name: RHOBETA.PAS
Written by: Victor Wong
Written on: Feb 23, 1988
Last updated: Mar 8, 1988
Procedure name: find rho beta
Purpose: To find the rho and beta values for area or severity distress types.

Methodology: Equation being used is as follows.

\[
\text{rho beta}
\]
\[
distress = \exp \left( - \left( -\frac{1}{\text{traffic}} \right) \right)
\]

When rho = traffic,

\[
\text{beta}
\]
\[
distress = \exp \left( - (1) \right)
\]
\[
= \exp (-1)
\]
\[
= 0.367
\]

To find the rho for a distress vs traffic curve, this methodology takes the first point as the reference and does a linear estimate of where the rho should be for each following point. For each of the rhos, the beta can be found by solving the above equation. After considering all the points, the maximum beta value is taken and the corresponding rho is accepted. These two values will be the final rho and beta for the curve.

The linear estimation for the rho value is determined as follows.

rho estimation for a point

\[
distress \text{ when rho equals traffic} - \text{distress of point 1}
\]
\[
= \frac{\text{distress of the point} - \text{distress of point 1}}{x \ (\text{traffic for the point} - \text{traffic of point 1}) + \text{traffic (1)}}
\]
\[
i.e. rho(i) = \frac{0.367 - \text{distress (1)}}{\text{distress(i) - distress(1)}}
\]
*(traffic(i) - traffic(1)) + Traffic (1)*

* By rearranging the distress vs traffic equation, we get *

\[
\ln(-\ln(\text{distress}))
\]

\[
\frac{\text{rho}}{\ln(\text{traffic})}
\]

*(Note that distress value is less than one. Thus - \ln(\text{distress}) will give a positive number. Otherwise, \ln(-\ln(\text{distress})) may be undefined.)*

*

procedure find_rho_beta
  (traffic: x_array_type; \{traffic values\}
  distress: y_array_type; \{distress values\}
  no_of_pt: integer; \{no of traffic-distress\}
  \{pairs\}
  var final_rho, final_beta : real);

{>>> variables declaration <<<}

var
  denominator,
  rho,
  beta,
  max_beta : real;
  index,
  i : integer;
  rho_of_pt,
  beta_of_pt: array[1..72] of real;

{>>> beginning of the procedure find_rho_beta<<<}
begin
  (If there is number of points is zero or less, then rho and beta values are zeros.)

  if (no_of_pt = 0) or (no_of_pt < 1) then
    begin
      rho:= 0.0;
      beta:= 1.0;
    end

  {If number of points = 88, then rho = the first value in the traffic area
  beta = 0.5. }

  else if (no_of_pt = 88) then
    begin
      rho:= traffic[1];
      beta:= 0.5;

  else
end

{If number of points = 99,
 then rho = 17.5 * 12.0
   beta = 1.0.)

else if (no_of_pt = 99) then
begin
   rho:= 17.5 * 12.0;
   beta:= 1.0;
end

(If number of points is greater than zero and not 88 or 99,
 then apply the methodology described above to find the rho and beta.)

else
begin
   {initialize the rhos and betas for the points to zero}
   for i:= 1 to no_of_pt do
   begin
      rho_of_pt[i]:= 0.0;
      beta_of_pt[i]:= 0.0;
   end;

   {for all the n points, find the rhos and betas}
   for i:= 2 to no_of_pt do
   begin
      denominator:= 1.0 * (distress[i] - distress[1]);
      if (denominator <> 0.0) then
      begin
         rho_of_pt[i]:= (0.367 - distress[1])/denominator
                        * (traffic[i]-traffic[1])
                        + traffic[1];
         if rho_of_pt[i] > 0.0 then
         begin
            beta_of_pt[i]:= - ln (1.0 * distress[i]);
            beta_of_pt[i]:= ln(beta_of_pt[i]) /
                            ln(rho_of_pt[i]/traffic[i]);
         end;
      end;
   end;

   {find the maximum of the betas}
   max_beta:= beta_of_pt[1];
   index:= 1;
   for i:= 2 to no_of_pt do
   if max_beta < abs (beta_of_pt[i]) then
   begin
      max_beta:= beta_of_pt[i];
      index:= i;
   end;

   (choose the maximum beta and
    the rho value corresponding to the beta)
rho := rho_of_pt[index];
beta := max_beta;
end;

{return the final values for rho and beta}
final_rho := rho;
final_beta := beta;

{>>> ending of the procedure find_rho_beta <<<}
end;
procedure fitcurve (x: x_array_type;
        y: y_array_type;
        no_of_pts: integer;
        x_min, x_max, x_interval: x_type;
        y_min, y_max, y_interval: y_type;
        psi_initial: real;
        distress_or_psi: graph_type);

{>>> constant declaration <<<}
const
  window_border_space = 10;
  window_width_space = 60;
  pixels_per_line = 8;
  pixels_per_char = 8;
  y_axis_lines = 8;

{>>> variable declaration <<<}
var
  rho, beta: real;

  {--- variables necessary for defining graph window ---}
  x_right, x_left,
  y_top, y_bottom: integer;

  {--- variables necessary for drawing the curve ---}
  spaces_per_interval,
  lines_per_interval: integer;
  delta_x: x_type;
  x_begin, x_end: x_type;
  y_begin, y_end: y_type;
  x_begin_coord,
  x_end_coord,
  y_begin_coord,
  y_end_coord : integer;
  response: char;
  y_valid: boolean;

{>>> beginning of procedure fitcurve <<<}
begin
  {--- find the rho and beta of the curve ---}
  {--- detail on the equation of the curve---}
  {--- is in file RHOBETA ---}
  find_rho_beta (x, y, no_of_pts, rho, beta);

  {--- process only when rho is positive ---}
  if rho > 0 then
    begin
      {tell user to wait for curve fitting}
      GotoXY (30,8);
      write ('Wait !! Fitting the curve.');

      {--- set window boundary ---}
      y_top:= 8 * pixels_per_line;
      y_bottom:= 17 * pixels_per_line;
      x_left:= window_border_space * pixels_per_char;
      x_right:= (window_width_space+window_border_space)
        * pixels_per_char;

      {--- set the axis interval ---}
      spaces_per_interval:= window_width_space
        div ((x_max-x_min) div x_interval);
      lines_per_interval:= y_axis_lines div
        round((y_min-y_max)/y_interval);

      {--- initialize variables needed for drawing the curve ---}
      {--- incrementally ---}
      delta_x := x_interval div 2;
      x_begin:= 1;
      x_end:= x_begin + delta_x;
      y_end:= 0;

      {--- keep drawing the curve incrementally as long as ---}
      {--- the x coordinate does not exceed the maximum x ---}
      {--- value and the y coordinate does not exceed the ---}
      {--- maximum y value ---}
      while (x_end < x_max) and (y_end < y_max) do
        begin
          if distress_or_psi = d then
            begin
              {this demostrates how the curve equation is being
               transformed to the one which is used in this program}
              y = exp (- (rho/x) ^ beta)
              => ln y = - (rho/x) * beta
              => ln (- ln y) = beta * ln (rho/x)
              => ln (ln y) = beta * ln (rho/x) since ln y <= 0
              => ln y = - exp (beta * ln (rho/x)) since ln y <= 0
              => y = exp (- exp (beta * ln (rho/x)))
            end
          {use the equation to calculate the y value of the}
          {starting point of a curve segment}
          y_begin:= beta * ln (rho/x_begin);

  501
if y_begin < 88 then
begin
  y_begin:= exp (y_begin);
  if y_begin < 88 then
    y_begin:= exp (- y_begin)
  else
    y_begin:= 0;
end
else
  y_begin:= 0;
if y_begin < 0.005 then y_begin:= 0;

{use the equation to calculate the y value of the ending (point)
(of a curve segment)
}y_end:= beta * ln (rho/x_end);
if y_end < 88 then
begin
  y_end:= exp (y_end);
  if y_end < 88 then
    y_end:= exp (- y_end)
  else
    y_end:= 0;
end
else
  y_end:= 0;
if y_end < 0.005 then y_end:= 0;
end;

if distress_or_psi = p then
begin
  {this demonstrates how the curve equation is being
   transformed to the one which is used in this program
   \begin{align*}
   \frac{\rho \beta}{P} &= P - (P - P_0) \exp - (-\frac{f}{18 \text{kip}}) \\
   \end{align*}
   or
   \begin{align*}
   y &= \psi_{\text{initial}} - (\psi_{\text{initial}} - 1.5) \exp(- (\rho/x)^\beta) \\
   \Rightarrow \quad (\psi_{\text{initial}} - y) &= \exp(- (\rho/x)^\beta) \\
   &= \exp(- \exp(\beta \ln(\rho/x))) \\
   \Rightarrow \quad (\psi_{\text{initial}} - y) &= \exp(- \exp(\beta \ln(\rho/x))) \\
   \Rightarrow \quad y &= (\psi_{\text{initial}} - y) - \\
   &\quad (\psi_{\text{initial}} - 1.5) \ast \\
   &\quad \exp(- \exp(\beta \ln(\rho/x)))
   \end{align*}
\}

502
{use the equation to calculate the y value of the}
{starting point of a curve segment}
y_begin:= beta * ln (rho/x_begin);
if y_begin < 88 then
begin
    y_begin:= exp (y_begin);
    if y_begin < 88 then
        y_begin:= psi_initial -
        (psi_initial-1.5) * exp (- y_begin)
    else
        y_begin:= psi_initial;
end
else
    y_begin:= psi_initial;
if y_begin < 0.005 then y_begin:= 0;

{use the equation to calculate the y value of the}
{ending point of a curve segment}
y_end:= beta * ln (rho/x_end);
if y_end < 88 then
begin
    y_end:= exp (y_end);
    if y_end < 88 then
        y_end:= psi_initial -
        (psi_initial-1.5) * exp (- y_end)
    else
        y_end:= psi_initial;
end
else
    y_end:= psi_initial;
if y_end < 0.005 then y_end:= 0;
end;

{calculate the window coordinates using the x }
{and y values}
x_begin_coord:= trunc (x_begin * ((spaces_per_interval
    * pixels_per_char)
    / x_interval));
y_begin_coord:= trunc (y_begin * ((lines_per_interval
    * pixels_per_line)
    / y_interval));
x_end_coord:= trunc (x_end * ((spaces_per_interval
    * pixels_per_char)
    / x_interval));
y_end_coord:= trunc (y_end * ((lines_per_interval
    * pixels_per_line)
    / y_interval));

{if the y coordinate is 0, increment it by 1 pixel so }
{that the curve will line up with the x axis since the}
{x axis is 1 to 2 pixels from the window boundary}
if y_begin_coord = 0 then y_begin_coord:= 1;
if y_end_coord = 0 then y_end_coord:= 1;
{draw the curve segment}
Draw (x_begin_coord,
    y_bottom- y_top- y_begin_coord,
    x_end_coord,
    y_bottom- y_top- y_end_coord,
    1);

{increment the x value}
x_begin:= x_end;
x_end:= x_begin + delta_x;
end;

{--- take message off the screen ---}
GotoXY (30,8);
write ('                   ');
end
else
begin
    GotoXY (30,8);
    write ('Sorry !! Cannot fit curve due to');
    GotoXY (30,9);
    write ('decreasing distress values.'Rho, Beta);
end;

{>>> end of procedure fitcurve <<<}
end;
Accumulated 18KIP vs AlligatorCracking/Rutting/PSI

Program Specification 2
Program Listing

{***********************************************}
{*** Procedure PlotGraph : Plots a graph using the given ***}
{*** information on a black and white ***}
{*** screen. (in High resolution mode)***}
{***********************************************}
{*** NOTE: Make sure when you use this procedure, the ***}
{*** types in the procedure declaration are well ***}
{*** defined. ***}
{***********************************************}

Procedure PlotGraph (title: title_type;
  id: id_type;
  x_label, y_label : label_type;
  x_factor,
  y_factor: factor_type;
  x_values: x_array_type;
  y_values: y_array_type;
  psi_values: y_array_type;
  psi_initial: real;
  no_values:integer;
  x_min,
  x_max,
  x_interval: x_type;
  y_min,
  y_max,
  y_interval: y_type;
  distress_or_psi: graph_type
);

{>>> constants declaration <<<}
const
  line_per_scr  = 25;
  char_per_line = 80;
  window_border_space= 10;
  window_width_space= 60;
  blank   = ' ';
  pixels_per_line = 8;
  pixels_per_char = 8;
  x_ndx_width = 4;
  y_ndx_width = 4;
  y_axis_lines = 8;

{>>> variables declaration <<<}
var
  x_value,
  y_value,
  indention : integer;
  response    : char;
i, j : integer;
y_top, y_bottom,
x_right, x_left,
no_x_intervals,
no_y_intervals,
spaces_per_interval,
lines_per_interval: integer;

begin
    {--- clear the screen first ---}
    ClrScr;

    {--- set the screen mode ---}
    {high resolution 640 x 200 with black background and one
     color}
    HiRes;

    {--- set x and y boundaries for the graph window ---}
    y_top := 8 * pixels_per_line;
    y_bottom := 17 * pixels_per_line;
    x_left := window_border_space * pixels_per_char;
    x_right := (window_width_space + window_border_space) *
                pixels_per_char;

    {--- output id on the left top corner ---}
    Writeln;
    Write (' ', id);

    {--- output centered title to screen ---}
    indentation := (char_per_line - length (title)) div 2
                   - length (id);
    for i := 1 to indentation do
        Write (blank);
    Writeln (title);

    {--- output y axis lable to screen ---}
    for i := 1 to 3 do Writeln;
    Writeln (' ', y_label);
    Writeln (' (' , y_factor, ',')');
    Writeln;

    {--- output y axis values ---}
    no_y_intervals := trunc ((y_max - y_min) / y_interval);
    if no_y_intervals <= y_axis_lines then
        begin
            lines_per_interval := y_axis_lines div no_y_intervals;
            for i := 1 to (y_axis_lines mod no_y_intervals) do
                writeln;
            for i := no_y_intervals downto 1 do
                begin
                    for j := 1 to (window_border_space - yndx_width - 2) do
                        write (blank);
                    write ((y_interval * i + y_min):yndx_width:2);
for j:= 1 to lines_per_interval do writeln;
end;
for j:= 1 to (window_border_space-y_ndx_width-2) do
  write (blank);
  writeln (y_min:y_ndx_width:2);
end;

(-- output x axis values --)
writeln;
for i:= 1 to (window_border_space-x_ndx_width) do
  write (blank);
no_x_intervals:= (x_max - x_min) div x_interval;
write (x_min:x_ndx_width);
nx:='n';
spaces_per_interval:= window_width_space
  div no_x_intervals;
if spaces_per_interval > x_ndx_width then
begin
  for i:= 1 to (spaces_per_interval-x_ndx_width) do
    write (blank);
  for i:= 1 to no_x_intervals do
    begin
      write ((x_interval*i+x_min):x_ndx_width);
      for j:= 1 to (spaces_per_interval-x_ndx_width) do
        write (blank);
      end;
    writeln;
end;

(-- output x axis label to screen --)
writeln;
indentation:= (char_per_line - length (x_label)) div 2;
for i:= 1 to indentation do
  write (blank);
writeln (x_label);
for i:= 1 to indentation do
  write (blank);
writeln ('(', x_factor, ')');

(-- output prompt to ask user to return to main menu --)
writeln;
indentation:= (char_per_line - 40) div 2;
for i:= 1 to indentation do
  write (blank);
write ('< Press RETURN to return to menu ... >');

(-- draw the box for the screen output --)
DrawBox (1,1,639,199);

draw (1,3*pixels_per_line,639,3*pixels_per_line,1);
draw (1,22*pixels_per_line,639,22*pixels_per_line,1);

(-- define graph output window --)
GraphWindow (x_left, y_top, x_right, y_bottom);
(-- draw y axis --)
Draw (1,1, 1,(y_bottom-y_top-1),1);

(-- draw x axis --)
Draw (1,(y_bottom-y_top-1), (x_right-x_left-1),
     (y_bottom-y_top-1),1);

(-- plot points --)
for i:= 1 to no_values do
begin
   x_value:= x_values[i] * (spaces_per_interval * 
pixels_per_char) div x_interval;
   case distress or psi of
      d: y_value:= trunc ((y_values[i] *
                          (lines_per_interval
                           * pixels_per_line) / y_interval));
      p: y_value:= trunc ((psi_values[i] *
                           (lines_per_interval
                           * pixels_per_line) / y_interval));
   end;
   if y_value = 0
      then y_value:= (y_bottom - y_top)-2
      else y_value:= (y_bottom - y_top) - y_value;
   PlotPoint (x_value, y_value,1);
end;

(-- fit curve to points --)
FitCurve (x_values, y_values, no_values, x_min, x_max,
          x_interval, y_min, y_max, y_interval,
          psi_initial, distress_or_psi);

(-- wait for user to respond to prompt --)
Read (response);

(-- return to text mode and clear screen --)
TextMode;
ClrScr;
end;

(>>> ending of procedure PlotGraph <<<)
Section 3:  Building Model File
GENERAL NARRATIVE

This option allows the user to build a model file which may be used for roadway analysis. The model file contains various information collected from many system database files. This includes location, distress, layer, environmental, traffic and deflection data. Because of the amount of data involved, processing all this information is time-consuming. To allow the flexibility of building the model file section by section, this option provides an alternative to process the above-mentioned data groups individually. However, it is advisory to use the automatic process in most cases to avoid unnecessary confusion.

MODEL.PRG is the driver program for building the model file. The program specifications for the programs called by MODEL.PRG follow the MODEL.PRG program listing. Unless otherwise indicated in a program specification, the DBF files are stored in the subdirectory \PAVEDB\FILES, the NDX files are stored in the subdirectory \PAVEDB\INDEXES, and the DAT files are stored in the subdirectory \PAVEDB\APPLICATION\MODEL. The programs are stored in the subdirectory \PAVEDB\APPLICATION\MODEL unless indicated otherwise in a program specification.
Program Flow for Building Model File Automatically by Choosing Option 1 of Mode.Prg Menu.

- location.dbf
- distress.dbf
- layer.dbf
- weather.dbf
- env.dbf
- traffic.dbf
- subgrade.dbf
- geosho.dbf
- surface.dbf
- disttemp.dbf
- dynaflld.dbf
- fallwght.dbf

FIGURE 26
Program Flow for Building Model File by Individual Data Files by Choosing Option 2 of Model.Frg Menu

- modl_loc.prg
- modl_dis.prg
- modl_lay.prg
- modl_sub.prg
- modl_sho.prg
- modl_suf.prg
- modl_wea.prg
- modl_env.prg
- modl_tmp.prg
- modl_trf.prg
- modl_dyn.prg
- modl_fal.prg

FIGURE 27
Building Model File

Program Specification 1

Program Name: MODEL.PRG

Purpose: To set up a menu for the option to build a model file. This program will ask the user to choose between two menu options: 1) To build the model file automatically, or 2) To build the model file by Individual data files.

Edit/Procedure Information:

1. The automatic option should be the one taken under most conditions. It will extract all the necessary data for the model file from various database files in the system.
2. The individual data files option should be taken with caution. It allows the user to extract information for the model file individually.

Input File(s):

1. Location database file (LOCATION.DBF) and index file (LOCATION.NDX).
2. Distress database file (DISTRESS.DBF) and index file (DISTRESS.NDX).
3. Layer database file (LAYER.DBF) and index file (LAYER.NDX).
4. Weather database file (WEATHER.DBF) and index file (WEATHER.NDX).
5. Environment database file (ENV.DBF) and index file (ENV.NDX).
6. Traffic database file (TRAFFIC.DBF) and index file (TRAFFIC.NDX).
7. Subgrade database file (SUBGRADE.DBF) and index file (SUBGRADE.NDX).
8. Shoulder database file (GEOSHO.DBF) and index file (GEOSHO.NDX).
9. Surface database file (SURFACE.DBF) and index file (SURFNDX.NDX).

10. District temperature database file (DISTTEMP.DBF) and index file (DISTTEMP.NDX).

11. Dynaflec database file (DYNAPLLD.DBF) and index file (DYNAPLLD.NDX).

12. Falling weight database file (FALLWGT.DBF) and index file (FALLWGT.NDX).

Intermediate and Input/Output File(s):

1. Temporary distress database file (TEMP_DIS.DBF).

2. Temporary layer database file (TEMP_LAY.DBF).

3. Intermediate distress data file (DISTRESS.DAT).


5. Intermediate base thickness data file (BASETHCK.DAT).

Output File(s):

1. Model database file (MODEL.DBF).

dBASE Program Called:

MODELING.PRG
MODL_LOC.PRG
MODL_DIS.PRG
MODL_LAY.PRG
MODL_WEA.PRG
MODL_ENV.PRG
MODL_TRF.PRG
MODL_SUB.PRG
MODL_SHO.PRG
MODL_SUF.PRG
MODL_TEMP.PRG
MODL_DYN.PRG
MODL_FAL.PRG
Building Model File

Program Specification 1
Model Menu Screen

TEXAS FLEXIBLE PAVEMENT DATABASE
BUILDING MODEL FILE

1) Build Model File Automatically
2) Build Model File by Individual Data Files

Option ——> _
Building Model File

Program Specification 1
Intermediate and Input/Output File Layout 1

File Name: TEMP_DIS.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Field Name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>SID_NO</td>
<td>Numeric</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>PAVETYP</td>
<td>Numeric</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>ALLG_RHO</td>
<td>Numeric</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>ALLG_BETA</td>
<td>Numeric</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>RUIT_RHO</td>
<td>Numeric</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>RUIT_BETA</td>
<td>Numeric</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>PSI_RHO</td>
<td>Numeric</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>PSI_BETA</td>
<td>Numeric</td>
<td>7</td>
<td>4</td>
</tr>
</tbody>
</table>
## Building Model File

Program Specification 1
Intermediate and Input/Output File Layout 2

**File Name:** TEMP_LAY.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Field Name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SID NO</td>
<td>Numeric</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>BASETYPE</td>
<td>Numeric</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>BASETHCK</td>
<td>Numeric</td>
<td>7</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
Building Model File

Program Specification 1
Intermediate and Input/Output File Layout 3

**File Name:** DISTRESS.DAT

<table>
<thead>
<tr>
<th>Columns</th>
<th>Size/Type</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 4</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>5 - 6</td>
<td>2N</td>
<td>Year</td>
</tr>
<tr>
<td>7 - 17</td>
<td>11N</td>
<td>Annual Cumulative 18KIP Equivalent Axle Load</td>
</tr>
<tr>
<td>18</td>
<td>1N</td>
<td>Patching Area Distress Value</td>
</tr>
<tr>
<td>19</td>
<td>1N</td>
<td>Rutting Area Distress Value</td>
</tr>
<tr>
<td>20</td>
<td>1N</td>
<td>Alligator Cracking Area Distress Value</td>
</tr>
<tr>
<td>21 - 27</td>
<td>7.5N</td>
<td>Serviceability Index Mean Value</td>
</tr>
</tbody>
</table>
### Building Model File

**Program Specification 1**  
Intermediate and Input/Output File Layout 4

**File Name:** RHOBETA.DAT

<table>
<thead>
<tr>
<th>Columns</th>
<th>Size/Type</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 2</td>
<td>2N</td>
<td>Pave Type</td>
</tr>
<tr>
<td>3 - 9</td>
<td>7.4N</td>
<td>Rho Value for Alligator Cracking Area Distress Curve</td>
</tr>
<tr>
<td>10 - 16</td>
<td>7.4N</td>
<td>Beta Value for Alligator Cracking Area Distress Curve</td>
</tr>
<tr>
<td>17 - 23</td>
<td>7.4N</td>
<td>Rho Value for Rutting Area Distress Curve</td>
</tr>
<tr>
<td>24 - 30</td>
<td>7.4N</td>
<td>Beta Value for Rutting Area Distress Curve</td>
</tr>
<tr>
<td>31 - 37</td>
<td>7.4N</td>
<td>Rho Value for PSI Curve</td>
</tr>
<tr>
<td>38 - 44</td>
<td>7.4N</td>
<td>Beta Value for PSI Curve</td>
</tr>
</tbody>
</table>
Building Model File

Program Specification 1
Intermediate and Input/Output File Layout 5

File Name: BASETHCK.DAT

<table>
<thead>
<tr>
<th>Columns</th>
<th>Size/Type</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 4</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>5 - 6</td>
<td>2N</td>
<td>Structure Number</td>
</tr>
<tr>
<td>7 - 8</td>
<td>2N</td>
<td>Layer Number</td>
</tr>
<tr>
<td>9 - 10</td>
<td>2N</td>
<td>Layer Description</td>
</tr>
<tr>
<td>11 - 12</td>
<td>2N</td>
<td>Layer Material</td>
</tr>
<tr>
<td>13 - 17</td>
<td>5.2N</td>
<td>Center Thickness of Layer</td>
</tr>
</tbody>
</table>
Building Model File

Program Specification 1
Program Listing

* File name: MODEL.PRG
* Program name: model
* Project 2456: Texas Flexible Pavement Database
* TAMU/TTI
* Written by: Victor Wong
* Created on: July 14, 1988
* Last updated: July 26, 1988
* Purpose: To call on modeling programs.

