Abstract

Base SAS® software for Personal Computers can be used as a very efficient and flexible data editor. Within a SAS program, the DATA step is used to create windows through which tasks such as data entry, data validation, and query are performed.

Introduction

There are many ways within the SAS System to create data entry screens; however, using the WINDOW and DISPLAY statements within base SAS software, data validation and cross-checking are also easily accomplished. Some of the advantages of this technique include the ability to customize the editor to meet specific needs, the availability of efficient table look-up facilities at any time during an edit session, cross-checking of input values to any desired level of detail, and the ability to perform calculations at any time during a SAS session. This technique also offers efficient query for large data files via the binary search.

Creating and Displaying Windows

You can use the WINDOW and DISPLAY statements within base SAS software for Personal Computers to create example windows for entering point-of-sale information. The WINDOW statement defines all windows at compile time and also validates literals, variables, and formats:

```sas
WINDOW SHOW COLOR=GREEN IROW=1 ICOLUMN=1 ROWS=10 COLUMNS=40 GROUP=ONE
#1 'Input Item Here: 'ITEM $20.
#2 'Input Quantity Here: 'QUANTITY 8.
#3 'Description: 'DESCR $20.
#4 'Cost: 'SALES DOLLAR10.2
#5 'Total Amount: 'TOTAL DOLLAR10.2;
```

The DISPLAY statement allows you to call a defined window any number of times during a single execution of the DATA step. For example:

```sas
WINDOW SHOW COLOR=GREEN IROW=1 ICOLUMN=1 ROWS=10 COLUMNS=40 GROUP=ONE
#1 'Input Item Here: 'ITEM $20.
#2 'Input Quantity Here: 'QUANTITY 8.
#3 'Description: 'DESCR $20.
#4 'Cost: 'SALES DOLLAR10.2
#5 'Total Amount: 'TOTAL DOLLAR10.2;
```

Data Entry and Validation

Now that our point-of-sale data entry screen has been developed and can be displayed, practical data entry is only a loop away. The special variable _CMD_ is used as a controller:

```sas
DATA SALES;
WINDOW SHOW COLOR=GREEN IROW=1 ICOLUMN=1 ROWS=10 COLUMNS=40 GROUP=ONE
#1 'Input Item Here: 'ITEM $20.
#2 'Input Quantity Here: 'QUANTITY 8.
#3 'Description: 'DESCR $20.
#4 'Cost: 'SALES DOLLAR10.2
#5 'Total Amount: 'TOTAL DOLLAR10.2;
DO UNTIL (UPCASE(_CMD_)='HALT');
DISPAY SHOW.ONE;
TOTAL=QUANTITY * SALES;
DISPAY SHOW.ONE;
OUTPUT;
END;
STOP;
```

Formats used in conjunction with the PUT function provide efficient table look-up facilities for cross-checking and validation of input values. You must first create formats that correspond to the table values:

```sas
PROC FORMAT;
VALUE $DESFMT '100'='1/2 inch lag bolt'
     '200'='3/4 inch lag bolt';
VALUE $COSTFMT '100'='19.95'
     '200'='9.87';
```

Then, you create a DATA step and use these formats to complete the appropriate values as in this example:

```sas
DATA SALES;
WINDOW SHOW COLOR=GREEN IROW=1 ICOLUMN=1 ROWS=10 COLUMNS=40 GROUP=ONE
#1 'Input Item Here: 'ITEM $20.
#2 'Input Quantity Here: 'QUANTITY 8.
#3 'Description: 'DESCR $20.
#4 'Cost: 'SALES DOLLAR10.2
#5 'Total Amount: 'TOTAL DOLLAR10.2;
DO UNTIL (UPCASE(_CMD_)='HALT');
DISPLAY SHOW.ONE;
*DO TABLE LOOK-UP AND CHARACTER CONVERSION;
DESCR=PUT (ITEM,$DESFMT.);
COST=INPUT (DESCR,COSTFMT.);
END;
STOP;
```
Data Query

The key to binary searching of a SAS data set is the POINT= and NOBS= options in the SET statement. Using these two options, calculations and comparisons are performed to find the desired record:

```plaintext
SEARCH=108;
\*SET UP SEARCH RANGE;
LO=1;
HI=NOBS;
MID=FLOOR((HI-LO)/2);
\*LOOP UNTIL DESIRED PART NUMBER IS FOUND;
DO WHILE (LO = MID);
  SET SAVE.INVENTORY POINT=MID NOBS=NOBS;
  IF SEARCH>PARTNO THEN LO=MID;
  ELSE IF SEARCH<PARTNO THEN HI=MID;
  MID=LO+FLOOR((HI-LO)/2);
END;
```

Let's look at how a binary search actually works. The initial pass on the file is represented below:

```
<table>
<thead>
<tr>
<th>Record</th>
<th>PARTNO</th>
<th>SEARCH</th>
<th>MID</th>
<th>LO</th>
<th>HI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>101</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>102</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>103</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>104</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>105</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>106</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>107</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>108</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>109</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>110</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

The results of the second pass follow:

```
<table>
<thead>
<tr>
<th>Record</th>
<th>PARTNO</th>
<th>SEARCH</th>
<th>MID</th>
<th>LO</th>
<th>HI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>101</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>102</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>103</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>104</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>105</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>106</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>107</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>108</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>109</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>110</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

The results of the third pass follow:

```
<table>
<thead>
<tr>
<th>Record</th>
<th>PARTNO</th>
<th>SEARCH</th>
<th>MID</th>
<th>LO</th>
<th>HI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>101</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>102</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>103</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>104</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>105</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>106</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>107</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>108</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>109</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>110</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

The above binary search, when used as a query method for a 40-byte, 1000-record file, can range from 20 seconds for an XT reading from a floppy drive to 4 seconds for an AT reading from a fixed disk.

Conclusion

The windowing facility in base SAS software for Personal Computers allows flexible creation of windows for customized data entry, query, and data validation with table look-up. You can also use this same facility to drive other applications -- simply use your imagination to create front-end report writing and presentation graphics.

SAS is a registered trademark of SAS Institute Inc., Cary, NC, USA.