* initialize variables
PRIVATE OPTION
PRIVATE REP
REP = .T.

* repeat option menu if escape key is not pressed
DO WHILE REP
  OPTION = ""
  * redisplay menu unitl invalid option or escape is entered
  DO WHILE .NOT. (OPTION $ "12") .AND. REP
    * option menu
    CLEAR
    @ 6,24 SAY "TEKSAS FLEXIBLE PAVEMENT DATABASE"
    @ 7,24 SAY " BUILDING MODEL FILE"
    @ 11,15 SAY "1) Build Model File Automatically"
    @ 12,15 SAY "2) Build Model File by Individual Data Files"
    @ 14,42 SAY "Option =-----" GET OPTION PICTURE "X"
    @ 3,9 TO 18,70 DOUBLE
  
  * read option
  READ

  * set repeat flag to false if escape key is pressed
  IF READKEY() = 12
    REP = .F.
  ENDF

  * warning for invalid option entry
  IF .NOT. (OPTION $ "12") .AND. REP
    @ 20,10 SAY "Enter only 1 or 2."
    WAIT
    @ 20,10 SAY "
  ENDF
ENDDO

* carry out the option
IF REP
DO CASE
   CASE OPTION = "1"
      ANS = "N"
      DO WHILE .NOT. (ANS $ "yynn")
         CLEAR
         ANS = "N"
         @10,20 SAY "This process will take more than 2 hours."
         @11,20 SAY "Do you still want this option? (Y/N) ";
         GET ANS PICTURE "X"
         READ
         ENDDO
      IF (ANS $ "yn")
         DO \pavedb\applicat\model\MODL_LOC
         DO \pavedb\applicat\model\MODL_DIS
         DO \pavedb\applicat\model\MODL_LAY
         DO \pavedb\applicat\model\MODL_WEA
         DO \pavedb\applicat\model\MODL_ENV
         DO \pavedb\applicat\model\MODL_TRF
         DO \pavedb\applicat\model\MODL_SUB
         DO \pavedb\applicat\model\MODL_SHO
         DO \pavedb\applicat\model\MODL_SUF
         DO \pavedb\applicat\model\MODL_TMP
         DO \pavedb\applicat\model\MODL_DYN
         DO \pavedb\applicat\model\MODL_FAL
      ENDF IF
   CASE OPTION = "2"
   END CASE
END DO
RETURN
Building Model File
Program Specification 2

Program Name: MODELING.PRG

Purpose: To set up a menu for building the model file by extracting information from individual group of data.

Edit/Procedure Information:

1. This program calls the appropriate subprograms to extract information from various system database files.

dBASE Programs Called:

MODL LOC.PRG
MODL DIS.PRG
MODL LAY.PRG
MODL WEA.PRG
MODL ENV.PRG
MODL TRF.PRG
MODL SUB.PRG
MODL SHO.PRG
MODL SUP.PRG
MODL TMP.PRG
MODL DYN.PRG
MODL FAL.PRG
Building Model File

Program Specification 2
Model Menu Screen

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>Retrieve Location Data</td>
</tr>
<tr>
<td>2)</td>
<td>Retrieve Rho and Beta Values for</td>
</tr>
<tr>
<td></td>
<td>Alligator Carcking, Rutting and PSI</td>
</tr>
<tr>
<td>3)</td>
<td>Retrieve Layer Data</td>
</tr>
<tr>
<td>4)</td>
<td>Retrieve Environmental Data</td>
</tr>
<tr>
<td>5)</td>
<td>Retrieve Traffic Data</td>
</tr>
<tr>
<td>6)</td>
<td>Retrieve Surface Deflection Data</td>
</tr>
</tbody>
</table>

Option ➞ _
* File name: MODELING.PRG
* Program name: modeling
* Project 2456: Texas Flexible Pavement Database
* TAMU/TTI
* Written by: Victor Wong
* Created on: July 14, 1988
* Last updated: July 26, 1988
* Purpose: To call on modeling programs individually.

* initialize variables
PRIVATE OPTION
PRIVATE REP
REP = .T.
* repeat option menu if escape key is not pressed
DO WHILE REP
  OPTION = " "
  * re-display menu until valid option or escape is pressed
  DO WHILE .NOT. (OPTION $ "123456") .AND. REP
    CLEAR
    @ 5,24 SAY "TEXAS FLEXIBLE PAVEMENT DATABASE"
    @ 6,20 SAY "BUILD MODEL FILE BY INDIVIDUAL DATA FILES"
    @ 9,20 SAY "1) Retrieve Location Data"
    @ 10,20 SAY "2) Retrieve Rho and Beta Values for"
    @ 11,20 SAY " Alligator Cracking, Rutting and PSI"
    @ 12,20 SAY "3) Retrieve Layer Data"
    @ 13,20 SAY "4) Retrieve Environmental Data"
    @ 14,20 SAY "5) Retrieve Traffic Data"
    @ 15,20 SAY "6) Retrieve Surface Deflection Data"
    @ 17,42 SAY "Option =-choice) GET OPTION PICTURE "X"
    @ 3,9 TO 20,70 DOUBLE

  * read option
  READ

  * set repeat flag to false if escape key is pressed
  IF READKEY() = 12
    REP = .F.
  ENDF

  * warning for invalid option entry
  IF .NOT. (OPTION $ "123456") .AND. REP
    @20,10 SAY "Enter only 1 through 6."
    WAIT
    @20,10 SAY ""
  ENDF
ENDDO
* carry out the option
IF REP
   DO CASE
      CASE OPTION = "1"
      CLEAR
      @10,15 SAY "This option will delete the most recent model file,"
      @11,15 SAY "and rebuild the model file."
      ANS = "N"
      @12,15 SAY "Do you still want to choose this option? (Y/N) ";
      GET ANS PICTURE "X"
      READ
      IF (ANS="Y") .OR. (ANS="y")
      CLEAR
      @10,18 SAY "Retrieving Location Data for Model File..."
      DO \pavedb\applicat\model\MODL_LOC
      ENDIF
      CASE OPTION = "2"
      CLEAR
      @10,15 SAY "Retrieving Rho and Beta Values for Model File..."
      DO \pavedb\applicat\model\MODL_DIS
      CASE OPTION = "3"
      CLEAR
      @10,20 SAY "Retrieving Layer Data for Model File..."
      DO \pavedb\applicat\model\MODL_LAY
      DO \pavedb\applicat\model\MODL_SUB
      DO \pavedb\applicat\model\MODL_SHO
      DO \pavedb\applicat\model\MODL_SUF
      CASE OPTION = "4"
      CLEAR
      @10,15 SAY "Retrieving Environmental Data for Model File..."
      DO \pavedb\applicat\model\MODL_WEA
      DO \pavedb\applicat\model\MODL_ENV
      DO \pavedb\applicat\model\MODL_TMP
      CASE OPTION = "5"
      CLEAR
      @10,20 SAY "Retrieving Traffic Data for Model File..."
      DO \pavedb\applicat\model\MODL_TRF
      CASE OPTION = "6"
      CLEAR
      @10,14 SAY "Retrieving Surface Deflection Data for Model File..."
      DO \pavedb\applicat\model\MODL_DYN
      DO \pavedb\applicat\model\MODL_FAL
   ENDCASE
   ENDF
ENDDO
RETURN
Building Model File
Program Specification 3

Program Name: MODL_LOC.PRG

Purpose: To find the location data for the model file.

Edit/Procedure Information:

1. The program takes each active section identification number in the location database and inspects its most recent widening flag from the geometric shoulder database. It ignores the section if the program cannot find the section in the geometric and shoulder database file or the widening flag value is two.

2. If the section should not be ignored, copy the section identification number, highway prefix, highway district and county number to the model file.

3. When this program is run, it destroys the old contents in the model file (MODEL.DBF).

Input File(s):

1. Location database file (LOCATION.DBF) with index file (LOCATION.NDX).

2. Geometric shoulder database file (GEOSHO.DBF) with index file (GEONDX.NDX).

Output File(s):

1. Model database file (MODEL.DBF).

File Layout(s):

1. LOCATION.DBF (See Appendix A.)

2. GEOSHO.DBF (See Appendix A.)

3. MODEL.DBF (See Appendix A.)
Program Flow for Modl_Loc.Prg

FIGURE 28
Building Model File

Program Specification 3
Program Listing

* File name: MODL_LOC.PRG
* Program name: modl_loc
* Project 2456: Texas Flexible Pavement Database
* TAMU/TTI
* Written by: Victor Wong
* Created on: June 29, 1988
* Last updated: July 14, 1988
* Purpose: To find location data for the model file.

* open files
SELECT A
USE \pavedb\files\LOCATION INDEX \pavedb\indexes\LOCATION
SELECT B
USE \pavedb\files\MODEL
ZAP
SELECT C
USE \pavedb\files\GEOSHQ INDEX \pavedb\indexes\GEONDX

* for each active sid number in the location file
SELECT A
DO WHILE .NOT. EOF()
   * see if the most recent widening flag of this section, if it exists
   * in the database; ignore if it is 2
   SELECT C
   SEEK STR(A->SID_NO, 4)
   IF FOUND()
      DO WHILE SID_NO = A->SID_NO
         SKIP
      ENDOO
      SKIP -1
      IF WIDENFIG = 2
         IGNORE = .T.
      ELSE
         IGNORE = .F.
      ENDFI
      ELSE
         IGNORE = .T.
      ENDFI
   * continue if this section should not be ignored
   IF .NOT. IGNORE
      SELECT B
      APPEND BLANK

      * copy the sid number
      REPLACE SID_NO WITH A->SID_NO

      * copy the highway type
REPLACE HWYTYPE WITH A->HWYPREFIX

* copy the district number
REPLACE HWYDIST WITH A->HWYDIST

* copy the county number
REPLACE CNTYNUM WITH A->CNTYNUM
ENDIF

* skip to the next sid number in location file
SELECT A
SKIP
ENDDO

* close files
CLOSE ALL
Building Model File

Program Specification 4

Program Name: MODL_DIS.PRG

Purpose: To find the distress data for the model file.

Edit/Procedure Information:

1. For each section identification number in the model file, this program will try to find the data in the distress database file. If there is such section in the distress file, the SID number, year, annual cumulative 18KIP equivalent axle load value, area of distress value for patching, rutting, and alligator cracking, serviceability index mean value, and the pavement type are written to an intermediate file (DISTRESS.DAT).

2. With the intermediate file, the program will call an external Turbo Pascal program (MODL_RB.PAS) to find the rho and beta values of the distress and psi curves. The Pascal program returns the values through another intermediate file (RHOBETA.DAT).

3. MODL_DIS.PRG then reads the rho and beta values from RHOBETA.DAT into an intermediate database file (TEMP_DIS.DBF). The values are in turn copied into the model file.

Input File(s):

1. Distress database file (DISTRESS.DBF) with index file (DISTRESS.NDX).

Intermediate and Input/Output File(s):

1. Temporary distress database file (TEMP_DIS.DBF).
2. Intermediate distress data file (DISTRESS.DAT).
3. Intermediate rho/beta value data file (RHOBETA.DAT).

Output File(s):

1. Model database file (MODEL.DBF).
Program Flow for Modl_Dis.Prg

FIGURE 29
Building Model File

Program Specification 4
Program Listing

* File name: MODL_DIS.PRG
* Program name: modl_dis
* Project 2456: Texas Flexible Pavement Database
* TAMU/TTI
* Written by: Victor Wong
* Created on: June 29, 1988
* Last updated: July 26, 1988
* Purpose: To find the rho and beta values for rutting,
* alligator cracking and psi.

* open files
SELECT A
USE \pavedb\files\MODEL
SELECT B
USE \pavedb\files\DISTRESS INDEX \pavedb\indexes\DISTRESS
SELECT C
USE \pavedb\files\TEMP_DIS
ZAP

* for each sid number in the model file
SELECT A
DO WHILE .NOT. EOF()
  * find the data in the distress database file
  SELECT B
  SEEK STR(A->SID_NO,4)

  * if it is found
  IF FOUND()
    * copy the data to a column arranged file
    COPY TO \pavedb\applicat\model\DISTRESS.DAT ;
      FIELDS SID NO,YEAR,A18KEAL,PATCH, RUTT, ALLGCR,;
      SIMEAN,PAVETYP WHILE SID_NO=A->SID_NO TYPE SDF

      * run the Turbo Pascal program to find the rho and beta
      RUN \pavedb\applicat\model\MODL_R_B

    * copy data to model file
    SELECT C
    APPEND FROM \pavedb\applicat\model\RHOBETA.DAT TYPE SDF
    SELECT A
    REPLACE PAVETYP WITH C->PAVETYP
    REPLACE ALLG_RHO WITH C->ALLG_RHO
    REPLACE ALLG_BETA WITH C->ALLG_BETA
    REPLACE RUTT_RHO WITH C->RUTT_RHO
    REPLACE RUTT_BETA WITH C->RUTT_BETA
    REPLACE PSI_RHO WITH C->PSI_RHO
    REPLACE PSI_BETA WITH C->PSI_BETA

534
ENDIF

* skip to the next sid number in model file
SELECT A
SKIP
ENDDO

* close files
CLOSE ALL
Building Model File
Program Specification 4
Program Listing

(* File name: MODL_R_B.PAS
* Program name: model_rho_beta
* Project 2456: Texas Flexible Pavement Database Conversion
* TAMU/TTI
* Written by: Victor Wong
* Created on: June 29, 1988
* Last updated: July 14, 1988
* Purpose: To find the rho's and beta's of alligator
* cracking, rutting and PSI for the model file.
* Input file: DISTRESS.DAT
* Output file: RHOBETA.DAT *)

{>>> PROGRAM MODL_R_B <<<}
program model_rho_beta;

{>>> constant declaration <<<}
const
  blank = ' ';
  psi_initial = 4.5;
  psi_final = 1.5;

{>>> type declaration <<<}
{$I \pavedb\applic\graph18k\declare}

{>>> variable declaration <<<}
var
  {--- variables for data processing ---}
  data_file:   text;

  {input data format}
  sid_no: string[4];  {sid number  4N}
  year: string[2];   {year  2N}
  al8keal: string[11]; {18kip  11N}
  patching: string[1];  {patching  1N}
  rutting: string[1];   {rutting  1N}
  allgcrlk: string[1];  {alligator crack 1N}
  simean: string[7];   {simean  7.5N}
  pavetyp: string[2];  {pavement type  2N}

  {variables needed for conversion of data from strings into }
  {numbers and variables needed for any numeric manipulation}
  {with the data}
  err_code,
  err_code1,
  err_code2: integer;  {error code for string to}
  sid_no_value,    {number conversion}
  (sid no value)
psi_value: real; {psi value}
patch_value, {patching value}
rutting_value, {rutting value}
allgcrk_value: integer; {alligator crack value}
patch_area, {patching area}
rutting_area, {rutting area}
allgcrk_area: real; {alligator crack area}
a18k1, a18k2, {if a18keal = aa bbb ccc ddd}
a18k3, a18k4, {then a18k1=aa,18k2=bbb,}
a18k1_carry, {81k3=ccc,18k4=ddd}
a18k2_carry,
a18k3_carry, {carry from the digit group}
a18k4_carry, {for addition}
accum_18k1,
accum_18k2,
accum_18k3, {accumulated 18k value by}
accum_18k4: integer; {digit group}
a18k1_zero,
a18k2_zero,
a18k3_zero,
a18k4_zero: boolean; {flag saying if the digit}
0 group = 0
a18keal1: string[2]; {portions of the a18keal str}
a18keal2,
a18keal3,
a18keal4:

:variables needed for finding rho's and beta's
allg_rho, allg_beta,
rutt_rho, runt_beta,
psi_rho, psi_beta: real;
{psi value}
psi,
allg_y,
rutt_y,
psi_y: y_array_type; {y value (allgcrk/rutt value)}
allg_x,
rutt_x,
psi_x,
allg_x1,
allg_x2,
allg_x3,
allg_x4,
rutt_x1,
rutt_x2,
rutt_x3,
rutt_x4,
psi_x1,
psi_x2,
psi_x3,
psi_x4: x_array_type; {x value by digit group}

(-- other variables in the program --)
c: char;
i: integer;
char_at_index: char;
index: integer;
found: boolean;
allg_count, rutt_count,
psi_count: integer;
outfile: text;

(*> include files <*)
($I \pavedb\applicat\graph18k\rhobeta)
{file with procedure to calculate}
{rho and beta}

(*> beginning of program <*)
begin

(*--- open the distress data file ---*)
assign (data_file, '"pavedb\applicat\model\distress.dat');
reset (data_file);

(*--- initialize some variables---*)
accum_18k1:= 0;
accum_18k2:= 0;
accum_18k3:= 0;
accum_18k4:= 0;

readln (data_file, sid_no, year, a18keal, patching, rutting,
       allgcrk, simean, pavetype);
                allg_count:= 0; {initialize the count of valid data to 0}
                rutt_count:= 0; {initialize the count of valid data to 0}
                psi_count:= 0; {initialize the count of valid data to 0}
                 patch_area:= 0.005;{initialize the patch area to 0.0}

{while not eof}
{read data, accumulate 18kip value }
{and store valid distress or psi value}
while not eof(data_file) do
  begin
    {convert the a18keal string value to 3 integer values)
    {so that the concatenation of a18k1, a18k2 and a18k3 }
    {will give the whole 18keal value }
    {if the err_code is non-zero, it means that digits }
    {are missing at those positions and it is the same as }
    {having zeros there }
    {the positions of leading zeros are indicated by the }
    {a18kX_zero flag }

    {split a18keal value string into 4 sections }
    (((xx,xxx,xxx,xxx))
     a18keal1:= copy(a18keal,1,2);
     a18keal2:= copy(a18keal,3,3);
     a18keal3:= copy(a18keal,6,3);
     a18keal4:= copy(a18keal,9,3);

538
(delete leading blank spaces of a18kealx string)
(and store convert the string to integer a18k0x values)
index:= 1;
char_at_index:= copy (a18keal1,index,1);
while (char_at_index = blank) and (index < 2) do
begin
  index:= index + 1;
  char_at_index:= copy (a18keal1,index,1);
end;
if index < 2 then
  val (copy(a18keal1,index,3-index),a18k1,err_code)
else if char_at_index <> blank then
  val (copy (a18keal1,index,3-index),a18k1,err_code)
else
begin
  a18k1:= 0;
  a18k1_zero:= true;
end;

index:= 1;
char_at_index:= copy (a18keal2,index,1);
while (char_at_index = blank) and (index < 3) do
begin
  index:= index + 1;
  char_at_index:= copy (a18keal2,index,1);
end;
if index < 3 then
  val (copy(a18keal2,index,4-index),a18k2,err_code)
else if char_at_index <> blank then
  val (copy (a18keal2,index,4-index),a18k2,err_code)
else
begin
  a18k2:= 0;
  a18k2_zero:= true;
end;

index:= 1;
char_at_index:= copy (a18keal3,index,1);
while (char_at_index = blank) and (index < 3) do
begin
  index:= index + 1;
  char_at_index:= copy (a18keal3,index,1);
end;
if index < 3 then
  val (copy(a18keal3,index,4-index),a18k3,err_code)
else if char_at_index <> blank then
  val (copy (a18keal3,index,4-index),a18k3,err_code)
else
begin
  a18k2:= 0;
  a18k3_zero:= true;
end;

index:= 1;
char_at_index := copy (a18keal4, index, 1);
while (char_at_index = blank) and (index < 3) do
begin
   index := index + 1;
   char_at_index := copy (a18keal4, index, 1);
end;
if index < 3 then
   val (copy(a18keal4, index, 4-index), a18k4, err_code)
else if char_at_index <> blank then
   val (copy (a18keal4, index, 4-index), a18k4, err_code)
else
   begin
      a18k4 := 0;
      a18k4_zero := true;
   end;

{if the a18keal value is not zero continue processing)
if not (a18k1_zero and a18k2_zero and a18k3_zero and
      a18k4_zero) then
begin
   {increment the accumulated 18kmp value
   accum_18k4 := accum_18k4 + a18k4;
   a18k4_carry := accum_18k4 div 1000;
   accum_18k4 := accum_18k4 mod 1000;
   accum_18k3 := accum_18k3 + a18k3 + a18k4_carry;
   a18k3_carry := accum_18k3 div 1000;
   accum_18k3 := accum_18k3 mod 1000;
   accum_18k2 := accum_18k2 + a18k2 + a18k3_carry;
   a18k2_carry := accum_18k2 div 1000;
   accum_18k2 := accum_18k2 mod 1000;
   accum_18k1 := accum_18k1 + a18k1 + a18k2_carry;

   {--- for each distress type --- }
   {--- do the necessary processing --- }
   {alligator cracking}
   {convert alligator crack data from string to
   (integer)}
   val (allgcrcr, allgcrcr_value, err_code1);

   {convert patching data from string to integer}
   val (patching, patch_value, err_code2);

   {if the conversion is successful}
   if (err_code1 = 0) and (err_code2 = 0) then
      begin
         {check if the value is valid i.e. <> 9)
         if (allgcrcr_value <> 9) then
            begin
               {increment valid data count}
               allg_count := allg_count + 1;
            end;

            {determine the percentage of alligator}
            {cracking area}
case allgcrk_value of
  0: allgcrc_area := 0.005;
  1: allgcrc_area := 0.055;
  2: allgcrc_area := 0.305;
  3: allgcrc_area := 0.50;
end;

(determine the percentage of patching)
(area)
if patch_value <> 9 then
  case patch_value of
  0: patch_area := 0.005;
  1: patch_area := 0.055;
  2: patch_area := 0.305;
  3: patch_area := 0.50;
end;

{take the sum of the two area %}
{as the value to be plotted}
allg_y [allg_count] := (allgcrc_area +
  patch_area);
if allg_y [allg_count] >= 1.0 then
  allg_y [allg_count] := 0.985;

{store the new accumulated 18k1p value}
{in x arrays}
allg_x1 [allg_count] := accum_18k1;
allg_x2 [allg_count] := accum_18k2;
allg_x3 [allg_count] := accum_18k3;
allg_x4 [allg_count] := accum_18k4;
end
else
begin
if err_code1 <> 0 then
begin
  gotoxy (22,14);
  write ('***WARNING***');
  write (' bad data encountered!');
  gotoxy (22,15);
  write ('   ');
  write ('INVALID CODE FOR ALLIGATOR ');
  write ('CRACKING ');
  gotoxy (22,17);
  write ('Still processing.');
  write ('  Please wait ...');
end;
if err_code2 <> 0 then
begin
  gotoxy (22,14);
  write ('***WARNING***');
  write (' bad data encountered!');
  gotoxy (22,15);
  write ('   ');
  write ('INVALID CODE FOR PATCHING');
write ('');
gotoxy (22,17);
write ('Still processing.');
write (' Please wait ...');
end;
end;
end;

{rutting}
{convert rutting data from string to integer}
val (rutting, rutting_value, err_code1);

{convert patching data from string to integer}
val (patching, patch_value, err_code2);

{if the conversion is successful}
if (err_code1 = 0) and (err_code2 = 0) then
begin
{check if the value is valid i.e. <> 9}
if (rutting_value <> 9) then
begin
{increment valid data count}
rutt_count := rutt_count + 1;

{determine the percentage of rutting}
{area}
case rutting_value of
0: rutting_area := 0.005;
1: rutting_area := 0.13;
2: rutting_area := 0.38;
3: rutting_area := 0.50;
end;

{determine the percentage of patching}
{area}
if patch_value <> 9 then
begin
case patch_value of
0: patch_area := 0.005;
1: patch_area := 0.055;
2: patch_area := 0.305;
3: patch_area := 0.5;
end;
end;

{take the sum of the two area %}
{as the value to be plotted}
rutt_y[rutt_count] := (rutting_area +
patch_area);
if rutt_y[rutt_count] >= 1.0 then
rutt_y[rutt_count] := 0.985;

{store the new accumulated 18kips value}
{in x arrays}
rutt_x1[rutt_count] := accum_18k1;
rutt_x2[rutt_count] := accum_18k2;
rutti_x3[rutti_count]:= accum_18k3;
rutti_x4[rutti_count]:= accum_18k4;
end
else
begin
  if err_code1 <> 0 then
  begin
    gotoxy (22,14);
    write ('***WARNING*** ');
    write ('bad data encountered');
    gotoxy (22,15);
    write (' ');
    write ('INVALID CODE FOR RUTTING');
    gotoxy (22,17);
    write ('Still processing.');
    write ('Please wait ...');
  end;
  if err_code2 <> 0 then
  begin
    gotoxy (22,14);
    write ('***WARNING*** ');
    write ('bad data encountered');
    gotoxy (22,15);
    write (' ');
    write ('INVALID CODE FOR PATCHING');
    gotoxy (22,17);
    write ('Still processing.');
    write ('Please wait ...');
  end;
end;

{psi}
{convert psi data from string to real}
val (smean,psi_value,err_code);

{if the conversion is successful}
if (err_code = 0) then
begin
  if psi_value <> 9.99999 then
  begin
    psi_count:= psi_count + 1;

    {store the psi value in the psi array}
    if psi_value = 0.0 then
      psi[psi_count]:= 0.0005
    else
      psi[psi_count]:= psi_value;

    {transform the psi value to a distress value}
    psi_y[psi_count]:= ((psi_initial -
                         psi[psi_count]) / (psi_initial-
                         psi_final))/10;

  end;
end;
{store the new accumulated 18kip value
(in x arrays)
psi_x1[psi_count]:= accum_18k1;
psi_x2[psi_count]:= accum_18k2;
psi_x3[psi_count]:= accum_18k3;
psi_x4[psi_count]:= accum_18k4;
end
else
begin
if err_code <> 0 then
begin
gotoxy (22,14);
write ('***WARNING*** ');
write ('bad data encountered');
gotoxy (22,15);
write ('');
write ('INVALID PSI VALUE ');
gotoxy (22,17);
write ('Still processing. ');
write ('Please wait ...');
end;
end;
end;
end; {if}

if not eof(data_file) then
  readln(data_file, sid_no, year, a18keal, patching,
         rutting, allgcrc, simean, pavetypr);
end;

{find a suitable accumulated 18kip factor for the x axis}
(on the graph)
{NOTE: 18 kip values are taken as the 3 most significant)
{ digits i.e. a number 1,234,567 will be expressed )
{ as 123 x 10^4
if accum_18k3=0 then
begin
  for i:= 1 to allg_count do allg_x[i]:= allg_x4[i];
  for i:= 1 to rutt_count do rutt_x[i]:= rutt_x4[i];
  for i:= 1 to psi_count do psi_x[i]:= psi_x4[i];
end
else if accum_18k2 = 0 then
begin
  if (a18k3 div 100) > 0 then
begin
    for i:= 1 to allg_count do allg_x[i]:= allg_x3[i];
    for i:= 1 to rutt_count do rutt_x[i]:= rutt_x3[i];
    for i:= 1 to psi_count do psi_x[i]:= psi_x3[i];
  end
  else if (a18k3 div 10) > 0 then
begin
    for i:= 1 to allg_count do
      allg_x[i]:= allg_x3[i]*10 + allg_x4[i] div 100;
    for i:= 1 to rutt_count do

allg_x[i] := rutt_x3[i]*10 + rutt_x4[i] div 100;
for i := 1 to psi_count do
  allg_x[i] := psi_x3[i]*10 + psi_x4[i] div 100;
end
else
begin
for i := 1 to allg_count do
  allg_x[i] := allg_x3[i]*100 + allg_x4[i] div 10;
for i := 1 to rutt_count do
  allg_x[i] := rutt_x3[i]*100 + rutt_x4[i] div 10;
for i := 1 to psi_count do
  allg_x[i] := psi_x3[i]*100 + psi_x4[i] div 10;
end
else if accum_18k1 = 0 then
begin
if (a18k2 div 100) > 0 then
begin
for i := 1 to allg_count do
  allg_x[i] := allg_x2[i];
for i := 1 to rutt_count do
  rutt_x[i] := rutt_x2[i];
for i := 1 to psi_count do
  psi_x[i] := psi_x2[i];
end
else if (a18k2 div 10 ) > 0 then
begin
for i := 1 to allg_count do
  allg_x[i] := allg_x2[i]*10 + allg_x3[i] div 100;
for i := 1 to rutt_count do
  rutt_x[i] := rutt_x2[i]*10 + rutt_x3[i] div 100;
for i := 1 to psi_count do
  psi_x[i] := psi_x2[i]*10 + psi_x3[i] div 100;
end
else
begin
for i := 1 to allg_count do
  allg_x[i] := allg_x2[i]*100 + allg_x3[i] div 10;
for i := 1 to rutt_count do
  rutt_x[i] := rutt_x2[i]*100 + rutt_x3[i] div 10;
for i := 1 to psi_count do
  psi_x[i] := psi_x2[i]*100 + psi_x3[i] div 10;
end
end
else
begin
if (a18k1 div 10) > 0 then
begin
for i := 1 to allg_count do
  allg_x[i] := allg_x1[i]*10 + allg_x2[i] div 100;
for i := 1 to rutt_count do
  rutt_x[i] := rutt_x1[i]*10 + rutt_x2[i] div 100;
for i := 1 to psi_count do
  psi_x[i] := psi_x1[i]*10 + psi_x2[i] div 100;
end
else
begin

end
for i:= 1 to allg_count do
    allg_x[i]:= allg_xl[i]*100 + allg_x2[i] div 10;
for i:= 1 to rutt_count do
    rutt_x[i]:= rutt-xl[i]*100 + rutt-x2[i] div 10;
for i:= 1 to psi_count do
    psi_x[i]:= psi-xl[i]*100 + psi-x2[i] div 10;
end
end;

Find_rho_beta (allg_x, allg_y, allg_count, allg_rho, allg_beta);
Find_rho_beta (rutt_x, rutt_y, rutt_count, rutt_rho, rutt_beta);
Find_rho_beta (psi_x, psi_y, psi_count, psi_rho, psi_beta);

assign (outfile, '"pavedb\applcat\model\rhobeta.dat"');
rewrite (outfile);
close(outfile);
close (data file);

>>> ending of program <<<
end.
Building Model File

Program Specification 5

Program Name:  MODL_IAY.PRG

Purpose:  To find the layer data for the model file.

Edit/Procedure Information:

1. For each section identification number in the model file, the program finds the section in the layer file. If it is found, the data needed to find the the total thickness of the most recent same material base layers is copied to an intermediate file (BASETHCK.DAT).

2. A Turbo Pascal program (BASETHCK.PAS) will take the data in BASETHCK.DAT and find the most recent base layer. Then it will keep adding the thickness of other base layers that have the same material code number. It does so by backtracking through the layers starting with the highest layer number. It stops when it encounter a base layer with a different material than the latest base layer, or the earliest layer is encountered.

3. The total base layer thickness is then passed back to the MODL_IAY.PRG program by writing the result in the intermediate file (BASETHCK.DAT). The type of base material is also passed in the file. The dBASE program (MODL_IAY.PRG) will copy the result to a temporary database file (TEMP_IAY.DBF) and then to the model file.

Input File(s):

1. Layer database file (LAYER.DBF) with index file (LAYER.NDX).

Intermediate and Input/Output File(s):

1. Temporary layer database file (TEMP_IAY.DBF).

2. Intermediate base thickness data file (BASETHCK.DAT).

Output File(s):

1. Model database file (MODEL.DBF).
Building Model File

Program Specification 5

File Layout(s):

1. LAYER.DBF (See Appendix A)

2. TEMP_LAY.DBF (Refer to Intermediate and Input/Output File Layout 2 in Program Specification 1.)

3. BASETHCK.DAT (Refer to Intermediate and Input/Output File Layout 5 in Program Specification 1.)

4. MODEL.DBF (See Appendix A)

Programs Called:

BASETHCK.PAS
Program Flow for Mod1_Lay.Prg

layer.dbf

temp_layer.dbf

modeling
option
3

mod1_lay.prg

basethck.dat

basethck.pas/basethck.com

model.dbf

FIGURE 30
Building Model File

Program Specification 5
Program Listing

* File name: MODL_LAY.PRG
* Program name: modl_lay
* Project 2456: Texas Flexible Pavement Database
* TAMU/TTI
* Written by: Victor Wong
* Created on: June 29, 1988
* Last updated: July 26, 1988
* Purpose: To find the layer data for the model file.

* open files
SELECT A
USE \pavedb\files\MODEL
SELECT B
USE \pavedb\files\LAYER INDEX \pavedb\indexes\LAYNDX
SELECT C
USE \pavedb\files\TEMP_LAY

* fo each sid number in the model file
SELECT A
DO WHILE .NOT. EOF()
   * find the sid number in layer file
   SELECT B
   SEEK STR(A->SID_NO,4)
   * if found
   IF FOUND()
      * copy the data to a column arranged file
      COPY TO \pavedb\applicat\model\BASETHCK.DAT ;
      FIELDS SID NO,STRUCNUM,LAYNUM,LAYDESC,LAYMATCL,CENITHK ;
      WHILE SID_NO=A->SID_NO TYPE SDF

      * run a Turbo Pascal program to find the base thickness
      RUN \pavedb\applicat\model\BASETHCK
      CLEAR
      @10,19 SAY "Retrieving Layer Data for Model File..."

      * copy the result to a temporary file
      SELECT C
      ZAP
      APPEND FROM \pavedb\applicat\model\BASETHCK.DAT TYPE SDF

      * copy the result to the model file
      SELECT A
      REPLACE BASETHCK WITH C->BASETHCK
      REPLACE BASETYPE WITH C->BASETYPE
      ENDIF
* next sid number in the model file
SELECT A
SKIP
ENDDO

* close files
CLOSE ALL
program find_base_thickness;

var
datafile: text;
base_thick,
total_base_thick: real;
err_code,
no_of_records,
index: integer;
data: array[1..30] of dataitem;
base_type,
previous_laynum: string[2];
exit_loop,
most_recent_base: boolean;

begin
assign (datafile, '\pavedb\applicat\model\basethck.dat');

(--read in data needed for the solution)
reset (datafile);
index:= 0;
while not eof(datafile) do
begin
 index:= index + 1;
with data[index] do
begin
  readln (datafile, sid_no, strunum, laynum, laydesc,
          laymatcl, centthk);
  if centthk[1] = ' ' then centthk[1] := '0';
end;
end;
no_of_records := index;
close (datafile);

(--find the total thickness of the most base material)
{initializations}
previous_laynum := '0';
most_recent_base := true;
exit_loop := false;
index := no_of_records;
total_base_thick := 0.0;

{for all the layers starting backwards}
while (index >= 1) and (not exit_loop) do
  with data[index] do
  begin
    {if the layer number is different from the previous one} if laynum <> previous_laynum then
    begin
      {if the layer is a base}
      if laydesc = '5' then
      begin
        {if the base layer is the most recent one}
        if most_recent_base then
        begin
          {make note of the base type}
          base_type := laymatcl;
          {the next base layer will not be the most recent}
          most_recent_base := false;
          {increment the total base thickness}
          val (centthk, base_thick, err_code);
          total_base_thick := total_base_thick + base_thick;
        end
        {if the base layer is not the most recent one} else
        begin
          {if the base material is the same as the most recent one}
          if laymatcl = base_type then
          begin
            {increment the total base thickness}
            val (centthk, base_thick, err_code);
            total_base_thick := total_base_thick + base_thick;
          end
          {if the base material is different from the most recent one}
        end
      end
else
    (exit the loop)
    exit_loop:= true;
end;
end;
end;

(increment the index)
previous_laynum:= laynum;
index:= index - 1;
end;

(output the solution)
rewrite (datafile);
    writeln (datafile, data[1].sid_no, base_type,
total_base_thick:7:2);
close (datafile);
end.
Building Model File

Program Specification 6

Program Name: MODL_WEA.PRG

Purpose: To find the weather data for the model file.

Edit/Procedure Information:

1. For each section identification number in the model file, the program determines if the record for the county in which the section resides exists in the weather database file (WEATHER.DBF). If so, the program finds the annual total freeze thaw cycle values and the mean precipitation values. This is done by summing the monthly values for that county.

2. The result is copied to the model file.

Input File(s):

1. Weather database file (WEATHER.DBF) with index file (WEATHER.NDX).

Output File(s):

1. Model database file (MODEL.DBF).

File Layout(s):

1. WEATHER.DBF (See Appendix A)
2. MODEL.DBF (See Appendix A)
Program Flow for Modl_Wea.Prg

FIGURE 31
Building Model File

Program Specification 6
Program Listing

* File name: MODL_WEAI.FRG
* Program name: modl_wea
* Project 2456: Texas Flexible Pavement Database
* TAMU/TTP
* Written by: Victor Wong
* Created on: June 29, 1988
* Last updated: July 26, 1988
* Purpose: To find the weather data for the model file.

* open files
SELECT A
USE \pavedb\files\MODEL
SELECT B
USE \pavedb\files\WEATHER INDEX \pavedb\indexes\WEATHER

* for each active sid number in the model file
SELECT A
DO WHILE .NOT. EOF()
* find the data in the weather database file
SELECT B
SEEK STR(A->CNTYNUM,3)

* copy the data to the model database file
IF FOUND()
* calculate the annual total freeze thaw cycles and
* mean precipitation values
TOT_F_T = 0
TOT_M_P = 0
DO WHILE CNTYNUM = A->CNTYNUM
    TOT_F_T = TOT_F_T + TFTCMN
    TOT_M_P = TOT_M_P + PRECMN
    SKIP
    ENDDO
REPLACE A->TFTCMN WITH TOT_F_T
REPLACE A->PRECMN WITH TOT_M_P
ENDIF

* skip to the next sid number in model file
SELECT A
SKIP
ENDDO

* close files
CLOSE ALL
Building Model File

Program Specification 7

Program Name: MODL_ENV.PRG

Purpose: To find the environment data for the model file.

Edit/Procedure Information:

1. For each section identification number in the model file, the program determines if the record for the county in which the section resides exists in the environment database file (ENV.DBF). If so, the program copies the Thornthwaite index value to the model file.

Input File(s):

1. Environment database file (ENV.DBF) with index file (ENV.NDX).

Output File(s):

1. Model database file (MODEL.DBF).

File Layout(s):

1. ENV.DBF (See Appendix A)
2. MODEL.DBF (See Appendix A)
Program Flow for Mod1_Env.Prg.

FIGURE 32
Building Model File

Program Specification 7
Program Listing

* File name: MODL_ENV.PRG
* Program name: modl_env
* Project 2456: Texas Flexible Pavement Database
* TAMU/TTI
* Written by: Victor Wong
* Created on: June 29, 1988
* Last updated: July 26, 1988
* Purpose: To find the environmental data for the model file.

* open files
SELECT A
USE \pavedb\files\MODEL
SELECT B
USE \pavedb\files\ENV INDEX \pavedb\indexes\ENV

* for each sid number in the model file
SELECT A
DO WHILE .NOT. EOF()
  * find the data in the env database file
    SELECT B
    SEEK A->CMTYNUM

    * copy the data to the model database file
    IF FOUND()
      * copy the mean thornthwaite index
        REPLACE A->THORNMIN WITH THORNMIN
    ENDF

    * skip to the next sid number in model file
    SELECT A
    SKIP
ENDDO

* close files
CLOSE ALL
Building Model File
Program Specification 8

Program Name: MODL_TRF.PRG

Purpose:
To find the traffic data for the model file.

Edit/Procedure Information:

1. For each section identification number in the model file, the program determines if the section exists in the traffic database file (TRAFFIC.DBF). If so, the program finds the latest average daily traffic (ADT) value and copies it to the model file.

Input File(s):

1. Traffic database file (TRAFFIC.DBF) with index file (TRAFFIC.NDX).

Output File(s):

1. Model database file (MODEL.DBF).

File Layout(s):

1. TRAFFIC.DBF (See Appendix A)
2. MODEL.DBF (See Appendix A)
Program Flow for Modl_Trf.Prg.

FIGURE 33
Building Model File

Program Specification 8
Program Listing

* File name: MODL_TRF.PRG
* Program name: modl_trf
* Project 2456: Texas Flexible Pavement Database
* TAMU/TTI
* Written by: Victor Wong
* Created on: June 29, 1988
* Last updated: July 26, 1988
* Purpose: To find the traffic data for the model file.

* open files
SELECT A
USE \pavedb\files\MODEL
SELECT B
USE \pavedb\files\TRAFFIC INDEX \pavedb\indexes\TRAFFIC

* for each active sid number in the location file
SELECT A
DO WHILE .NOT. EOF()
  * find the data in the traffic database file
  SELECT B
  SEEK STR(A->SID_NO,4)

  * copy the data to the model database file
  IF FOUND()
    * copy the most recent one way ADT value
    DO WHILE SID_NO = A->SID_NO
      SKIP
    ENDDO
    SKIP -1
    REPLACE A->CURADT WITH AADT1WAY
  ENDF

  * skip to the next sid number in model file
  SELECT A
  SKIP
ENDO

* close files
CLOSE ALL
Building Model File

Program Specification 9

Program Name: MODL_SUB.PRG

Purpose: To find the subgrade data for the model file.

Edit/Procedure Information:

1. For each section identification number in the model file, the program determines if the section exists in the subgrade database file (SUBGRADE.DBF). If so, the program finds the subgrade layer in the layer database file (LAYER.DBF) and copies the layer material code number for the subgrade to the model file. The percentage passing 200 sieve value in the subgrade database file is also copied to the model file.

Input File(s):

1. Subgrade database file (SUBGRADE.DBF) with index file (SUBGNDX.NDX).

Output File(s):

1. Model database file (MODEL.DBF).

File Layout(s):

1. SUBGRADE.DBF (See Appendix A)
2. MODEL.DBF (See Appendix A)
Program Flow for Modl_Sub.Prg.

FIGURE 34
Building Model File
Program Specification 9
Program Listing

* File name: MODL_SUB.PRG
* Program name: modl_sub
* Project 2456: Texas Flexible Pavement Database
* TAMU/TII
* Written by: Victor Wong
* Created on: June 29, 1988
* Last updated: July 26, 1988
* Purpose: To find the subgrade data for the model file.

* open files
SELECT A
USE \pavedb\files\MODEL
SELECT B
USE \pavedb\files\SUBGRADE INDEX \pavedb\indexes\SUBGNDX
SELECT C
USE \pavedb\files\LAYER INDEX \pavedb\indexes\LAYNDX

* for each sid number in the model file
SELECT A
DO WHILE .NOT. EOF()
  * find the data in the subgrade database file
  SELECT B
  SEEK STR(A->SID_NO,4)

  * copy the data to the model database file
  IF FOUND()
    * copy the subgrade material type from layer file
    SELECT C
    SEEK STR(B->SID_NO,4)+STR(B->STRUCNUM,2)+STR(B->LAYNUM,2) IF FOUND()
    REPLACE A->SUBGRDITY WITH LAYMATCL
  ENDF

  * copy the percent passing 200 sieve
  SELECT B
  REPLACE A->PPSV200 WITH PPSV200

  * copy the liquid limit
  REPLACE A->LIQLIM WITH LIQLIM
  ENDF

  * skip to the next sid number in model file
  SELECT A
  SKIP
ENDDO

* close files
CLOSE ALL
Building Model File
Program Specification 10

Program Name: MODL_SHO.PRG

Purpose: To find the geometric shoulder data for the model file.

Edit/Procedure Information:

1. For each section identification number in the model file, the program determines if the section exists in the geometric shoulder database file (GEOSH0.DBF). If so, the program finds the latest shoulder layer and copies the shoulder type value to the model file.

Input File(s):

1. Geometric shoulder database file (GEOSH0.DBF) with index file (GEONDX.NDX).

Output File(s):

1. Model database file (MODEL.DBF).

File Layout(s):

1. GEOSH0.DBF (Appendix A.)
2. MODEL.DBF (Appendix A.)
Program Flow for Modl_Sh0.Prg.

FIGURE 35
Building Model File

Program Specification 10
Program Listing

* File name: MODL_SHO.PRG
* Program name: modl_sho
* Project 2456: Texas Flexible Pavement Database
* TAMU/TII
* Written by: Victor Wong
* Created on: June 29, 1988
* Last updated: July 26, 1988
* Purpose: To find the shoulder data for the model file.

* open files
SELECT A
USE \pavedb\files\MODEL
SELECT B
USE \pavedb\files\GOAL_INDEX \pavedb\indexes\GEONDX

* for each sid number in the model file
SELECT A
DO WHILE .NOT. EOF()
  * find the data in the shoulder database file
  SELECT B
  SEEK STR(A->SID_NO,4)

  * copy the data to the model database file
  IF FOUND()
    * go to the most recent surface
    DO WHILE SID_NO = A->SID_NO
      SKIP
    ENDDO
    SKIP -1

    * copy the shoulder surface type
    REPLACE A->SHOFTYP WITH SHOFTYP
  ENDFI

  * skip to the next sid number in model file
  SELECT A
  SKIP
ENDDO

* close files
CLOSE ALL
Building Model File
Program Specification 11

Program Name: MODL_SUP.PRG

Purpose: To find the surface data for the model file.

Edit/Procedure Information:

1. For each section identification number in the model file, the program determines if the section exists in the surface database file (SURFACE.DBF). If so, the program finds the surface layers in the layer database file (LAYER.DBF). The program then adds up the center thickness of the layers. After all the surface layers for a SID have been processed, the program copies the total surface thickness value to the model file and begins processing the next SID.

Input File(s):

1. Surface database file (SURFACE.DBF) with index file (SURFNDX.NDX).
2. Layer database file (LAYER.DBF) with index file (LAYNDX.NDX).

Output File(s):

1. Model database file (MODEL.DBF).

File Layout(s):

1. SURFACE.DBF (See Appendix A.)
2. LAYER.DBF (See Appendix A.)
3. MODEL.DBF (See Appendix A.)
Program Flow for Modl_Suf.Prg.

FIGURE 36
Building Model File

Program Specification 11
Program Listing

* File name: MODL_SUF.PRG
* Program name: modl_suf
* Project 2456: Texas Flexible Pavement Database
* TAMU/TTI
* Written by: Victor Wong
* Created on: June 29, 1988
* Last updated: July 26, 1988
* Purpose: To find the surface data for the model file.

* open files
SELECT A
USE \pavedb\files\MODEL
SELECT B
USE \pavedb\files\SURFACE INDEX \pavedb\indexes\SURFNDX
SELECT C
USE \pavedb\files\LAYER INDEX \pavedb\indexes\LAYNDX

* for each sid number in the model file
SELECT A
DO WHILE .NOT. EOF() 
   * find the data in the surface database file
   SELECT B
   SEEK STR(A->SID_NO,4)

   * copy the data to the model database file
   IF FOUND() 
      * copy the total thickness of surface layers from layer
      * file
      TOT S T = 0
      PREV LAY = 0
      DO WHILE SID_NO = A->SID_NO
         SELECT C
         SEEK STR(B->SID_NO,4)+STR(B->STUCNUM,2)+ ;
         STR(B->LAYNUM,2)
         IF FOUND() .AND. (CENTH_thk <> 99.99) .AND.;
            (LAYNUM <> PREV LAY)
            TOT S T = TOT S T + CENTH_thk
            PREV LAY = LAYNUM
         ENDF
      SELECT B
      SKIP
   ENDDO
   REPLACE A->SURFTHCK WITH TOT S T

   * copy the percent asphalt for most recent surface layer
   SELECT B
   SKIP -1
   REPLACE A->ASAPPLRT WITH ASAPPLRT
ENDIF

* skip to the next sid number in model file
SELECT A
SKIP
ENDDO

* close files
CLOSE ALL
Building Model File
Program Specification 12

Program Name: MODL_TMP.PRG

Purpose: To find the district temperature data for the model file.

Edit/Procedure Information:

1. For each section identification number in the model file, the program determines if the district in which the section resides exists in the district temperature database file (DISTTEMP.DBF). If so, the program copies the temperature constant value to the model file.

Input File(s):

1. District temperature database file (DISTTEMP.DBF) with index file (DISTTEMP.NDX).

Output File(s):

1. Model database file (MODEL.DBF).

File Layout(s):

1. DISTTEMP.DBF (See Appendix A.)

2. MODEL.DBF (See Appendix A.)
Program Flow for Modl_Tmp.Prg.

FIGURE 37
Building Model File

Program Specification 12
Program Listing

* File name: MODL_TMP.PRG
* Program name: modl_tmp
* Project 2456: Texas Flexible Pavement Database
* TAMU/TTI
* Written by: Victor Wong
* Created on: June 29, 1988
* Last updated: July 26, 1988
* Purpose: To find the district temperature constant for * model file.

* open files
SELECT A
USE \paved\files\MODEL
SELECT B
USE \paved\files\DISTTEMP INDEX \paved\indexes\DISTTEMP

* for each sid number in the model file
SELECT A
DO WHILE .NOT. EOF()
   * find the data in the district temperature database file
   SELECT B
   SEEK A->HWYDIST

   * copy the data to the model database file
   IF FOUND()
      SELECT A
      REPLACE TEMPOONS WITH B->TEMPOONS
   ENDIF

   * skip to the next sid number in model file
   SELECT A
   SKIP
ENDDO

* close files
CLOSE ALL
Building Model File
Program Specification 13

Program Name: MODL_DYN.PRG

Purpose: To find the dynaflect data for the model file.

Edit/Procedure Information:

1. For each section identification number in the model file, the program determines if the section exists in the dynaflect database file (DYNAFLDL.DBF). If so, the program finds the average of each sensor reading and copies the results to the model file.

Input File(s):

1. Dynaflec database file (DYNAFLDL.DBF) with index file (DYNAFLDL.NDX).

Output File(s):

1. Model database file (MODEL.DBF).

File Layout(s):

1. DYNAFLDL.DBF (See Appendix A.)
2. MODEL.DBF (See Appendix A.)

FIGURE 38
Building Model File

Program Specification 13
Program Listing

* File name: MODL_DYN.PRG
* Program name: mod1_dyn
* Project 2456: Texas Flexible Pavement Database
* TAMU/TTI
* Written by: Victor Wong
* Created on: June 29, 1988
* Last updated: July 26, 1988
* Purpose: To find the dynaflec data for the model file.

* open files
SELETE A
USE \pavedb\files\MODEL
SELETE B
USE \pavedb\files\DYNAPFLD INDEX \pavedb\indexes\DYNAPFLD

* for each SID number in model file
SELETE A
DO WHILE .NOT. EOF()
    * find the data in the dynaflec database file
    SELETE B
    SEEK STR(A->SID_NO,4)

    * copy the data to the model database file
    IF FOUND()
        AV_SENS1RD = 0
        AV_SENS2RD = 0
        AV_SENS3RD = 0
        AV_SENS4RD = 0
        AV_SENS5RD = 0
        * for all the stations
        DO WHILE SID_NO = A->SID_NO
            * calculate the average sensor reading
            AV_SENS1RD = AV_SENS1RD + SENS1RD
            AV_SENS2RD = AV_SENS2RD + SENS2RD
            AV_SENS3RD = AV_SENS3RD + SENS3RD
            AV_SENS4RD = AV_SENS4RD + SENS4RD
            AV_SENS5RD = AV_SENS5RD + SENS5RD
            SKIP
        ENDDO
        SELECT A
        REPLACE DYNAPFLC1 WITH AV_SENS1RD/14
        REPLACE DYNAPFLC2 WITH AV_SENS2RD/14
        REPLACE DYNAPFLC3 WITH AV_SENS3RD/14
        REPLACE DYNAPFLC4 WITH AV_SENS4RD/14
        REPLACE DYNAPFLC5 WITH AV_SENS5RD/14
    ENDDIF
* skip to the next sid number in model file
  SELECT A
  SKIP
  ENDDO

* close files
CLOSE ALL
Building Model File

Program Specification 14

Program Name: MODL_FAL.PRG

Purpose: To find the falling weight data for the model file.

Edit/Procedure Information:

1. For each section identification number in the model file, the program determines if the section exists in the falling weight database file (FALLWGT.DBF). If so, the program finds the average of each sensor reading and copies the results to the model file.

Input File(s):

1. Falling weight database file (FALLWGT.DBF) with index file (FALLWGT.NDX).

Output File(s):

1. Model database file (MODEL.DBF).

File Layout(s):

1. FALLWGT.DBF (See Appendix A.)
2. MODEL.DBF (See Appendix A.)

FIGURE 39
Building Model File

Program Specification 14
Program Listing

* File name: MODL_FAL.PRG
* Program name: modl_fal
* Project 2456: Texas Flexible Pavement Database
* TAMU/TTI
* Written by: Victor Wong
* Created on: June 29, 1988
* Last updated: July 26, 1988
* Purpose: To find falling weight data for the model file.

* open files
SELECT A
USE \\pavedb\\files\MODEL
SELECT B
USE \\pavedb\\files\FALLWGT INDEX \\pavedb\\indexes\FALLWGT
* for each sid number in the model file
SELECT A
DO WHILE .NOT. EOF()
  * find the data in the falling weight database file
  SELECT B
  SEEK STR(A->SID_NO,4)
  * copy the data to the model database file
  IF FOUND()
    * find the most recent layer
    DO WHILE SID_NO = A-> SID_NO
      SKIP
    ENDDO
    SKIP -1
  * calculate the average ssi reading
    AV_SSIGP1 = (SSIGP11+SSIGP21+SSIGP31+SSIGP41+SSIGP51) / 5
    AV_SSIGP2 = (SSIGP12+SSIGP22+SSIGP32+SSIGP42+SSIGP52) / 5
    AV_SSIGP3 = (SSIGP13+SSIGP23+SSIGP33+SSIGP43+SSIGP53) / 5
    AV_SSIGP4 = (SSIGP14+SSIGP24+SSIGP34+SSIGP44+SSIGP54) / 5
    AV_SSIGP5 = (SSIGP15+SSIGP25+SSIGP35+SSIGP45+SSIGP55) / 5
    SELECT A
    REPLACE SSIGP1 WITH AV_SSIGP1
    REPLACE SSIGP2 WITH AV_SSIGP2
    REPLACE SSIGP3 WITH AV_SSIGP3
    REPLACE SSIGP4 WITH AV_SSIGP4
    REPLACE SSIGP5 WITH AV_SSIGP5
    ENDF

  * skip to the next sid number in model file
  SELECT A
  SKIP
ENDDO

* close files
CLOSE ALL
Section 4: Create the Distress File
Program Narrative for Distress

The Applications program (APPLICAT.PRG) calls the distress program (DISTRESS.PRG and DISTIVISL.PRG) to create the Distress vs. 18 KEAL dBASE File. The file is created from the Master dBASE files - Location, Layer Identification, Traffic, Visual, Geometric and Shoulder Information and Serviceability Index. This file is used to create the 18 KIP graphs as well as to create the Model dBASE file. This program can be found in the subdirectory \PAVEDB\APPLICAT\DISTRESS. The files are stored in the subdirectory \PAVEDB\FILES. The indices are stored in the subdirectory \PAVEDB\INDEXES.
Program Flow for Distress.Prg

FIGURE 40
**Distress vs.18 Keal File**

**PROGRAM SPECIFICATION**

**Program Name:** DISTRESS.PRG

**Program Called:** DISTVISL.PRG

**Purpose:** To create a dBASE III Distress vs 18 Keal File

**Input Files:**

<table>
<thead>
<tr>
<th>File Type</th>
<th>Files</th>
<th>Indices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometric &amp; Shoulder</td>
<td>GEOSHO.DBF</td>
<td>GEONDX.NDX</td>
</tr>
<tr>
<td>Layer Identification File</td>
<td>LAYER.DBF</td>
<td>LAYNDX.NDX</td>
</tr>
<tr>
<td>Location File</td>
<td>LOCATION.DBF</td>
<td>LOCSDID.NDX</td>
</tr>
<tr>
<td>Serviceability Index File</td>
<td>SI.DBF</td>
<td>SI.NDX</td>
</tr>
<tr>
<td>Traffic File</td>
<td>TRAFFIC.DBF</td>
<td>TRAFFIC.NDX</td>
</tr>
<tr>
<td>Visual File</td>
<td>VISUAL.DBF</td>
<td>VISUAL.NDX</td>
</tr>
</tbody>
</table>

**Output Files:**

Distress vs 18 Keal File (DISTRESS.DBF)
**Distress vs. 18 KEAL File**

**File Layout**

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Source</th>
<th>Note**</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID NO</td>
<td>*</td>
<td>4N</td>
<td>L</td>
<td></td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>YEAR</td>
<td>*</td>
<td>2N</td>
<td>A</td>
<td></td>
<td>Year</td>
</tr>
<tr>
<td>A18KEAL</td>
<td></td>
<td>11N</td>
<td>T</td>
<td>1</td>
<td>Annual Cumulative 18 Keal oneway</td>
</tr>
<tr>
<td>RTTT</td>
<td></td>
<td>1N</td>
<td>V</td>
<td>2</td>
<td>Rutting Area</td>
</tr>
<tr>
<td>ALLGCR</td>
<td></td>
<td>1N</td>
<td>V</td>
<td>2</td>
<td>Alligator Cracking Area</td>
</tr>
<tr>
<td>PATCH</td>
<td></td>
<td>1N</td>
<td>V</td>
<td>2</td>
<td>Patch</td>
</tr>
<tr>
<td>SI.MEAN</td>
<td></td>
<td>7.5N</td>
<td>SI</td>
<td>6</td>
<td>Serviceability Index Mean</td>
</tr>
<tr>
<td>CON.YEAR</td>
<td></td>
<td>2N</td>
<td>LI</td>
<td>3</td>
<td>Year of Last Major Construction</td>
</tr>
<tr>
<td>CON.MON</td>
<td></td>
<td>2N</td>
<td>LI</td>
<td>3</td>
<td>Month of Last Major Construction</td>
</tr>
<tr>
<td>PAV.TYP</td>
<td></td>
<td>2N</td>
<td>G</td>
<td>4</td>
<td>Pavement Type</td>
</tr>
<tr>
<td>OVR.YEAR</td>
<td></td>
<td>2N</td>
<td>LI</td>
<td>5</td>
<td>Year of Last HMAC Overlay</td>
</tr>
<tr>
<td>OVR.MON</td>
<td></td>
<td>2N</td>
<td>LI</td>
<td>5</td>
<td>Month of Last HMAC Overlay</td>
</tr>
</tbody>
</table>

*L* - LOCATION  
*T* - TRAFFIC FILE  
*V* - VISUAL FILE  
*LI* - LAYER IDENTIFICATION FILE (LAYER)  
*G* - GEOMETRIC & SHOULDER INFORMATION FILE (GEOSH)  
*SI* - SERVICEABILITY INDEX FILE  

**NOTE:** The referenced note can be found on the following page.
NOTES:

A. A record is created for 1972 and for every year there after. (eg. If the year of the run is 1987, there is a record for 1972, 73, 74, 75 . . . 87).

1. The 18 Keal is accumulated from the date of last major construction until and including 1972 for the first record. Every record there after only has the 18 Keal for that year.

2. This field is initialized to 9. The RUTTSL, RUTTMD, and RUTTSV fields are checked. If a 1, 2, or 3 exists in one of these fields, a 1, 2, or 3 is put in RUTT. If zeros exist in all three fields, a zero is put in RUTT. If a VISUAL record does not exist for the year, a 9 is put in RUTT (Note: The single digit fields RUTTSL, RUTTMD, and RUTTSV will have the values 000, 100, 200, 300, 010, 020, 030, 001, 002, or 003). The same process is followed for ALLGCRSL, ALLGCRMD, ALLGCRSV and PATCHFR, PATCHFR, PATCHGD.

3. If the last major construction was before January 1, 1973, the year and month of the most recent base layer is used. If the section (SID record) was constructed after December 31, 1972, the year and month of the subgrade is used.

4. The Pavement Type which corresponds to the year of the record is used.

5. The year and month of the most recent HMAC overlay for the year of the Distress vs 18 KIP record is used.

6. This field is initialized to 9.99999. So if a SI record does not exist, this field should have a value of 9.99999.
* SUBSYSTEM: APPLICATIONS
* PROGRAM NAME: DISTRESS.FRG  5/23/88
* CALLED FROM: APPLICAT.FRG
* REVISED ON: 06/02/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMI/TII
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO CREATE THE DISTRESS VS 18 KEAL FILE

* THE FOLLOWING FILES ARE NEEDED TO RUN THIS PROGRAM:
* GEOSHO.DBF  GEONDX.NDX
* LAYER.DBF  LAYNDX.NDX
* TRAFFIC.DBF  TRAFFIC.NDX

SET TALK OFF
SET ECHO OFF
SET SAFETY OFF
CLEAR ALL

* assign databases to different work areas
SELECT 1
IF .NOT. FILE('\PAVEDB\FILES\DISTRESS.DBF')
   ? "DISTRESS FILE not found. Please Check . . . "
   WAIT
   RETURN TO MASTER
ENDIF
IF .NOT. FILE('\PAVEDB\INDEXES\DISTRESS.NDX')
   ? "DISTRESS INDEX not found. Please Check . . . "
   WAIT
   RETURN TO MASTER
ENDIF
USE \PAVEDB\FILES\DISTRESS INDEX \PAVEDB\INDEXES\DISTRESS
ZAP

SELECT 2
USE \PAVEDB\FILES\TRAFFIC INDEX \PAVEDB\INDEXES\TRAFFIC

SELECT 4
USE \PAVEDB\FILES\LAYER INDEX \PAVEDB\INDEXES\LAYNDX

SELECT 5
USE \PAVEDB\FILES\GEOSHO INDEX \PAVEDB\INDEXES\GEONDX

SELECT 7
USE \PAVEDB\FILES\LOCATION INDEX \PAVEDB\INDEXES\LOCSID

* INITIALIZE VARIABLES
CLEAR
MYEAR = 1972
MYEAR2 = 72
@ 10, 10 Say " "
MTEMP = " "
ACCEPT "Please enter the year of the run (19YY): " to MTEMP
MYEARRUN = VAL(MTEMP)
CLEAR
SET STAT ON
@ 10, 10 Say "Processing. Please wait . . . . ."

SELECT 7
GOTO TOP
DO WHILE .NOT. EOF()
  DO WHILE MYEARRUN >= MYEAR
    * Initialize variables
    STORE 9 TO MEXIT, MALLGCR, MPATCH
    STORE 9.99999 TO MSIMean
    STORE 0 TO MAAKEAL
    STORE 9999 TO MSID_NO
    STORE 99 TO MOONYEAR, MCONMON, MPAVEYTP, MOVYR, MOVRRMON

    * Getting Sid no from Location file
    MSID_NO = G->SID_NO
    MSTRUCT = 1
    MLAY = 1
    IF .NOT. G->ACTFLAG .AND. G->INACTION < MYEAR2
      MYEAR = MYEAR + 1
    LOOP
  ENDF

  * GET CONSTRUCTION YEAR & MONTH FROM LAYER IDENTIFICATION FILE
  SELE 4
  SEEK STR(MSID_NO,4)+STR(MSTRUCT,2)+STR(MLAY,2)
  IF FOUND()
    IF D->JOBCOMM > 1972
      MOONYEAR = D->JOBCOMM
      MCONMON = D->JOBCOMMMOV
      MSTRUCT = D->STRUCTNUM
    ELSE
      DO WHILE D->SID_NO = MSID_NO
        SKIP
      ENDDO
      SKIP -1
      DO WHILE D->JOBCOMM > MYEAR2
        SKIP -1
      ENDDO
      MSTRUCT = D->STRUCTNUM
      DO WHILE .NOT. BOF() .AND. D->LAYDESC <> 5
        SKIP -1
      ENDDO
      IF D->SID_NO <> MSID NO
        MOONYEAR = 0
        MCONMON = 0
      ELSE
        MOONYEAR = D->JOBCOMM
      END
MCONMON = D->JOBCMIPR
ENDIF
ENDIF
ELSE
MCONVYEAR = 0
MCONMON = 0
MBSTRUC = 0
ENDIF

* GET PAVEMENT TYPE FROM GEOMETRIC AND SHOULDER INFORMATION FILE
SELE 5
SEEK STR(MSID_NO,4)+STR(MBSTRUC,2)
IF FOUND()
  MPAVETYP = E->PAVETYP
  IF WIDENNFIG = 2
    MYEAR = MYEAR + 1
  LOOP
ENDIF
ELSE
  MPAVETYP = 0
ENDIF

* Getting 18 Kip from Traffic Data
SELE 2
IF MYEAR = 1972
  SEEK STR(MSID_NO,4)
  DO WHILE B->SID_NO = MSID_NO .AND. B->YEAR < 1973
    MA18KEAL = MA18KEAL + B->A18KEAL
    SKIP
  ENDDO
ELSE
  SEEK STR(MSID_NO,4)+STR(MYEAR,4)
  IF FOUND()
    MA18KEAL = B->A18KEAL
  ENDIF
ENDIF

* GET OVERLAY INFORMATION FROM LAYER IDENTIFICATION FILE
SELE 4
SEEK STR(MSID_NO,4)
IF FOUND()
  DO WHILE D->SID_NO = MSID_NO
    SKIP
  ENDDO
  SKIP -1
  DO WHILE D->JOBCMIPR > MYEAR2
    SKIP -1
  ENDDO
  DO WHILE D->LAYMATCL <> 1 .AND. D->LAYMATCL <> 2 .AND. D->LAYMATCL <> 4
    SKIP -1
  ENDDO
  IF D->SID_NO = MSID_NO .AND. (D->LAYMATCL=1 .OR. D->LAYMATCL=2 .OR. D->LAYMATCL=4)
    MOVRYEAR = D->JOBCMIPR
  ENDIF
MOVRFMON = D->JOBCOMMNO
ELSE
  MOVRYEAR = 0
  MOVRFMON = 0
ENDIF
ELSE
  MOVRYEAR = 0
  MOVRFMON = 0
ENDIF

SELECT 1
APPEND BLANK
REPLACE SID NO  WITH MSID NO
REPLACE YEAR    WITH MYEAR2
REPLACE A18KEAL WITH MA18KEAL
REPLACE RUIT    WITH MRUIT
REPLACE ALLGCR  WITH MALLGCR
REPLACE PATCH   WITH MPATCH
REPLACE SIMEAN  WITH MSIMEAN
REPLACE CONYEAR WITH MONYEAR
REPLACE COMMON  WITH MONMON
REPLACE PAVETYP WITH MAPAVETYP
REPLACE OVRYEAR WITH MOVRYEAR
REPLACE OVRMON  WITH MOVRFMON

MYEAR = MYEAR + 1
MYEAR2 = MYEAR2 + 1
ENDDO
MYEAR = 1972
MYEAR2 = 72
SELECT 7
SKIP
ENDDO
CLOSE DATABASES
CLEAR
DO \PAVEDB\EDITUPD\DISTVISL
CLEAR ALL
SET STAT OFF
RETURN
PROGRAM LISTING

* * SUBSYSTEM: APPLICATIONS
* PROGRAM NAME: DISTVISL.PRG 5/23/88
* REVISED ON: 07/01/88
* CALLED FROM: DISTRESS.PRG
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TII
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO ADD THE VISUAL & SI INFORMATION TO THE
* DISTRESS VS 18 KEAL FILE
*
* THE FOLLOWING FILES ARE NEEDED TO RUN THIS PROGRAM:
* SI.DBF SL.NDX
* SIDIST.NDX (This File is not mandatory)
* VISUAL.DBF VISUAL.NDX
* VISLDIST.NDX (This File is not mandatory)
* DISTRESS.DBF DISTRESS.DBF
*
* assign databases to different work areas
SET SAFETY OFF
SELECT 1
USE \PAVEDB\FILES\DISTRESS INDEX \PAVEDB\INDEXES\DISTRESS

SELECT 2
USE \PAVEDB\FILES\VISUAL
INDEX ON STR(SID_NO,4)+STR(YEAR,2) TO \PAVEDB\INDEXES\VISLDIST

SELECT 3
USE \PAVEDB\FILES\SI
INDEX ON STR(SID_NO,4)+STR(YEAR,2) TO \PAVEDB\INDEXES\SIDIST

* INITIALIZE VARIABLES
SELECT 1
GO TO TOP
STORE 9 TO MRUTT, MALGCR, MPATCH
STORE 9.99999 TO MSIMEAN
MSID NO = A->SID NO
MYEAR2 = A->YEAR
DO WHILE .NOT. EOF()
    STORE 9 TO MRUTT, MALGCR, MPATCH
    STORE 9.99999 TO MSIMEAN

* GET VISUAL INFORMATION
SELECT 2
SEEK STR(MSID NO,4)+STR(MYEAR2,2)
IF FOUND()
    DO CASE
    CASE RUTTSL <> 0
        MRUTT = B->RUTTSL
    CASE RUTMID <> 0
        MRUTT = B->RUTMID

596
CASE RUTTSV <> 0
    MRUTT = B->RUTTSV
otherwise
    MRUTT = 0
endcase
do case
    case ALLGCRSL <> 0
        MALLGCR = B->ALLGCRSL
    case ALLGCRMID <> 0
        MALLGCR = B->ALLGCRMID
    case ALLGCRSV <> 0
        MALLGCR = B->ALLGCRSV
otherwise
    MALLGCR = 0
endcase
do case
    case PATCHGD <> 0
        MPATCH = B->PATCHGD
    case PATCHFR <> 0
        MPATCH = B->PATCHFR
    otherwise
        MPATCH = 0
endcase
else
    MRUTT = 9
    MALLGCR = 9
    MPATCH = 9
endif

g* get SI information
select 3
seek str(MSID_NO,4)+str(MYEAR2,2)
if found()
    MSIMEAN = C->SIMEAN
else
    MSIMEAN = 9.99999
endif
select 1
replace RUTT WITH MRUTT
replace ALLGCR WITH MALLGCR
replace PATCH WITH MPATCH
replace SIMEAN WITH MSIMEAN
skip
MSID_NO = A->SID_NO
MYEAR2 = A->YEAR
endo
close databases
clear
set safety on
return
CHAPTER 6

DATABASE MAINTENANCE SUBSYSTEM
GENERAL NARRATIVE

The Database Maintenance programs permit the user to reindex files, change the installation defaults or backup all the files. These programs are selected from the Main Menu of the Flexible Pavement System and are called by the main program DBMAIN.PRG. See the Main Menu Program Flow Diagram (Figure 5) in Chapter 1.

The Reindex program (REINDEX.PRG) reindexes all the master files and some of the temporary files. All the files used are in the subdirectory \PAVEDB\FILES and the respective indices are in \PAVEDB\INDEXES. REINDEX.PRG is stored in \PAVEDB.

The Installation program (INSTDEFL.PRG) sets up the drive that is to be used, the floppy drive to backup up files to and the defaults for the printers. The defaults are stored in a memory variable file in INSTALL.MEM. At the start of the program, the file is loaded into memory. The defaults can be changed at any time by going through the Installation Menu (choice 6 on the Main Menu). INSTDEFL.PRG and INSTALL.MEM are stored in \PAVEDB.

The Backup program (BACKUP.PRG) backs up all the master files to the floppy disk drive designated in the installation menu. This program is in \PAVEDB\BACKUP.
**Program Specification**

**Program Name:** REINDEX.PRG

**Purpose:** To reindex all the master files.

**Input/Output Files:**

The following files are used along with their indices:

<table>
<thead>
<tr>
<th>Master Files/Table</th>
<th>Data File</th>
<th>Index File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>LOCATION.DBF</td>
<td>LOCSSID.NDX</td>
</tr>
<tr>
<td>Layer Identification</td>
<td>LAYER.DBF</td>
<td>LAYNDX.NDX</td>
</tr>
<tr>
<td>Geometric &amp; Shoulder</td>
<td>GEOSHO.DBF</td>
<td>GEONDX.NDX</td>
</tr>
<tr>
<td>Surface</td>
<td>SURFACE.DBF</td>
<td>SURFNDX.NDX</td>
</tr>
<tr>
<td>Subgrade</td>
<td>SUBGRADE.DBF</td>
<td>SUBGNDX.NDX</td>
</tr>
<tr>
<td>Layer Thickness</td>
<td>LAYTHICK.DBF</td>
<td>LAYTNDX.NDX</td>
</tr>
<tr>
<td>Visual Rating</td>
<td>VISUAL.DBF</td>
<td>VISUAL.NDX</td>
</tr>
<tr>
<td>Serviceability Index</td>
<td>ST.DBF</td>
<td>SI.NDX</td>
</tr>
<tr>
<td>Falling Weight SSI</td>
<td>FALLIGHT.DBF</td>
<td>FALLIGHT.NDX</td>
</tr>
<tr>
<td>Dynasflcet Measure</td>
<td>DYNAFLLD.DBF</td>
<td>DYNAFLLD.NDX</td>
</tr>
<tr>
<td>Skid Measurement</td>
<td>SKID.DBF</td>
<td>SKID.NDX</td>
</tr>
<tr>
<td>Environment</td>
<td>ENV.DBF</td>
<td>ENV.NDX</td>
</tr>
<tr>
<td>Weather</td>
<td>WEATHER.DBF</td>
<td>WEATHER.NDX</td>
</tr>
<tr>
<td>Traffic</td>
<td>TRAFFIC.DBF</td>
<td>TRAFFIC.NDX</td>
</tr>
<tr>
<td>County Name Table</td>
<td>CNTYTBTL.DBF</td>
<td>CNTYTBNO.NDX</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Temporary Files</th>
<th>Data File</th>
<th>Index File</th>
</tr>
</thead>
<tbody>
<tr>
<td>PES SSI</td>
<td>PESSION.DBF</td>
<td>PESSION.NDX</td>
</tr>
<tr>
<td>PES Skid</td>
<td>PESSKID.DBF</td>
<td>PESSKID.NDX</td>
</tr>
<tr>
<td>PES Visual</td>
<td>PESVISL.DBF</td>
<td>PESVISL.NDX</td>
</tr>
<tr>
<td>PES MRM</td>
<td>PESMRM.DBF</td>
<td>PESMRM.NDX</td>
</tr>
<tr>
<td>PES General</td>
<td>PESGEN.DBF</td>
<td>PESGEN.NDX</td>
</tr>
<tr>
<td>PES Scores</td>
<td>PESSCR.DBF</td>
<td>PESSCR.NDX</td>
</tr>
<tr>
<td>Location</td>
<td>LOCNCHNG.DBF</td>
<td>LOCNCHNG.NDX</td>
</tr>
<tr>
<td>Layer</td>
<td>LOCN_NEW.DBF</td>
<td>LOCN_NEW.NDX</td>
</tr>
<tr>
<td>Geometric and Shoulder</td>
<td>LAYRCHNG.DBF</td>
<td>LAYRCHNG.NDX</td>
</tr>
<tr>
<td>Layer Thickness</td>
<td>LAYR_NEW.DBF</td>
<td>LAYR_NEW.NDX</td>
</tr>
<tr>
<td>Surface</td>
<td>GEOSCHNG.DBF</td>
<td>GEOSCHNG.NDX</td>
</tr>
<tr>
<td>Subgrade</td>
<td>GEOS_NEW.DBF</td>
<td>GEOS_NEW.NDX</td>
</tr>
<tr>
<td>Layer Thickness</td>
<td>LAYTCHNG.DBF</td>
<td>LAYTCHNG.NDX</td>
</tr>
<tr>
<td>Surface</td>
<td>LAYT_NEW.DBF</td>
<td>LAYT_NEW.NDX</td>
</tr>
<tr>
<td>Subgrade</td>
<td>SURFCHNG.DBF</td>
<td>SURFCHNG.NDX</td>
</tr>
<tr>
<td></td>
<td>SURF_NEW.DBF</td>
<td>SURF_NEW.NDX</td>
</tr>
<tr>
<td></td>
<td>SUBGCHNG.DBF</td>
<td>SUBGCHNG.NDX</td>
</tr>
<tr>
<td></td>
<td>SUBG_NEW.DBF</td>
<td>SUBG_NEW.NDX</td>
</tr>
</tbody>
</table>
Program REINDEX

* SUBSYSTEM: DATABASE MAINTENANCE
* PROGRAM NAME: REINDEX.PRG 010/04/88
* MODIFIED ON: 10/21/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TIT
* AUTHOR: TREVOR X. PEREIRA
* PURPOSE: TO REINDEX ALL THE MASTER AND TEMPORARY FILES USED IN THE FLEXIBLE PAVEMENT SYSTEM
*
CLEAR
SET PATH TO \PAVEDB\FILES;\PAVEDB\INDEXES;\PAVEDB\EDITUPDT
SET STAT ON
SET ECHO Off
SET TALK Off
@ 3, 10 SAY "REINDEXING Files. Please Wait . . ."
? " 
? "Reindexing Layer File . . ."
USE LAYER INDE LAYNDX
REIN
? "Reindexing Layer Thickness File . . ."
USE LAYTHICK INDE LAYNDX
REIN
? "Reindexing Geometric & Shoulder File . . ."
USE GEOSHON INDE GEONDX
REIN
? "Reindexing Surface File . . ."
USE SURFACE INDE SURFNDX
REIN
? "Reindexing Subgrade File . . ."
USE SUBGRADE INDE SUBGNDX
REIN
? "Reindexing Serviceability Index File . . ."
USE SI INDE SI
REIN
USE VISUAL INDE VISUAL
REIN
? "Reindexing Skid File . . ."
USE SKID INDE SKID
REIN
? "Reindexing Dynaflect File . . ."
USE DYNAFLD INDE DYNAFLLD
REIN
? "Reindexing Falling Weight File . . ."
USE FALLWHT INDE FALLWHT
REIN
? "Reindexing Environment File . . ."
USE ENV INDE ENV

603
? "Reindexing Weather File . . ."
USE WEATHER INDE WEATHER
REIN
? "Reindexing Location File . . ."
USE LOCATION INDE LOCSD
REIN
? "Reindexing Traffic File . . ."
USE TRAFFIC INDE TRAFFIC
REIN
? "Reindexing all PES Files . . ."
USE PESSSI INDE PESSSI
REIN
USE PESSKD INDE PESSKD
REIN
USE PESVISL INDE PESVISL
REIN
USE PESMRM INDE PESMRM
REIN
USE PESGEN INDE PESGEN
REIN
USE PESSCR INDE PESSCR
REIN

? "Reindexing all temporary Inventory Files . . ."
IF FILE('"PAVEDB\EDITUPDT\LAYRCHNG.DBF")
    USE LAYRCHNG INDE LAYRCHNG
    REIN
ENDIF
IF FILE('"PAVEDB\EDITUPDT\LOCNCHNG.DBF")
    USE LOCNCHNG INDE LOCNCHNG
    REIN
ENDIF
IF FILE('"PAVEDB\EDITUPDT\LAYTCHNG.DBF")
    USE LAYTCHNG INDE LAYTCHNG
    REIN
ENDIF
IF FILE('"PAVEDB\EDITUPDT\GEOSCHNG.DBF")
    USE GEOSCHNG INDE GEOSCHNG
    REIN
ENDIF
IF FILE('"PAVEDB\EDITUPDT\SURFCHNG.DBF")
    USE SURFCHNG INDE SURFCHNG
    REIN
ENDIF
IF FILE('"PAVEDB\EDITUPDT\SUBGCHNG.DBF")
    USE SUBGCHNG INDE SUBGCHNG
    REIN
ENDIF
USE LAYR_NEW INDE LAYR_NEW
REIN
USE LOCN_NEW INDE LOCN_NEW
REIN
USE LAYT_NEW INDE LAYT_NEW
REIN
USE GEOS_NEW INDE GEOS_NEW
REIN
USE SURF_NEW INDE SURF_NEW
REIN
USE SUBG_NEW INDE SUBG_NEW
REIN
IF FILE("\PAVEDB\FILES\DISTRESS")
   USE \PAVEDB\FILES\DISTRESS INDE \PAVEDB\INDEXES\DISTRESS
   REIN
ENDIF

CLOSE ALL
SET PATH TO
? "Reindexing done ! ! !"
WAIT
SET STAT OFF
PROGRAM SPECIFICATION

Program Name:  INSIDEFL.PRG

Purpose:  To set up the printer and drive defaults for the system.

Input/Output Files:

Memory Variable File - INSTALL.MEM
Program INSIDEFL

* SUBSYSTEM: DATABASE MAINTAINENCE
* PROGRAM NAME: INSIDEFL.PRG  09/19/88
* PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* TAMU/TII
* AUTHOR: TREvor X. PEREIRA
* PURPOSE: TO INSTALL THE MAIN MENU SYSTEM DEFAULTS
* AND STORE THEM TO A MEMORY VARIABLE FILE.
*
SET SAFETY OFF
CLEA
@ 3, 25 SAY "INSTALLATION MENU"
@ 6, 10 SAY "Please enter the drive being used (NORMALLY C )" GET IDRIVE
@ 7, 15 SAY "DO NOT specify drive 'A' or drive 'B'"
@ 9, 10 SAY "Please enter the drive you want to BACK UP files:" GET IDRIVE2
@ 10, 15 SAY "MUST Specify 'A' or 'B' only"
@ 12, 10 SAY "Which port do you want to send the SUMMARY REPORT: " GET IPORT1
@ 13, 10 SAY "and other Landscape reports"
@ 14, 15 SAY "Specify LPT1, LPT2 or LPT3"
@ 16, 10 SAY "Which port do you want to send the INVENTORY UPDATE REPORT: "
GET IPORT2
@ 17, 15 SAY "Specify LPT1, LPT2 or LPT3"
READ
SET DEFAULT TO &IDRIVE
MDRIVE = IDRIVE
MDRIVE2 = IDRIVE2
MPORT1 = IPORT1
MPORT2 = IPORT2
SAVE TO \PAVEDB\INSTALL ALL LIKE I*
RELE ALL LIKE I*
SET SAFETY ON
RETURN
**Program Specification**

**Program Name:** BACKUP.PRG

**Purpose:** To backup the master files to floppy diskettes.

**Input/Output Files:**

The following files are used:

<table>
<thead>
<tr>
<th>Location</th>
<th>LOCATION.DBF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer Identification</td>
<td>LAYER.DBF</td>
</tr>
<tr>
<td>Geometric &amp; Shoulder</td>
<td>GEOSHO.DBF</td>
</tr>
<tr>
<td>Surface</td>
<td>SURFACE.DBF</td>
</tr>
<tr>
<td>Subgrade</td>
<td>SUBGRADE.DBF</td>
</tr>
<tr>
<td>Layer Thickness</td>
<td>LAYTHICK.DBF</td>
</tr>
<tr>
<td>Visual Rating</td>
<td>VISUAL.DBF</td>
</tr>
<tr>
<td>Serviceability Index</td>
<td>SI.DBF</td>
</tr>
<tr>
<td>Falling Weight SSI</td>
<td>FALLWGT.DBF</td>
</tr>
<tr>
<td>Dynaflct Measure</td>
<td>DYNAPLID.DBF</td>
</tr>
<tr>
<td>Skid Measurement</td>
<td>SKID.DBF</td>
</tr>
<tr>
<td>Environment</td>
<td>ENV.DBF</td>
</tr>
<tr>
<td>Weather</td>
<td>WEATHER.DBF</td>
</tr>
<tr>
<td>Traffic</td>
<td>TRAFFIC.DBF</td>
</tr>
</tbody>
</table>
Program BACKUP

* * SUBSYSTEM: BACKUP.PRG
* * PROGRAM NAME: BACKUP.PRG 07/03/88
* * MODIFIED ON: 09/20/88
* * CALLED FROM: DOMAIN.PRG
* * PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
* * TAMU/TII
* * AUTHOR: TREVOR X. PEREIRA
* * PURPOSE: TO BACKUP DATA TO FLOPPY DISKETTES
*
CLEAR
SET TYPE TO 0
STORE 0 TO MSIZE, MNUMFIELDS, MHEADER, MOUNT, MDISKS, MBYTES
STORE 0 TO MLOCATION, MSLAYER, MLAYTHICK, MGEOSHO, MSURFACE, MSUBGRADE,
MVISUAL
STORE 0 TO MDYNAFLD, MSI, MIRAFFIC, MWWEATHER, MPALLIGHT, MENV, MSKID
STORE "" TO MNULL
SET ESCAPE OFF
@ 10, 10 SAY "Calculating file sizes. Please Wait . . . ."

SET DEFAULT TO &MDRIVE
MBACKDR = MDRIVE2 + ":"
* Making sure the Master Files are present
IF .NOT. FILE("\PAVEDB\FILES\LOCATION.DBF"
    ? "LOCATION file is not found. Please Check . . ."
    WAIT
    RETURN TO MASTER
ENDIF
IF .NOT. FILE("\PAVEDB\FILES\LAYER.DBF"
    ? "LAYER file is not found. Please Check . . ."
    WAIT
    RETURN TO MASTER
ENDIF
IF .NOT. FILE("\PAVEDB\FILES\LAYTHICK.DBF"
    ? "LAYTHICK file is not found. Please Check . . ."
    WAIT
    RETURN TO MASTER
ENDIF
IF .NOT. FILE("\PAVEDB\FILES\GEOSHO.DBF"
    ? "GEOSHO file is not found. Please Check . . ."
    WAIT
    RETURN TO MASTER
ENDIF
IF .NOT. FILE("\PAVEDB\FILES\SURFACE.DBF"
    ? "SURFACE file is not found. Please Check . . ."
    WAIT
    RETURN TO MASTER
ENDIF
IF .NOT. FILE("\PAVEDB\FILES\SUBGRADE.DBF")

609
? "SUBGRADE file is not found. Please Check . . ."
WAIT
RETURN TO MASTER
ENDIF
IF .NOT. FILE("\PAVEDB\FILES\VISUAL.DBF")
? "VISUAL file is not found. Please Check . . ."
WAIT
RETURN TO MASTER
ENDIF
IF .NOT. FILE("\PAVEDB\FILES\DYNAFLD.DBF")
? "DYNAFLD file is not found. Please Check . . ."
WAIT
RETURN TO MASTER
ENDIF
IF .NOT. FILE("\PAVEDB\FILES\SI.DBF")
? "SI file is not found. Please Check . . ."
WAIT
RETURN TO MASTER
ENDIF
IF .NOT. FILE("\PAVEDB\FILES\TRAFFIC.DBF")
? "TRAFFIC file is not found. Please Check . . ."
WAIT
RETURN TO MASTER
ENDIF
IF .NOT. FILE("\PAVEDB\FILES\WEATHER.DBF")
? "WEATHER file is not found. Please Check . . ."
WAIT
RETURN TO MASTER
ENDIF
IF .NOT. FILE("\PAVEDB\FILES\FALLOWGT.DBF")
? "FALLOWGT file is not found. Please Check . . ."
WAIT
RETURN TO MASTER
ENDIF
IF .NOT. FILE("\PAVEDB\FILES\ENV.DBF")
? "ENV file is not found. Please Check . . ."
WAIT
RETURN TO MASTER
ENDIF
IF .NOT. FILE("\PAVEDB\FILES\SKID.DBF")
? "SKID file is not found. Please Check . . ."
WAIT
RETURN TO MASTER
ENDIF
* Calculation file sizes of all files
USE \PAVEDB\FILES\LOCATION
STORE RECOUNT() * RESIZE() TO MSIZE
MNULL = ""
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MLOCATION = MSIZE + MHEADER + 20
MBYTES = MBYTES + MLOCATION
MDISKs = 1

USE \PAVEDB\FILES\LAYER
MNULL = ""
STORE 0 TO MNUMFIELDS, MHEADER, MSIZE
STORE RECOUNT() * RECSIZE() TO MSIZE
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
   MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MHEADER = MSIZE + MHEADER + 20
MBYTES = MBYTES + MHEADER
IF MBYTES > 360000
   MBYTES = MHEADER
   MDISKs = MDISKs + 1
ENDIF

USE \PAVEDB\FILES\LAYTHICK
MNULL = ""
STORE 0 TO MNUMFIELDS, MHEADER, MSIZE
STORE RECOUNT() * RECSIZE() TO MSIZE
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
   MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MIAYTHICK = MSIZE + MHEADER + 20
MBYTES = MBYTES + MIAYTHICK
IF MBYTES > 360000
   MBYTES = MIAYTHICK
   MDISKs = MDISKs + 1
ENDIF

USE \PAVEDB\FILES\GEOSH0
MNULL = ""
STORE 0 TO MNUMFIELDS, MHEADER, MSIZE
STORE RECOUNT() * RECSIZE() TO MSIZE
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
   MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MGEOSH0 = MSIZE + MHEADER + 20
MBYTES = MBYTES + MGEOSH0
IF MBYTES > 360000
   MBYTES = MGEOSH0
   MDISKs = MDISKs + 1
ENDIF

USE \PAVEDB\FILES\SURFACE
MNULL = ""
STORE 0 TO MNUMFIELDS, MHEADER, MSIZE
STORE RECOUNT() * RECSIZE() TO MSIZE
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
   MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MSURFACE = MSIZE + MHEADER + 20
MBYTES = MBYTES + MSURFACE
IF MBYTES > 360000
    MBYTES = MSURFACE
    MDISKS = MDISKS + 1
ENDIF

USE \PAVEDB\FILES\SUBGRADE
MNULL = ""
STORE 0 TO MNUMFIELDS, MHEADER, MSIZE
STORE RECOUNT() * RECSIZE() TO MSIZE
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MSUBGRADE = MSIZE + MHEADER + 20
MBYTES = MBYTES + MSUBGRADE
IF MBYTES > 360000
    MBYTES = MSUBGRADE
    MDISKS = MDISKS + 1
ENDIF

USE \PAVEDB\FILES\VISUAL
MNULL = ""
STORE 0 TO MNUMFIELDS, MHEADER, MSIZE
STORE RECOUNT() * RECSIZE() TO MSIZE
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MVISUAL = MSIZE + MHEADER + 20
MBYTES = MBYTES + MVISUAL
IF MBYTES > 360000
    MBYTES = MVISUAL
    MDISKS = MDISKS + 1
ENDIF

USE \PAVEDB\FILES\DYNAPLID
MNULL = ""
STORE 0 TO MNUMFIELDS, MHEADER, MSIZE
STORE RECOUNT() * RECSIZE() TO MSIZE
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MDYNAPLID = MSIZE + MHEADER + 20
MBYTES = MBYTES + MDYNAPLID
IF MBYTES > 360000
    MBYTES = MDYNAPLID
    MDISKS = MDISKS + 1
ENDIF
USE \PAVEDB\FILES\FALLWGT
MNULL = ""
STORE 0 TO MNUMFIELDS, MHEADER, MSIZE
STORE RECCOUNT() * RECSIZE() TO MSIZE
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MFALLWGT = MSIZE + MHEADER + 20
MBYTES = MBYTES + MFALLWGT
IF MBYTES > 360000
    MBYTES = MFALLWGT
    MDISK5 = MDISK5 + 1
ENDIF

USE \PAVEDB\FILES\SI
MNULL = ""
STORE 0 TO MNUMFIELDS, MHEADER, MSIZE
STORE RECCOUNT() * RECSIZE() TO MSIZE
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MSI = MSIZE + MHEADER + 20
MBYTES = MBYTES + MSI
IF MBYTES > 360000
    MBYTES = MSI
    MDISK5 = MDISK5 + 1
ENDIF

USE \PAVEDB\FILES\TRAFFIC
MNULL = ""
STORE 0 TO MNUMFIELDS, MHEADER, MSIZE
STORE RECCOUNT() * RECSIZE() TO MSIZE
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MTRAFFIC = MSIZE + MHEADER + 20
MBYTES = MBYTES + MTRAFFIC
IF MBYTES > 360000
    MBYTES = MTRAFFIC
    MDISK5 = MDISK5 + 1
ENDIF

USE \PAVEDB\FILES\WEATHER
MNULL = ""
STORE 0 TO MNUMFIELDS, MHEADER, MSIZE
STORE RECCOUNT() * RECSIZE() TO MSIZE
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MWEATHER = MSIZE + MHEADER + 20
MEBYTES = MEBYTES + MWEATHER
IF MEBYTES > 360000
    MEBYTES = MWEATHER
    MDISK5 = MDISK5 + 1
ENDIF

USE \PAVEDB\FILES\ENV
MNULL = ""
STORE 0 TO MNUMFIELDS, MHEADER, MSIZE
STORE RECCOUNT() * RECSIZE() TO MSIZE
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MSIZE = MSIZE + MHEADER + 20
MEBYTES = MEBYTES + MSIZE
IF MEBYTES > 360000
    MEBYTES = MSIZE
    MDISK5 = MDISK5 + 1
ENDIF

USE \PAVEDB\FILES\SKID
MNULL = ""
STORE 0 TO MNUMFIELDS, MHEADER, MSIZE
STORE RECCOUNT() * RECSIZE() TO MSIZE
DO WHILE MNULL < FIELD(MNUMFIELDS + 1)
    MNUMFIELDS = MNUMFIELDS + 1
ENDDO
MHEADER = (32 * MNUMFIELDS) + 34
MSKID = MSIZE + MHEADER + 20
MEBYTES = MEBYTES + MSKID
IF MEBYTES > 360000
    MDISK5 = MDISK5 + 1
ENDIF

USE
MGET = ""
@ 10, 05 SAY "You will need " + STR(MDISK5, 3) + " blank formatted disk(s) of 360 K"
@ 12, 05 SAY "If you do not have the diskettes ready, press the Esc key."
@ 13, 05 SAY "Otherwise press any key to continue."
CLEA TYPE
READ
IF READKEY() = 12
    CLEAR
    RETURN TO MASTER
ELSE
    SET SAFETY ON
    CLEAR
    @ 5, 0 SAY "Please wait. Backing up LOCATION files . . ."
    IF FILE("\PAVEDB\FILES\BACKUP\LOCATION.DBF")
        DELE FILE \PAVEDB\FILES\BACKUP\LOCATION.DBF
    ENDFILE
    COPY FILE \PAVEDB\FILES\LOCATION.DBF TO \PAVEDB\FILES\BACKUP\LOCATION.DBF
    SET DEFAULT TO &MBACKUP
MDISKSPACE = DISKSPACE()
DO WHILE MDISKSPACE < MLOCATION
   ? "Not enough space on diskette in drive " + MBACKDRV
   ? "Please replace with another diskette and press any key to continue"
   WAIT " "
   MDISKSPACE = DISKSPACE()
ENDDO
SET DEFAULT TO &MDRIVE
COPY FILE "/PAVEDB\FILES\LOCATION.DBF TO &MBACKDVR\LOCATION.DBF

? "Please wait. Backing up IAYER files . . ."
IF FILE("\PAVEDB\FILES\BACKUP\IAYER.DBF")
   DELE FILE \PAVEDB\FILES\BACKUP\IAYER.DBF
ENDIF
COPY FILE \PAVEDB\FILES\IAYER.DBF TO \PAVEDB\FILES\BACKUP\IAYER.DBF
SET DEFAULT TO &MBACKDVR
MDISKSPACE = DISKSPACE()
DO WHILE MDISKSPACE < MAYER
   ? "Not enough space on diskette in drive " + MBACKDRV
   ? "Please replace with another diskette and press any key to continue"
   WAIT " "
   MDISKSPACE = DISKSPACE()
ENDDO
SET DEFAULT TO &MDRIVE
COPY FILE \PAVEDB\FILES\IAYER.DBF TO &MBACKDVR\IAYER.DBF

? "Please wait. Backing up IAYTHICK files . . ."
IF FILE("\PAVEDB\FILES\BACKUP\IAYTHICK.DBF")
   DELE FILE \PAVEDB\FILES\BACKUP\IAYTHICK.DBF
ENDIF
COPY FILE \PAVEDB\FILES\IAYTHICK.DBF TO \PAVEDB\FILES\BACKUP\IAYTHICK.DBF
SET DEFAULT TO &MBACKDVR
MDISKSPACE = DISKSPACE()
DO WHILE MDISKSPACE < MLAYTHICK
   ? "Not enough space on diskette in drive " + MBACKDRV
   ? "Please replace with another diskette and press any key to continue"
   WAIT " "
   MDISKSPACE = DISKSPACE()
ENDDO
SET DEFAULT TO &MDRIVE
COPY FILE \PAVEDB\FILES\IAYTHICK.DBF TO &MBACKDVR\IAYTHICK.DBF

? "Please wait. Backing up GEOSHIC files . . ."
IF FILE("\PAVEDB\FILES\BACKUP\GEOSHIC.DBF")
   DELE FILE \PAVEDB\FILES\BACKUP\GEOSHIC.DBF
ENDIF
COPY FILE \PAVEDB\FILES\GEOSHIC.DBF TO \PAVEDB\FILES\BACKUP\GEOSHIC.DBF
SET DEFAULT TO &MBACKDVR
MDISKSPACE = DISKSPACE()
DO WHILE MDISKSPACE < MGEOSHIC
   ? "Not enough space on diskette in drive " + MBACKDRV
   ? "Please replace with another diskette and press any key to continue"
   WAIT " "
   MDISKSPACE = DISKSPACE()
ENDDO
SET DEFAULT TO &Mdrive
COPY FILE \PAVEDB\FILES\GEOSHO.DBF TO &Mbackdrv\GEOSHO.DBF

? "Please wait. Backing up SURFACE files . . ."
IF FILE('\PAVEDB\FILES\BACKUP\SURFACE.DBF')
   DELETE FILE \PAVEDB\FILES\BACKUP\SURFACE.DBF
ENDIF
COPY FILE \PAVEDB\FILES\SURFACE.DBF TO \PAVEDB\FILES\BACKUP\SURFACE.DBF
SET DEFAULT TO &Mbackdrv
MDISKSPACE = DISKSPACE()
DO WHILE MDISKSPACE < MSURFACE
   ? "Not enough space on diskette in drive " + Mbackdrv
   ? "Please replace with another diskette and press any key to continue"
   WAIT " "
   MDISKSPACE = DISKSPACE()
ENDDO
SET DEFAULT TO &Mdrive
COPY FILE \PAVEDB\FILES\SURFACE.DBF TO &Mbackdrv\SURFACE.DBF

? "Please wait. Backing up SUBGRADE files . . ."
IF FILE('\PAVEDB\FILES\BACKUP\SUBGRADE.DBF')
   DELETE FILE \PAVEDB\FILES\BACKUP\SUBGRADE.DBF
ENDIF
COPY FILE \PAVEDB\FILES\SUBGRADE.DBF TO \PAVEDB\FILES\BACKUP\SUBGRADE.DBF
SET DEFAULT TO &Mbackdrv
MDISKSPACE = DISKSPACE()
DO WHILE MDISKSPACE < MSUBGRADE
   ? "Not enough space on diskette in drive " + Mbackdrv
   ? "Please replace with another diskette and press any key to continue"
   WAIT " "
   MDISKSPACE = DISKSPACE()
ENDDO
SET DEFAULT TO &Mdrive
COPY FILE \PAVEDB\FILES\SUBGRADE.DBF TO &Mbackdrv\SUBGRADE.DBF

? "Please wait. Backing up VISUAL files . . ."
IF FILE('\PAVEDB\FILES\BACKUP\VISUAL.DBF')
   DELETE FILE \PAVEDB\FILES\BACKUP\VISUAL.DBF
ENDIF
COPY FILE \PAVEDB\FILES\VISUAL.DBF TO \PAVEDB\FILES\BACKUP\VISUAL.DBF
SET DEFAULT TO &Mbackdrv
MDISKSPACE = DISKSPACE()
DO WHILE MDISKSPACE < MVISUAL
   ? "Not enough space on diskette in drive " + Mbackdrv
   ? "Please replace with another diskette and press any key to continue"
   WAIT " "
   MDISKSPACE = DISKSPACE()
ENDDO
SET DEFAULT TO &Mdrive
COPY FILE \PAVEDB\FILES\VISUAL.DBF TO &Mbackdrv\VISUAL.DBF

? "Please wait. Backing up DYNAFLEXT files . . ."
IF FILE('\PAVEDB\FILES\BACKUP\DYNAFLD.DBF')
DELETE FILE \PAVEDB\FILES\BACKUP\DYNAFL1D.DBF
ENDIF
COPY FILE \PAVEDB\FILES\DYNAFL1D.DBF TO \PAVEDB\FILES\BACKUP\DYNAFL1D.DBF
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
DO WHILE MDISKSPACE < MDYNAFL1D
  ? "Not enough space on diskette in drive " + MBACKDRV
  ? "Please replace with another diskette and press any key to continue"
  WAIT " "
  MDISKSPACE = DISKSPACE()
ENDDO
SET DEFAULT TO &MDFILE
COPY FILE \PAVEDB\FILES\DYNAFL1D.DBF TO &MBACKDRV\DYNAFL1D.DBF

? "Please wait. Backing up SI files . . ."
IF FILE('\PAVEDB\FILES\BACKUP\SI.DBF')
  DELETE FILE '\PAVEDB\FILES\BACKUP\SI.DBF'
ENDIF
COPY FILE \PAVEDB\FILES\SI.DBF TO \PAVEDB\FILES\BACKUP\SI.DBF
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
DO WHILE MDISKSPACE < MSI
  ? "Not enough space on diskette in drive " + MBACKDRV
  ? "Please replace with another diskette and press any key to continue"
  WAIT " "
  MDISKSPACE = DISKSPACE()
ENDDO
SET DEFAULT TO &MDFILE
COPY FILE \PAVEDB\FILES\SI.DBF TO &MBACKDRV\SI.DBF

? "Please wait. Backing up TRAFFIC files . . ."
IF FILE('\PAVEDB\FILES\BACKUP\TRAFFIC.DBF')
  DELETE FILE '\PAVEDB\FILES\BACKUP\TRAFFIC.DBF'
ENDIF
COPY FILE \PAVEDB\FILES\TRAFFIC.DBF TO \PAVEDB\FILES\BACKUP\TRAFFIC.DBF
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
DO WHILE MDISKSPACE < MIRAFFIC
  ? "Not enough space on diskette in drive " + MBACKDRV
  ? "Please replace with another diskette and press any key to continue"
  WAIT " "
  MDISKSPACE = DISKSPACE()
ENDDO
SET DEFAULT TO &MDFILE
COPY FILE \PAVEDB\FILES\TRAFFIC.DBF TO &MBACKDRV\TRAFFIC.DBF

? "Please wait. Backing up WEATHER files . . ."
IF FILE('\PAVEDB\FILES\BACKUP\WEATHER.DBF')
  DELETE FILE '\PAVEDB\FILES\BACKUP\WEATHER.DBF'
ENDIF
COPY FILE \PAVEDB\FILES\WEATHER.DBF TO \PAVEDB\FILES\BACKUP\WEATHER.DBF
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
DO WHILE MDISKSPACE < MWEATHER
? "Not enough space on diskette in drive " + MBACKDRV
? "Please replace with another diskette and press any key to continue"
WAIT " "
MDISKSPACE = DISKSPACE()
ENDDO
SET DEFAULT TO &MDRIVE
COPY FILE \PAVEDB\FILES\WEATHER.DBF TO &MBACKDRV\WEATHER.DBF

? "Please wait. Backing up FALWIGHT files . . ."
IF FILE('\PAVEDB\FILES\BACKUP\FALWIGHT.DBF')
  DELETE FILE '\PAVEDB\FILES\BACKUP\FALWIGHT.DBF'
ENDIF
COPY FILE \PAVEDB\FILES\FALWIGHT.DBF TO \PAVEDB\FILES\BACKUP\FALWIGHT.DBF
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
DO WHILE MDISKSPACE < MFALWIGHT
  ? "Not enough space on diskette in drive " + MBACKDRV
  ? "Please replace with another diskette and press any key to continue"
  WAIT " "
  MDISKSPACE = DISKSPACE()
ENDDO
SET DEFAULT TO &MDRIVE
COPY FILE \PAVEDB\FILES\FALWIGHT.DBF TO &MBACKDRV\FALWIGHT.DBF

? "Please wait. Backing up ENV files . . ."
IF FILE('\PAVEDB\FILES\BACKUP\ENV.DBF')
  DELETE FILE '\PAVEDB\FILES\BACKUP\ENV.DBF'
ENDIF
COPY FILE \PAVEDB\FILES\ENV.DBF TO \PAVEDB\FILES\BACKUP\ENV.DBF
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
DO WHILE MDISKSPACE < MENV
  ? "Not enough space on diskette in drive " + MBACKDRV
  ? "Please replace with another diskette and press any key to continue"
  WAIT " "
  MDISKSPACE = DISKSPACE()
ENDDO
SET DEFAULT TO &MDRIVE
COPY FILE \PAVEDB\FILES\ENV.DBF TO &MBACKDRV\ENV.DBF

? "Please wait. Backing up SKID files . . ."
IF FILE('\PAVEDB\FILES\BACKUP\SKID.DBF')
  DELETE FILE '\PAVEDB\FILES\BACKUP\SKID.DBF'
ENDIF
COPY FILE \PAVEDB\FILES\SKID.DBF TO \PAVEDB\FILES\BACKUP\SKID.DBF
SET DEFAULT TO &MBACKDRV
MDISKSPACE = DISKSPACE()
DO WHILE MDISKSPACE < MSKID
  ? "Not enough space on diskette in drive " + MBACKDRV
  ? "Please replace with another diskette and press any key to continue"
  WAIT " "
  MDISKSPACE = DISKSPACE()
ENDDO
SET DEFAULT TO &MDRIVE
COPY FILE \PAVEDB\FILES\SKID.DBF TO &MBACKDRV\SKID.DBF

ENDIF
SET TYPE TO 20
SET SAFETY OFF
CLOSE DATABASES
CLEAR
RETURN TO MASTER
APPENDIX A
County Name Table

File Name: CNTYTB.L DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTYNUM</td>
<td>*</td>
<td>3N</td>
<td>County Number</td>
</tr>
<tr>
<td>CNTYNAME</td>
<td></td>
<td>13C</td>
<td>County Name</td>
</tr>
</tbody>
</table>

The table is sorted on the key field.
**File Layout**

File Name: DISTRESS.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID NO</td>
<td>*</td>
<td>4N</td>
<td></td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>YEAR</td>
<td>*</td>
<td>2N</td>
<td></td>
<td>Year</td>
</tr>
<tr>
<td>A18KEAL</td>
<td>1N</td>
<td>11N</td>
<td>T</td>
<td>Annual Cumulative 18 Keal one way</td>
</tr>
<tr>
<td>RUTT</td>
<td>1N</td>
<td>V</td>
<td></td>
<td>Rutting Area</td>
</tr>
<tr>
<td>ALLGCR</td>
<td>1N</td>
<td>V</td>
<td></td>
<td>Alligator Cracking Area</td>
</tr>
<tr>
<td>PATCH</td>
<td>1N</td>
<td>V</td>
<td></td>
<td>Patch</td>
</tr>
<tr>
<td>SIMEAN</td>
<td>7.5N</td>
<td>SI</td>
<td></td>
<td>Serviceability Index Mean</td>
</tr>
<tr>
<td>CONYEAR</td>
<td>2N</td>
<td>LI</td>
<td></td>
<td>Year of Last Major Construction</td>
</tr>
<tr>
<td>CONMON</td>
<td>2N</td>
<td>LI</td>
<td></td>
<td>Month of Last Major Construction</td>
</tr>
<tr>
<td>PAVETYP</td>
<td>2N</td>
<td>G</td>
<td></td>
<td>Pavement Type</td>
</tr>
<tr>
<td>OVRYEAR</td>
<td>2N</td>
<td>LI</td>
<td></td>
<td>Year of Last HMAC Overlay</td>
</tr>
<tr>
<td>OVRMON</td>
<td>2N</td>
<td>LI</td>
<td></td>
<td>Month of Last HMAC Overlay</td>
</tr>
</tbody>
</table>

**File Source**

T - TRAFFIC FILE
V - VISUAL FILE
LI - LAYER IDENTIFICATION FILE (LAYER)
G - GEOMETRIC & SHOULDER INFORMATION FILE (GBOSHO)
SI - SERVICEABILITY INDEX FILE
**District Temperature Constant Table**

File Name: DISTTEMP.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISTRICT</td>
<td>*</td>
<td>2N</td>
<td>District Number</td>
</tr>
<tr>
<td>TEMPCONS</td>
<td></td>
<td>2N</td>
<td>District Temperature Constant</td>
</tr>
</tbody>
</table>

The table is sorted on the key field.
## Monitoring Data

**Dynaflect Measurement File**

File Name: DYNAFLID.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>STRUCNUM</td>
<td>*</td>
<td>2N</td>
<td>Structure Number</td>
</tr>
<tr>
<td>LAYNUM</td>
<td>*</td>
<td>2N</td>
<td>Layer Identification Number</td>
</tr>
<tr>
<td>YEAR</td>
<td>*</td>
<td>2N</td>
<td>Year</td>
</tr>
<tr>
<td>MONTH</td>
<td>*</td>
<td>2N</td>
<td>Month</td>
</tr>
<tr>
<td>DAY</td>
<td></td>
<td>2N</td>
<td>Day</td>
</tr>
<tr>
<td>STATION</td>
<td>*</td>
<td>2N</td>
<td>Station</td>
</tr>
<tr>
<td>SENS1RD</td>
<td>5.3N</td>
<td></td>
<td>Sensor 1 Reading</td>
</tr>
<tr>
<td>SENS2RD</td>
<td>5.3N</td>
<td></td>
<td>Sensor 2 Reading</td>
</tr>
<tr>
<td>SENS3RD</td>
<td>5.3N</td>
<td></td>
<td>Sensor 3 Reading</td>
</tr>
<tr>
<td>SENS4RD</td>
<td>5.3N</td>
<td></td>
<td>Sensor 4 Reading</td>
</tr>
<tr>
<td>SENS5RD</td>
<td>5.3N</td>
<td></td>
<td>Sensor 5 Reading</td>
</tr>
</tbody>
</table>
Environmental Data

Environment Measurement File

File Name: ENV.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTYNUM</td>
<td>*</td>
<td>3N</td>
<td>County Number</td>
</tr>
<tr>
<td>THORNIN</td>
<td></td>
<td>S10.3N</td>
<td>Thornthwaite Index - Mean</td>
</tr>
<tr>
<td>THORNSD</td>
<td></td>
<td>S10.3N</td>
<td>Thornthwaite Index - Standard Deviation</td>
</tr>
<tr>
<td>THORNYRS</td>
<td></td>
<td>2N</td>
<td>Thornthwaite Index - No. Years Averaged</td>
</tr>
</tbody>
</table>
**Monitoring Data**

**Falling Weight SSI File**

File Name: FALILWGH.T DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>STRUCNUM</td>
<td>*</td>
<td>2N</td>
<td>Structure Number</td>
</tr>
<tr>
<td>LAYNUM</td>
<td>*</td>
<td>2N</td>
<td>Layer Number</td>
</tr>
<tr>
<td>YEAR</td>
<td>*</td>
<td>2N</td>
<td>Year</td>
</tr>
<tr>
<td>MONTH</td>
<td>*</td>
<td>2N</td>
<td>Month</td>
</tr>
<tr>
<td>DAY</td>
<td>*</td>
<td>2N</td>
<td>Day</td>
</tr>
<tr>
<td>RWSIAVG</td>
<td>4.1N</td>
<td></td>
<td>Average SSI for the Roadway</td>
</tr>
<tr>
<td>SSITEMP</td>
<td>3N</td>
<td></td>
<td>Temperature For the Roadway</td>
</tr>
<tr>
<td>SSIGP11</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 1 Geophone no. 1</td>
</tr>
<tr>
<td>SSIGP12</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 1 Geophone no. 2</td>
</tr>
<tr>
<td>SSIGP13</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 1 Geophone no. 3</td>
</tr>
<tr>
<td>SSIGP14</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 1 Geophone no. 4</td>
</tr>
<tr>
<td>SSIGP15</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 1 Geophone no. 5</td>
</tr>
<tr>
<td>SSIGP16</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 1 Geophone no. 6</td>
</tr>
<tr>
<td>SSIGP17</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 1 Geophone no. 7</td>
</tr>
<tr>
<td>SSIGP21</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 2 Geophone no. 1</td>
</tr>
<tr>
<td>SSIGP22</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 2 Geophone no. 2</td>
</tr>
<tr>
<td>SSIGP23</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 2 Geophone no. 3</td>
</tr>
<tr>
<td>SSIGP24</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 2 Geophone no. 4</td>
</tr>
<tr>
<td>SSIGP25</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 2 Geophone no. 5</td>
</tr>
<tr>
<td>SSIGP26</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 2 Geophone no. 6</td>
</tr>
<tr>
<td>SSIGP27</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 2 Geophone no. 7</td>
</tr>
<tr>
<td>SSIGP31</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 3 Geophone no. 1</td>
</tr>
<tr>
<td>SSIGP32</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 3 Geophone no. 2</td>
</tr>
<tr>
<td>SSIGP33</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 3 Geophone no. 3</td>
</tr>
<tr>
<td>SSIGP34</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 3 Geophone no. 4</td>
</tr>
<tr>
<td>SSIGP35</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 3 Geophone no. 5</td>
</tr>
<tr>
<td>SSIGP36</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 3 Geophone no. 6</td>
</tr>
<tr>
<td>SSIGP37</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 3 Geophone no. 7</td>
</tr>
<tr>
<td>SSIGP41</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 4 Geophone no. 1</td>
</tr>
<tr>
<td>SSIGP42</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 4 Geophone no. 2</td>
</tr>
<tr>
<td>SSIGP43</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 4 Geophone no. 3</td>
</tr>
<tr>
<td>SSIGP44</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 4 Geophone no. 4</td>
</tr>
<tr>
<td>SSIGP45</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 4 Geophone no. 5</td>
</tr>
</tbody>
</table>

628
Monitoring Data

Falling Weight SSI File (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSIGP46</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 4 Geophone no. 6</td>
</tr>
<tr>
<td>SSIGP47</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 4 Geophone no. 7</td>
</tr>
<tr>
<td>SSIGP51</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 5 Geophone no. 1</td>
</tr>
<tr>
<td>SSIGP52</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 5 Geophone no. 2</td>
</tr>
<tr>
<td>SSIGP53</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 5 Geophone no. 3</td>
</tr>
<tr>
<td>SSIGP54</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 5 Geophone no. 4</td>
</tr>
<tr>
<td>SSIGP55</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 5 Geophone no. 5</td>
</tr>
<tr>
<td>SSIGP56</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 5 Geophone no. 6</td>
</tr>
<tr>
<td>SSIGP57</td>
<td>5.2N</td>
<td></td>
<td>SSI Reading 5 Geophone no. 7</td>
</tr>
</tbody>
</table>
Functional Classification Table

File Name: FUNCLTBL.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE</td>
<td>*</td>
<td>1N</td>
<td>Functional Classification Code</td>
</tr>
<tr>
<td>DESCRPT</td>
<td></td>
<td>40C</td>
<td>Description of Functional Classification</td>
</tr>
</tbody>
</table>

The table is sorted on the key field.
Inventory Data

Geometric and Shoulder Information File

File Name: GEOSHO.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>STRUCNUM</td>
<td>*</td>
<td>2N</td>
<td>Structure Number</td>
</tr>
<tr>
<td>PAVETYP</td>
<td></td>
<td>2N</td>
<td>Type of Pavement</td>
</tr>
<tr>
<td>LANEMID</td>
<td></td>
<td>4.1N</td>
<td>Lane Width (Feet)</td>
</tr>
<tr>
<td>OUTSHOWD</td>
<td></td>
<td>4.1N</td>
<td>Outside Shoulder Width (Feet)</td>
</tr>
<tr>
<td>SHOSFTYP</td>
<td></td>
<td>1N</td>
<td>Shoulder Surface Type</td>
</tr>
<tr>
<td>SHOBSTYP</td>
<td></td>
<td>2N</td>
<td>Shoulder Base Type</td>
</tr>
<tr>
<td>SHOSFTHK</td>
<td></td>
<td>5.2N</td>
<td>Shoulder Surface Thickness (Inch)</td>
</tr>
<tr>
<td>SHOBSTHK</td>
<td></td>
<td>5.2N</td>
<td>Shoulder Base Thickness (Inches)</td>
</tr>
<tr>
<td>WIDENFLG</td>
<td></td>
<td>1N</td>
<td>Flag for Type Of Widening</td>
</tr>
</tbody>
</table>
## Inventory Data

### Layer Identification File

**File Name:** LAYER.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>STRUCNUM</td>
<td>*</td>
<td>2N</td>
<td>Structure Number</td>
</tr>
<tr>
<td>LAYNUM</td>
<td>*</td>
<td>2N</td>
<td>Layer Number</td>
</tr>
<tr>
<td>LAYDESC</td>
<td></td>
<td>2N</td>
<td>Layer Description</td>
</tr>
<tr>
<td>CENTTHK</td>
<td></td>
<td>5.2N</td>
<td>Layer Center Thickness</td>
</tr>
<tr>
<td>LAYMATCL</td>
<td></td>
<td>2N</td>
<td>Material Type Classification</td>
</tr>
<tr>
<td>JOBCMPYR</td>
<td></td>
<td>2N</td>
<td>Year Job Completed</td>
</tr>
<tr>
<td>JOBCMPMO</td>
<td></td>
<td>2N</td>
<td>Month Job Completed</td>
</tr>
<tr>
<td>WIDENLYR</td>
<td></td>
<td>2N</td>
<td>Year Layer Widened</td>
</tr>
<tr>
<td>WIDENLMO</td>
<td></td>
<td>2N</td>
<td>Month Layer Widened</td>
</tr>
</tbody>
</table>
Layer Description Table

File Name: LAYERBL.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE</td>
<td>*</td>
<td>2N</td>
<td>Layer Description Code</td>
</tr>
<tr>
<td>CODE_DESC</td>
<td></td>
<td>4C</td>
<td>Layer Description Abbreviation</td>
</tr>
<tr>
<td>CODE_LAYR</td>
<td></td>
<td>25C</td>
<td>Layer Description Explanation</td>
</tr>
</tbody>
</table>

The table is sorted on the key field.
**Inventory Data**

**Layer Thickness Across the Road File**

File Name: LAYTHICK.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>STRUCNUM</td>
<td>*</td>
<td>2N</td>
<td>Structure Number</td>
</tr>
<tr>
<td>LAYNUM</td>
<td>*</td>
<td>2N</td>
<td>Layer Number</td>
</tr>
<tr>
<td>FC3THK</td>
<td>5.2N</td>
<td></td>
<td>Thickness - 3rd Pos. from Center</td>
</tr>
<tr>
<td>FC2THK</td>
<td>5.2N</td>
<td></td>
<td>Thickness - 2nd Pos. from Center</td>
</tr>
<tr>
<td>FC1THK</td>
<td>5.2N</td>
<td></td>
<td>Thickness - 1st Pos. from Center</td>
</tr>
<tr>
<td>CENTTHK</td>
<td>5.2N</td>
<td></td>
<td>Thickness - Center</td>
</tr>
<tr>
<td>FC3DIS</td>
<td>4.1N</td>
<td></td>
<td>Distance from the Center/3rd Pos.</td>
</tr>
<tr>
<td>FC2DIS</td>
<td>4.1N</td>
<td></td>
<td>Distance from the Center/2nd Pos.</td>
</tr>
<tr>
<td>FC1DIS</td>
<td>4.1N</td>
<td></td>
<td>Distance from the Center/1st Pos.</td>
</tr>
</tbody>
</table>
Inventory Data

Location File

File Name: LOCATION.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>ACTIVFLAG</td>
<td></td>
<td>1L</td>
<td>Active/Inactive Flag</td>
</tr>
<tr>
<td>HWYDIST</td>
<td></td>
<td>2N</td>
<td>SDHPT Highway District</td>
</tr>
<tr>
<td>CNTYNUM</td>
<td></td>
<td>3N</td>
<td>County Number</td>
</tr>
<tr>
<td>HPMSSAM</td>
<td></td>
<td>12C</td>
<td>HPMS Sample Number</td>
</tr>
<tr>
<td>HPMSSSEC</td>
<td></td>
<td>1N</td>
<td>HPMS Section Subdivision</td>
</tr>
<tr>
<td>FUNCAS</td>
<td></td>
<td>2N</td>
<td>Functional Classification</td>
</tr>
<tr>
<td>HWYPREFX</td>
<td></td>
<td>2C</td>
<td>Highway Number Prefix</td>
</tr>
<tr>
<td>HWYNUM</td>
<td></td>
<td>4N</td>
<td>Highway Number</td>
</tr>
<tr>
<td>HWYSUFFX</td>
<td></td>
<td>1C</td>
<td>Highway Number Suffix</td>
</tr>
<tr>
<td>BEGMPST</td>
<td></td>
<td>3N</td>
<td>Beginning Milepoint</td>
</tr>
<tr>
<td>BDISSIGN</td>
<td></td>
<td>1C</td>
<td>Beginning Displacement Sign</td>
</tr>
<tr>
<td>BMPSTDIS</td>
<td></td>
<td>2N</td>
<td>Beginning Milepost Displacement</td>
</tr>
<tr>
<td>ENDMST</td>
<td></td>
<td>3N</td>
<td>Ending Milepost</td>
</tr>
<tr>
<td>EDISSIGN</td>
<td></td>
<td>1C</td>
<td>Ending Displacement Sign</td>
</tr>
<tr>
<td>EMPSTDIS</td>
<td></td>
<td>2N</td>
<td>Ending Milepost Displacement</td>
</tr>
<tr>
<td>LANEID</td>
<td></td>
<td>1C</td>
<td>Lane Identification (R,L)</td>
</tr>
<tr>
<td>CONTROL</td>
<td></td>
<td>4N</td>
<td>Control</td>
</tr>
<tr>
<td>SECTION</td>
<td></td>
<td>2N</td>
<td>Section</td>
</tr>
<tr>
<td>BEGIN</td>
<td></td>
<td>6.3N</td>
<td>Beginning Milepoint</td>
</tr>
<tr>
<td>END</td>
<td></td>
<td>6.3N</td>
<td>Ending Milepoint</td>
</tr>
<tr>
<td>MPMIMO</td>
<td></td>
<td>2N</td>
<td>Milepoint Month</td>
</tr>
<tr>
<td>MPRTYR</td>
<td></td>
<td>2N</td>
<td>Milepoint Year</td>
</tr>
<tr>
<td>NUMLANES</td>
<td></td>
<td>2N</td>
<td>Number of Lanes</td>
</tr>
<tr>
<td>PREVSID</td>
<td></td>
<td>4N</td>
<td>Pointer to the Previous SID Number for this</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Location</td>
</tr>
<tr>
<td>NEXTISID</td>
<td></td>
<td>4N</td>
<td>Pointer to the Next SID Number for this</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Location</td>
</tr>
<tr>
<td>INACTYR</td>
<td></td>
<td>2N</td>
<td>Inactive Year</td>
</tr>
<tr>
<td>INACTIMO</td>
<td></td>
<td>2N</td>
<td>Inactive Month</td>
</tr>
<tr>
<td>COMMENT</td>
<td></td>
<td>40C</td>
<td>Comments on SID Number</td>
</tr>
</tbody>
</table>
Material Type Table

File Name: MATLITBL.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATCODE</td>
<td>*</td>
<td>2N</td>
<td>Code for Material Classification Type</td>
</tr>
<tr>
<td>MATDESC</td>
<td></td>
<td>20C</td>
<td>Description of Material Type</td>
</tr>
<tr>
<td>MATSHORT</td>
<td></td>
<td>4C</td>
<td>Short Form for Material Classification Type</td>
</tr>
<tr>
<td>LAYRDES</td>
<td></td>
<td>2C</td>
<td>Type of Layer (surface, subbase, base, subgrade)</td>
</tr>
</tbody>
</table>

The table is sorted on the key field.
File Name: MODEL.DBF

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>HWTYPE</td>
<td>2C</td>
<td></td>
<td>Highway Type</td>
</tr>
<tr>
<td>HWYDIST</td>
<td>2N</td>
<td></td>
<td>District</td>
</tr>
<tr>
<td>CNTYNUM</td>
<td>3N</td>
<td></td>
<td>County Number</td>
</tr>
<tr>
<td>PAVETYP</td>
<td>2N</td>
<td></td>
<td>Pavement Type</td>
</tr>
<tr>
<td>ALLG RHO</td>
<td>7.4N</td>
<td></td>
<td>Alligator Rho Value</td>
</tr>
<tr>
<td>ALLG BETA</td>
<td>7.4N</td>
<td></td>
<td>Alligator Beta Value</td>
</tr>
<tr>
<td>RUIT RHO</td>
<td>7.4N</td>
<td></td>
<td>Rutting Rho Value</td>
</tr>
<tr>
<td>RUIT BETA</td>
<td>7.4N</td>
<td></td>
<td>Rutting Beta Value</td>
</tr>
<tr>
<td>PSI RHO</td>
<td>7.4N</td>
<td></td>
<td>PSI Rho Value</td>
</tr>
<tr>
<td>PSI_BETA</td>
<td>7.4N</td>
<td></td>
<td>PSI Beta Value</td>
</tr>
<tr>
<td>BASETYP</td>
<td>2N</td>
<td></td>
<td>Base Type</td>
</tr>
<tr>
<td>BASETHCK</td>
<td>7.2N</td>
<td></td>
<td>Base Thickness</td>
</tr>
<tr>
<td>TFCMN</td>
<td>10.3N</td>
<td></td>
<td>Total Freeze Thaw Cycle Mean</td>
</tr>
<tr>
<td>PECMN</td>
<td>10.3N</td>
<td></td>
<td>Inches of Precipitation</td>
</tr>
<tr>
<td>THORNMN</td>
<td>10.3N</td>
<td></td>
<td>Thornthwaite Index Mean</td>
</tr>
<tr>
<td>CURADT</td>
<td>6N</td>
<td></td>
<td>Current Average Daily Traffic</td>
</tr>
<tr>
<td>SUBGRDTY</td>
<td>2N</td>
<td></td>
<td>Subgrade Type</td>
</tr>
<tr>
<td>PRSV200</td>
<td>4.1N</td>
<td></td>
<td>Percent Passing #200 Sieve</td>
</tr>
<tr>
<td>LIQLM</td>
<td>4.1N</td>
<td></td>
<td>Liquid Limit</td>
</tr>
<tr>
<td>SHOPTYP</td>
<td>1N</td>
<td></td>
<td>Shoulder Surface Type</td>
</tr>
<tr>
<td>SURFTHCK</td>
<td>7.2N</td>
<td></td>
<td>Surface Thickness</td>
</tr>
<tr>
<td>ASAPPLRT</td>
<td>4.2N</td>
<td></td>
<td>Aggregate Application Rate</td>
</tr>
<tr>
<td>TEMPCONS</td>
<td>2N</td>
<td></td>
<td>District Temperature Constant</td>
</tr>
<tr>
<td>DYNALC1</td>
<td>5.3N</td>
<td></td>
<td>Average Dynaflect Measurement - Sensor 1</td>
</tr>
<tr>
<td>DYNALC2</td>
<td>5.3N</td>
<td></td>
<td>Average Dynaflect Measurement - Sensor 2</td>
</tr>
<tr>
<td>DYNALC3</td>
<td>5.3N</td>
<td></td>
<td>Average Dynaflect Measurement - Sensor 3</td>
</tr>
<tr>
<td>DYNALC4</td>
<td>5.3N</td>
<td></td>
<td>Average Dynaflect Measurement - Sensor 4</td>
</tr>
<tr>
<td>DYNALC5</td>
<td>5.3N</td>
<td></td>
<td>Average Dynaflect Measurement - Sensor 5</td>
</tr>
<tr>
<td>SSIGP1</td>
<td>5.2N</td>
<td></td>
<td>Average SSI for Geophone 1</td>
</tr>
<tr>
<td>SSIGP2</td>
<td>5.3N</td>
<td></td>
<td>Average SSI for Geophone 2</td>
</tr>
<tr>
<td>SSIGP3</td>
<td>5.3N</td>
<td></td>
<td>Average SSI for Geophone 3</td>
</tr>
<tr>
<td>SSIGP4</td>
<td>5.3N</td>
<td></td>
<td>Average SSI for Geophone 4</td>
</tr>
<tr>
<td>SSIGP5</td>
<td>5.3N</td>
<td></td>
<td>Average SSI for Geophone 5</td>
</tr>
</tbody>
</table>
Pavement Type Table

File Name: PAVETYPE.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAVECODE</td>
<td>*</td>
<td>2N</td>
<td>Type of Pavement Code</td>
</tr>
<tr>
<td>BASETYPE</td>
<td></td>
<td>25C</td>
<td>Type of Base</td>
</tr>
<tr>
<td>BASETHK</td>
<td></td>
<td>17C</td>
<td>Surface Thickness</td>
</tr>
<tr>
<td>BASESEAL</td>
<td></td>
<td>12C</td>
<td>Surface Seal</td>
</tr>
</tbody>
</table>

The table is sorted on the key field.
# Monitoring Data

Serviceability Index File

File Name: SI.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>STRUCNUM</td>
<td>*</td>
<td>2N</td>
<td>Structure Number</td>
</tr>
<tr>
<td>LAYNUM</td>
<td>*</td>
<td>2N</td>
<td>Layer Identification Number</td>
</tr>
<tr>
<td>YEAR</td>
<td>*</td>
<td>2N</td>
<td>Identifying Year</td>
</tr>
<tr>
<td>ACTMONTH</td>
<td></td>
<td>2N</td>
<td>Actual Month of Measurement</td>
</tr>
<tr>
<td>ACTDAY</td>
<td></td>
<td>2N</td>
<td>Actual Day of Measurement</td>
</tr>
<tr>
<td>ACTYEAR</td>
<td></td>
<td>2N</td>
<td>Actual Year of Measurement</td>
</tr>
<tr>
<td>SICOUNT</td>
<td></td>
<td>3N</td>
<td>Serviceability Index - Count of Observation</td>
</tr>
<tr>
<td>SIMEAN</td>
<td></td>
<td>7.5N</td>
<td>Serviceability Index - Mean</td>
</tr>
<tr>
<td>SISD</td>
<td></td>
<td>7.5N</td>
<td>Serviceability Index - Standard Deviation</td>
</tr>
<tr>
<td>SILOWVAL</td>
<td></td>
<td>3.1N</td>
<td>Serviceability Index - Low Value</td>
</tr>
<tr>
<td>SIHIVAL</td>
<td></td>
<td>3.1N</td>
<td>Serviceability Index - High Value</td>
</tr>
</tbody>
</table>
# Monitoring Data

**Skid Measuring File**

File Name: SKID.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID_NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>STRUCNUM</td>
<td>*</td>
<td>2N</td>
<td>Structure number</td>
</tr>
<tr>
<td>LAYNUM</td>
<td>*</td>
<td>2N</td>
<td>Layer number</td>
</tr>
<tr>
<td>YEAR</td>
<td>*</td>
<td>2N</td>
<td>Year of Measurement</td>
</tr>
<tr>
<td>MONTH</td>
<td>*</td>
<td>2N</td>
<td>Month of Measurement</td>
</tr>
<tr>
<td>SKIDNUMM</td>
<td></td>
<td>2N</td>
<td>Skid Number - Mean</td>
</tr>
<tr>
<td>SKIDNUMH</td>
<td></td>
<td>2N</td>
<td>Skid Number - High</td>
</tr>
<tr>
<td>SKIDNUML</td>
<td></td>
<td>2N</td>
<td>Skid Number - Low</td>
</tr>
</tbody>
</table>
Inventory Data
Subgrade File

File Name: SUBGRADE.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>STRUCNUM</td>
<td>*</td>
<td>2N</td>
<td>Structure Number</td>
</tr>
<tr>
<td>LAYNUM</td>
<td>*</td>
<td>2N</td>
<td>Layer Number</td>
</tr>
<tr>
<td>PPSV200</td>
<td></td>
<td>4.1N</td>
<td>Percent Passing NO. 200 Sieve</td>
</tr>
<tr>
<td>PLASTIX</td>
<td></td>
<td>4.1N</td>
<td>Plasticity Index</td>
</tr>
<tr>
<td>LIQLIM</td>
<td></td>
<td>4.1N</td>
<td>Liquid Limit</td>
</tr>
<tr>
<td>TXTRIAXL</td>
<td></td>
<td>3.1N</td>
<td>Texas Triaxial Class</td>
</tr>
<tr>
<td>PERMIX</td>
<td></td>
<td>5.2N</td>
<td>Permeability Index</td>
</tr>
</tbody>
</table>
Inventory Data

Surface File

File Name: SURFACE.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID_NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>STRUCNUM</td>
<td>*</td>
<td>2N</td>
<td>Structure Number</td>
</tr>
<tr>
<td>LAYNUM</td>
<td>*</td>
<td>2N</td>
<td>Layer Number</td>
</tr>
<tr>
<td>AGAPPIRT</td>
<td></td>
<td>3N</td>
<td>Aggregate Application Rate</td>
</tr>
<tr>
<td>ADMXYP</td>
<td></td>
<td>12C</td>
<td>Type Admixture</td>
</tr>
<tr>
<td>ADMXPER</td>
<td></td>
<td>5.2N</td>
<td>Percent Admixture</td>
</tr>
<tr>
<td>ASAPPIRT</td>
<td></td>
<td>4.2N</td>
<td>Asphalt Application Rate</td>
</tr>
</tbody>
</table>
## Monitoring Data

### Traffic Measurement File

**File Name**: TRAFFIC.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID_NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>YEAR</td>
<td>*</td>
<td>2N</td>
<td>Year of Measurement</td>
</tr>
<tr>
<td>AADT1WAY</td>
<td></td>
<td>6N</td>
<td>Annual Average Daily Traffic</td>
</tr>
<tr>
<td>A18KEAL</td>
<td></td>
<td>9N</td>
<td>Annual Cumulative 18KEAL one way</td>
</tr>
<tr>
<td>PCITRK</td>
<td></td>
<td>4.1N</td>
<td>Percent Trucks</td>
</tr>
</tbody>
</table>

**NOTE**: 18KEAL is 18000 lbs. equivalent axle load. This figure is calculated using the AASHTO Equivalency factors which convert any weighted truck axle to a number of equivalent of 18000 lbs. single axles. The basics for the equivalency is the observed pavement damage done by different axle loads at the AASHTO Road Test (1959-1960).
## Monitoring Data
### Visual File

**File Name:** VISUAL.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID_NO</td>
<td>*</td>
<td>4N</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>STRUCNUM</td>
<td>*</td>
<td>2N</td>
<td>Structure Number</td>
</tr>
<tr>
<td>LAYNUM</td>
<td>*</td>
<td>2N</td>
<td>Layer Number</td>
</tr>
<tr>
<td>YEAR</td>
<td>*</td>
<td>2N</td>
<td>Identifying Year</td>
</tr>
<tr>
<td>ACTYEAR</td>
<td></td>
<td>2N</td>
<td>Actual Year of Survey</td>
</tr>
<tr>
<td>ACIMONTH</td>
<td></td>
<td>2N</td>
<td>Actual Month of Survey</td>
</tr>
<tr>
<td>RUTITSL</td>
<td></td>
<td>1N</td>
<td>Rutting - Slight - Area Rating</td>
</tr>
<tr>
<td>RUTITMD</td>
<td></td>
<td>1N</td>
<td>Rutting / Moderate - Area Rating</td>
</tr>
<tr>
<td>RUTITSV</td>
<td></td>
<td>1N</td>
<td>Rutting / Severe - Area Rating</td>
</tr>
<tr>
<td>BLKCRKSL</td>
<td></td>
<td>1N</td>
<td>Block Crack / Slight - Area Rating</td>
</tr>
<tr>
<td>BLKCRKMD</td>
<td></td>
<td>1N</td>
<td>Block Crack / Moderate - Area Rating</td>
</tr>
<tr>
<td>BLKCRKSV</td>
<td></td>
<td>1N</td>
<td>Block Crack / Severe - Area Rating</td>
</tr>
<tr>
<td>ALLGCRSLL</td>
<td></td>
<td>1N</td>
<td>Alligator Crack/Slight - Area Rating</td>
</tr>
<tr>
<td>ALLGCRM</td>
<td></td>
<td>1N</td>
<td>AlligatorCrack/Mod- Area Rating</td>
</tr>
<tr>
<td>ALLGCRSV</td>
<td></td>
<td>1N</td>
<td>AlligatorCrack/Severe - Area Rating</td>
</tr>
<tr>
<td>LONGCRSL</td>
<td></td>
<td>1N</td>
<td>Longitudinal Crack/Slight - Area Rating</td>
</tr>
<tr>
<td>LONGCRM</td>
<td></td>
<td>1N</td>
<td>Longitudinal Crack/Mod - Area Rating</td>
</tr>
<tr>
<td>LONGCRSV</td>
<td></td>
<td>1N</td>
<td>Longitudinal Crack/Severe - Area Rating</td>
</tr>
<tr>
<td>TRANCRLS</td>
<td></td>
<td>1N</td>
<td>Transverse Crack/Slight - Area Rating</td>
</tr>
<tr>
<td>TRANCRMD</td>
<td></td>
<td>1N</td>
<td>Transverse Crack/Mod - Area Rating</td>
</tr>
<tr>
<td>TRANCRSV</td>
<td></td>
<td>1N</td>
<td>Transverse Crack/Severe - Area Rating</td>
</tr>
<tr>
<td>SEALRCD</td>
<td></td>
<td>1N</td>
<td>Cracks - Sealing Code</td>
</tr>
<tr>
<td>PATCHGD</td>
<td></td>
<td>1N</td>
<td>Patching / Good - Area Rating</td>
</tr>
<tr>
<td>PATCHPR</td>
<td></td>
<td>1N</td>
<td>Patching / Fair - Area Rating</td>
</tr>
<tr>
<td>PATCHPM</td>
<td></td>
<td>1N</td>
<td>Patching / Poor - Area Rating</td>
</tr>
<tr>
<td>FAILMILE</td>
<td></td>
<td>1N</td>
<td>Failures per Mile Occur. Rating</td>
</tr>
<tr>
<td>PRS</td>
<td></td>
<td>3N</td>
<td>Pavement Rating Score</td>
</tr>
<tr>
<td>PESPVIRS</td>
<td>4.2N</td>
<td></td>
<td>PES Rating Score</td>
</tr>
<tr>
<td>UVURS</td>
<td>4.2N</td>
<td></td>
<td>Unweighted Visual Utility Rating Score</td>
</tr>
</tbody>
</table>
Environmental Data
Weather Measurement File

File Name: WEATHER.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTYNUM</td>
<td>*</td>
<td>3N</td>
<td>County Number</td>
</tr>
<tr>
<td>MONTH</td>
<td>*</td>
<td>2N</td>
<td>Month number</td>
</tr>
<tr>
<td>PRECYRS</td>
<td></td>
<td>2N</td>
<td>Precipitation</td>
</tr>
<tr>
<td>PRECMN</td>
<td></td>
<td>8.3N</td>
<td>Precipitation</td>
</tr>
<tr>
<td>PRECSD</td>
<td></td>
<td>8.3N</td>
<td>Precipitation</td>
</tr>
<tr>
<td>TFTCYRS</td>
<td></td>
<td>2N</td>
<td>Total Freeze Thaw Cycle</td>
</tr>
<tr>
<td>TFTCMN</td>
<td></td>
<td>8.3N</td>
<td>Total Freeze Thaw Cycle</td>
</tr>
<tr>
<td>TFTCSD</td>
<td></td>
<td>8.3N</td>
<td>Total Freeze Thaw Cycle</td>
</tr>
<tr>
<td>WFTCYRS</td>
<td></td>
<td>2N</td>
<td>Wet Freeze Thaw Cycle</td>
</tr>
<tr>
<td>WFTCMN</td>
<td></td>
<td>8.3N</td>
<td>Wet Freeze Thaw Cycle</td>
</tr>
<tr>
<td>WFTCSD</td>
<td></td>
<td>8.3N</td>
<td>Wet Freeze Thaw Cycle</td>
</tr>
<tr>
<td>MIMPYRS</td>
<td></td>
<td>2N</td>
<td>Maximum Temperature</td>
</tr>
<tr>
<td>MIMPMN</td>
<td></td>
<td>8.3N</td>
<td>Maximum Temperature</td>
</tr>
<tr>
<td>MIMPSD</td>
<td></td>
<td>8.3N</td>
<td>Maximum Temperature</td>
</tr>
<tr>
<td>AIMPYRS</td>
<td></td>
<td>2N</td>
<td>Averaged Temperature</td>
</tr>
<tr>
<td>AIMPMN</td>
<td></td>
<td>8.3N</td>
<td>Averaged Temperature</td>
</tr>
<tr>
<td>AIMPSD</td>
<td></td>
<td>8.3N</td>
<td>Averaged Temperature</td>
</tr>
</tbody>
</table>

645
Widening Flag Table

File Name: WIDENFLG.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Key</th>
<th>Size/Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WIDNCODE</td>
<td>1N</td>
<td></td>
<td>Code for Widening Flag</td>
</tr>
<tr>
<td>WIDNDESC</td>
<td>26C</td>
<td></td>
<td>Description for the Widening Flag Code</td>
</tr>
</tbody>
</table>

The table is sorted on the key field.
APPENDIX B
## TEXAS FLEXIBLE PAVEMENT DATABASE

### VARIABLE DESCRIPTION FOR ALL FILES

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Source File</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A18KEAL</td>
<td>T</td>
<td>Traffic Measurement File</td>
</tr>
<tr>
<td>AADTIWAY</td>
<td>T</td>
<td>Traffic Measurement File</td>
</tr>
<tr>
<td>ACTDAY</td>
<td>S</td>
<td>Actual Day of Measurement</td>
</tr>
<tr>
<td>ACTMONTH</td>
<td>S</td>
<td>Actual Month of Measurement</td>
</tr>
<tr>
<td>ACTVFLAG</td>
<td>L</td>
<td>Active/Inactive Flag</td>
</tr>
<tr>
<td>ACTYEAR</td>
<td>S</td>
<td>Actual Year of Measurement</td>
</tr>
<tr>
<td>ADXPER</td>
<td>R</td>
<td>Percent Admixture</td>
</tr>
<tr>
<td>ADXTPY</td>
<td>R</td>
<td>Type Admixture</td>
</tr>
<tr>
<td>AGAPPLRT</td>
<td>R</td>
<td>Aggregate Application Rate</td>
</tr>
<tr>
<td>AASAPPLRT</td>
<td>R</td>
<td>Asphalt Application Rate</td>
</tr>
<tr>
<td>BDISSIGN</td>
<td>L</td>
<td>Beginning Displacement Sign</td>
</tr>
<tr>
<td>BEGMPNT</td>
<td>L</td>
<td>Beginning Milepoint</td>
</tr>
<tr>
<td>BEGMPSST</td>
<td>L</td>
<td>Beginning Milepost</td>
</tr>
<tr>
<td>BMEPSSTDIS</td>
<td>L</td>
<td>Beginning Milepost Displacement</td>
</tr>
<tr>
<td>CENTTHK</td>
<td>H</td>
<td>Thickness - Center</td>
</tr>
<tr>
<td>CNTYNUM</td>
<td>LWE</td>
<td>County Number</td>
</tr>
<tr>
<td>CONTROL</td>
<td>L</td>
<td>Control</td>
</tr>
<tr>
<td>DAY</td>
<td>FD</td>
<td>Day</td>
</tr>
<tr>
<td>EDISSIGN</td>
<td>L</td>
<td>Ending Displacement Sign</td>
</tr>
<tr>
<td>EMPSSTDIS</td>
<td>L</td>
<td>Ending Milepost Displacement</td>
</tr>
<tr>
<td>ENDMPSNT</td>
<td>L</td>
<td>Ending Milepost</td>
</tr>
<tr>
<td>FC1DIS</td>
<td>H</td>
<td>Distance from the Center/1st Pos.</td>
</tr>
<tr>
<td>FC1THK</td>
<td>H</td>
<td>Thickness - 1st Pos. from Center</td>
</tr>
<tr>
<td>FC2DIS</td>
<td>H</td>
<td>Distance from the Center/2nd Pos.</td>
</tr>
<tr>
<td>FC2THK</td>
<td>H</td>
<td>Thickness - 2nd Pos. from Center</td>
</tr>
<tr>
<td>FC3DIS</td>
<td>H</td>
<td>Distance from the Center/3rd Pos.</td>
</tr>
<tr>
<td>FC3THK</td>
<td>H</td>
<td>Thickness - 3rd Pos. from Center</td>
</tr>
<tr>
<td>FUNCLAS</td>
<td>L</td>
<td>Functional Classification</td>
</tr>
<tr>
<td>HPMSSAM</td>
<td>L</td>
<td>HPMS Sample Number</td>
</tr>
<tr>
<td>HPMSSEC</td>
<td>L</td>
<td>HPMS Section Subdivision</td>
</tr>
<tr>
<td>HWYDIST</td>
<td>L</td>
<td>SDHPT Highway District</td>
</tr>
<tr>
<td>HWYNUM</td>
<td>L</td>
<td>Highway Number</td>
</tr>
<tr>
<td>HWYPREFIX</td>
<td>L</td>
<td>Highway Number Prefix</td>
</tr>
<tr>
<td>HWYSUFFIX</td>
<td>L</td>
<td>Highway Number Suffix</td>
</tr>
<tr>
<td>INACIMO</td>
<td>L</td>
<td>Inactive Month</td>
</tr>
<tr>
<td>INACIYR</td>
<td>L</td>
<td>Inactive Year</td>
</tr>
<tr>
<td>JOBCOMPNO</td>
<td>I</td>
<td>Month Job Completed</td>
</tr>
<tr>
<td>JOBCMFYR</td>
<td>I</td>
<td>Year Job Completed</td>
</tr>
<tr>
<td>LANEID</td>
<td>L</td>
<td>Lane Identification (R,L)</td>
</tr>
<tr>
<td>LANEWID</td>
<td>G</td>
<td>Lane Width (Feet)</td>
</tr>
<tr>
<td>LAYDESC</td>
<td>I</td>
<td>Layer Description</td>
</tr>
<tr>
<td>LAYMATCL</td>
<td>I</td>
<td>Material Type Classification</td>
</tr>
<tr>
<td>LAYNUM</td>
<td>FHIRBSVK</td>
<td>Layer Number</td>
</tr>
<tr>
<td>LIQLIM</td>
<td>B</td>
<td>Liquid Limit</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Source File</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MONTH</td>
<td>FDWK</td>
<td>Month</td>
</tr>
<tr>
<td>MPNIMO</td>
<td>L</td>
<td>Milepoint Month</td>
</tr>
<tr>
<td>MPTNVR</td>
<td>L</td>
<td>Milepoint Year</td>
</tr>
<tr>
<td>NEXTSID</td>
<td>L</td>
<td>Pointer to the Next SID Number for this location</td>
</tr>
<tr>
<td>NO LANES</td>
<td>G</td>
<td>Number of Lanes - One Way</td>
</tr>
<tr>
<td>NUMLANES</td>
<td>L</td>
<td>Number of Lanes</td>
</tr>
<tr>
<td>OUTSHWD</td>
<td>G</td>
<td>Outside Shoulder Width (Feet)</td>
</tr>
<tr>
<td>PAVETYP</td>
<td>G</td>
<td>Type of Pavement</td>
</tr>
<tr>
<td>PCTTRK</td>
<td>T</td>
<td>Traffic Measurement File</td>
</tr>
<tr>
<td>PERMIX</td>
<td>B</td>
<td>Permeability Index</td>
</tr>
<tr>
<td>PLASTIX</td>
<td>B</td>
<td>Plasticity Index</td>
</tr>
<tr>
<td>PPSV200</td>
<td>B</td>
<td>Percent Passing NO. 200 Sieve</td>
</tr>
<tr>
<td>PREVSID</td>
<td>L</td>
<td>Pointer to the Previous SID Number for this location</td>
</tr>
<tr>
<td>RNWSTAVG</td>
<td>F</td>
<td>Average SSI for the Roadway</td>
</tr>
<tr>
<td>SECTION</td>
<td>L</td>
<td>Section</td>
</tr>
<tr>
<td>SENS1RD</td>
<td>D</td>
<td>Sensor 1 Reading</td>
</tr>
<tr>
<td>SENS2RD</td>
<td>D</td>
<td>Sensor 2 Reading</td>
</tr>
<tr>
<td>SENS3RD</td>
<td>D</td>
<td>Sensor 3 Reading</td>
</tr>
<tr>
<td>SENS4RD</td>
<td>D</td>
<td>Sensor 4 Reading</td>
</tr>
<tr>
<td>SENS5RD</td>
<td>D</td>
<td>Sensor 5 Reading</td>
</tr>
<tr>
<td>SHOBSTHK</td>
<td>G</td>
<td>Shoulder Base Thickness (Inches)</td>
</tr>
<tr>
<td>SHOBSTYP</td>
<td>G</td>
<td>Shoulder Base Type</td>
</tr>
<tr>
<td>SHOSPTHK</td>
<td>G</td>
<td>Shoulder Surface Thickness (Inch)</td>
</tr>
<tr>
<td>SHOSPTYP</td>
<td>G</td>
<td>Shoulder Surface Type</td>
</tr>
<tr>
<td>SICOUNT</td>
<td>S</td>
<td>Serviceability Index - Count of Observation</td>
</tr>
<tr>
<td>SID NO</td>
<td>FDHICRBLSTK</td>
<td>Section Identification Number</td>
</tr>
<tr>
<td>SIHIVAL</td>
<td>S</td>
<td>Serviceability Index - High Value</td>
</tr>
<tr>
<td>SIDLOWVAL</td>
<td>S</td>
<td>Serviceability Index - Low Value</td>
</tr>
<tr>
<td>SIMMEAN</td>
<td>S</td>
<td>Serviceability Index - Mean</td>
</tr>
<tr>
<td>SISD</td>
<td>S</td>
<td>Serviceability Index - Standard Deviation</td>
</tr>
<tr>
<td>SKIDNUMH</td>
<td>K</td>
<td>Skid Measurement File</td>
</tr>
<tr>
<td>SKIDNUMIL</td>
<td>K</td>
<td>Skid Measurement File</td>
</tr>
<tr>
<td>SKIDNUMM</td>
<td>K</td>
<td>Skid Measurement File</td>
</tr>
<tr>
<td>SSI1G12</td>
<td>F</td>
<td>SSI Reading 1 Geophone no. 1</td>
</tr>
<tr>
<td>SSI1G12</td>
<td>F</td>
<td>SSI Reading 1 Geophone no. 2</td>
</tr>
<tr>
<td>SSI1G13</td>
<td>F</td>
<td>SSI Reading 1 Geophone no. 3</td>
</tr>
<tr>
<td>SSI1G14</td>
<td>F</td>
<td>SSI Reading 1 Geophone no. 4</td>
</tr>
<tr>
<td>SSI1G15</td>
<td>F</td>
<td>SSI Reading 1 Geophone no. 5</td>
</tr>
<tr>
<td>SSI1G16</td>
<td>F</td>
<td>SSI Reading 1 Geophone no. 6</td>
</tr>
<tr>
<td>SSI1G17</td>
<td>F</td>
<td>SSI Reading 1 Geophone no. 7</td>
</tr>
<tr>
<td>SSI1G18</td>
<td>F</td>
<td>SSI Reading 2 Geophone no. 1</td>
</tr>
<tr>
<td>SSI1G19</td>
<td>F</td>
<td>SSI Reading 2 Geophone no. 2</td>
</tr>
<tr>
<td>SSI1G20</td>
<td>F</td>
<td>SSI Reading 2 Geophone no. 3</td>
</tr>
<tr>
<td>SSI1G21</td>
<td>F</td>
<td>SSI Reading 2 Geophone no. 4</td>
</tr>
<tr>
<td>SSI1G22</td>
<td>F</td>
<td>SSI Reading 2 Geophone no. 5</td>
</tr>
<tr>
<td>SSI1G23</td>
<td>F</td>
<td>SSI Reading 2 Geophone no. 6</td>
</tr>
<tr>
<td>SSI1G24</td>
<td>F</td>
<td>SSI Reading 2 Geophone no. 7</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Source File</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SSIGP31</td>
<td>F</td>
<td>SSI Reading 3 Geophone no. 1</td>
</tr>
<tr>
<td>SSIGP32</td>
<td>F</td>
<td>SSI Reading 3 Geophone no. 2</td>
</tr>
<tr>
<td>SSIGP33</td>
<td>F</td>
<td>SSI Reading 3 Geophone no. 3</td>
</tr>
<tr>
<td>SSIGP34</td>
<td>F</td>
<td>SSI Reading 3 Geophone no. 4</td>
</tr>
<tr>
<td>SSIGP35</td>
<td>F</td>
<td>SSI Reading 3 Geophone no. 5</td>
</tr>
<tr>
<td>SSIGP36</td>
<td>F</td>
<td>SSI Reading 3 Geophone no. 6</td>
</tr>
<tr>
<td>SSIGP37</td>
<td>F</td>
<td>SSI Reading 3 Geophone no. 7</td>
</tr>
<tr>
<td>SSIGP41</td>
<td>F</td>
<td>SSI Reading 4 Geophone no. 1</td>
</tr>
<tr>
<td>SSIGP42</td>
<td>F</td>
<td>SSI Reading 4 Geophone no. 2</td>
</tr>
<tr>
<td>SSIGP43</td>
<td>F</td>
<td>SSI Reading 4 Geophone no. 3</td>
</tr>
<tr>
<td>SSIGP44</td>
<td>F</td>
<td>SSI Reading 4 Geophone no. 4</td>
</tr>
<tr>
<td>SSIGP45</td>
<td>F</td>
<td>SSI Reading 4 Geophone no. 5</td>
</tr>
<tr>
<td>SSIGP46</td>
<td>F</td>
<td>SSI Reading 4 Geophone no. 6</td>
</tr>
<tr>
<td>SSIGP47</td>
<td>F</td>
<td>SSI Reading 4 Geophone no. 7</td>
</tr>
<tr>
<td>SSIGP51</td>
<td>F</td>
<td>SSI Reading 5 Geophone no. 1</td>
</tr>
<tr>
<td>SSIGP52</td>
<td>F</td>
<td>SSI Reading 5 Geophone no. 2</td>
</tr>
<tr>
<td>SSIGP53</td>
<td>F</td>
<td>SSI Reading 5 Geophone no. 3</td>
</tr>
<tr>
<td>SSIGP54</td>
<td>F</td>
<td>SSI Reading 5 Geophone no. 4</td>
</tr>
<tr>
<td>SSIGP55</td>
<td>F</td>
<td>SSI Reading 5 Geophone no. 5</td>
</tr>
<tr>
<td>SSIGP56</td>
<td>F</td>
<td>SSI Reading 5 Geophone no. 6</td>
</tr>
<tr>
<td>SSIGP57</td>
<td>F</td>
<td>SSI Reading 5 Geophone no. 7</td>
</tr>
<tr>
<td>SSITEMP</td>
<td>F</td>
<td>Temperature for the Roadway</td>
</tr>
<tr>
<td>STATION</td>
<td>D</td>
<td>Station</td>
</tr>
<tr>
<td>STRUCTNUM</td>
<td>FDHIGRBSVK</td>
<td>Structure Number</td>
</tr>
<tr>
<td>TXTRIAXL</td>
<td>B</td>
<td>Texas Triaxial Class</td>
</tr>
<tr>
<td>WIDENFLG</td>
<td>G</td>
<td>Type Of Widening Flag</td>
</tr>
<tr>
<td>WIDENIMO</td>
<td>I</td>
<td>Month Layer Widened</td>
</tr>
<tr>
<td>WIDENLYR</td>
<td>I</td>
<td>Year Layer Widened</td>
</tr>
<tr>
<td>YEAR</td>
<td>FDUK</td>
<td>Year</td>
</tr>
<tr>
<td>YEAR</td>
<td>SV</td>
<td>Identifying Year</td>
</tr>
</tbody>
</table>

** Source File Key

- **F** - Falling Weight SSI
- **D** - Dynaflect Measurement
- **H** - Layer Thickness Across the Road
- **I** - Layer Identification
- **G** - Geometric & Shoulder
- **R** - Surface
- **B** - Subgrade
- **L** - Location
- **W** - Weather
- **E** - Environment
- **S** - Serviceability Index
- **V** - Visual
- **T** - Traffic
- **K** - Skid
APPENDIX C
SYSTEM CONFIGURATION

The following are the requirements of the Texas Flexible Pavement System:

Hardware requirements:

- IBM Personal Computer XT or Compatible
- 640K bytes of RAM
- 360K Floppy Disk Drive
- 20 Megabyte Hard Disk
- Printer (able to condensed print - 133 columns)
- Monochrome or Color Monitor

Software requirements:

- DOS 3.3 or later
- dBASE III Plus (By Aston Tate)
- TURBO Pascal 3.0 (For source code modification)

Space requirements:

- 20 Megabytes of storage space.
APPENDIX D
INSTALLATION

The installation is accomplished by the program INSTALL.BAT. It creates the appropriate subdirectories for the Flexible Pavement System and then calls program SETUP.BAT. Program SETUP.BAT unarchives the files from 8 floppy diskettes onto the hard disk. The unarchive program PKXARC.COM by PKWARE INC. is used to unarchive programs. The program is activated from the hard disk prompt by typing "A:INSTALL".

The 8 floppy diskettes contain the following files:

DISK 1 - BACKUP.ARC
          INQUIRY.ARC
          REPORTS.ARC
          PAVEDB.ARC
          PKXARC.COM
          FLEXPAVE.BAT
          INSTALL.BAT
          SETUP.BAT

DISK 2 - EDITUPDT.ARC
          E_DIST.ARC

DISK 3 - E_TRAFFI.ARC

DISK 4 - E_PES.ARC

DISK 5 - APPLICAT.ARC
          A_DIST.ARC
          A_MODEL.ARC
          A_GRAPH.ARC

DISK 6 - FILES1.ARC

DISK 7 - FILES2.ARC

DISK 8 - INDEXES.ARC
ECHO OFF
REM SUBSYSTEM: PAVEDB
REM PROGRAM NAME: INSTALL.BAT 10/27/88
REM PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
REM TAMU/TTI
REM AUTHOR: TREVOR X. PEREIRA
REM PURPOSE: TO INSTALL THE FLEIBLE PAVEMENT DATABASE SYSTEM
REM ON THE HARD DISK
CLS
ECHO'
ECHO'
ECHO'
REM
REM asks user if he wants to install flexible pavement system
REM
ECHO You are about to install the Flexible Pavement system on
ECHO your hard disk under \PAVEDB.
ECHO If you already have the system installed on your computer,
ECHO stop this operation by pressing ctrl+break simultaneously.
ECHO'
ECHO'
ECHO If you do not want to continue, press ctrl+break simultaneously.
ECHO If you wish to continue, press any key.
PAUSE
CLS
ECHO'
ECHO'
ECHO'
ECHO'
ECHO'
ECHO'
ECHO'
ECHO'
ECHO'
ECHO'
ECHO'
ECHO' Please wait. Creating subdirectories.
CD\
MD \PAVEDB
MD \PAVEDB\APPLICAT
MD \PAVEDB\APPLICAT\DISTRESS
MD \PAVEDB\APPLICAT\GRAPH18K
MD \PAVEDB\APPLICAT\MODEL
MD \PAVEDB\BACKUP
MD \PAVEDB\EDITUPDT
MD \PAVEDB\EDITUPDT\BACKUP
MD \PAVEDB\EDITUPDT\DISTRESS
MD \PAVEDB\EDITUPDT\PES
MD \PAVEDB\EDITUPDT\TRAFFIC
MD \PAVEDB\FILES
MD \PAVEDB\FILES\BACKUP
MD \PAVEDB\INDEXES
MD \PAVEDB\INQUIRY
MD \PAVEDB\REPORTS
CD \\

CLS
copy a:setup.bat
setup.bat
PROGRAM LISTING

REM
REM SUBSYSTEM: PAVEDB
REM PROGRAM NAME: SETUP.BAT 010/04/88
REM CALLED FROM: INSTALL.BAT
REM PROJECT 2456 - TEXAS FLEXIBLE PAVEMENT DATABASE CONVERSION
REM TAMU/TTI
REM AUTHOR: TREVOR X. PEREIRA
REM PURPOSE: TO SET UP THE FLEXIBLE PAVEMENT SYSTEM ON THE USERS HARD DISK.
REM

echo off
ECHO '
ECHO '
ECHO '
ECHO '
ECHO '
ECHO '
ECHO '
ECHO '
ECHO '
ECHO ' Please wait

CD \
COPY A:PKXARC.COM
CD \PAVEDB
PKXARC A:PAVEDB
CD \PAVEDB\INQUIRY
PKXARC A:INQUIRY
CD \PAVEDB\REPORTS
PKXARC A:REPORTS
CD \PAVEDB\BACKUP
PKXARC A:BACKUP

CLS
ECHO '
ECHO '
ECHO '
ECHO '
ECHO '
ECHO '
ECHO '
ECHO '
ECHO '
ECHO '
ECHO '
ECHO '
ECHO ' Please insert FLEXPAVE DISK 2
PAUSE
CD \PAVEDB\EDITUPDT
PKXARC A:EDITUPDT
CD \PAVEDB\EDITUPDT\DISTRESS
PKXARC A:E_DIST

662
CLS
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO Please insert FLEXPAVE DISK 3 
PAUSE 
CD \PAVEDB\EDITUPDT\TRAFFIC 
PKXRARC A:E_TRAFFIC 

CLS 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO Please insert FLEXPAVE DISK 4 
PAUSE 
CD \PAVEDB\EDITUPDT\PES 
PKXRARC A:E_PES 

CLS 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO ' 
ECHO Please insert FLEXPAVE DISK 5 
PAUSE 
CD \PAVEDB\APPLICAT 
PKXRARC A:APPLICAT 
CD \PAVEDB\APPLICAT\DISTRESS 
PKXRARC A:A_DIST 
CD \PAVEDB\APPLICAT\GRAPH18K 
PKXRARC A:A_GRAPH 
CD \PAVEDB\APPLICAT\MODEL 
PKXRARC A:A_MODEL 

CLS 
ECHO '
ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO Please insert FLEXPAVE DISK 6 PAUSE CD \PAVEDB\FILES PKXARC A:FILES1

CLS ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO Please insert FLEXPAVE DISK 7 PAUSE PKXARC A:FILES2

CLS ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO Please insert FLEXPAVE DISK 8 PAUSE CD \PAVEDB\INDEXES PKXARC A:INDEXES

CD \ DEL PKXARC.COM DEL \PAVEDB\INSTALL.MEM

CLS ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO ' ECHO '
ECHO ' 
ECHO ' 
ECHO INSTALLATION COMPLETE !!! 
ECHO TYPE "FLEXPAVE" TO START SYSTEM AND SET DEFAULTS 
cd \pavedb 
ECHO ON