COMPUTERIZED MATERIAL LOADING SCHEDULES: AN AID IN IMPLEMENTING HIGH DENSITY STORAGE SYSTEMS

Darryl W. Kolojcar, Houston Lighting & Power Co.

ABSTRACT

This paper reports the development and use of software to provide a schedule for implementing new storage equipment at Houston Lighting & Power Co.'s service center warehouses. The output from this development has been used by warehouse personnel to determine where specific material is scheduled to be stored in modular drawer additions. Each page of the output report provides a list of material representing a sub-set of the full list of items being moved. These items are cross-referenced to their proposed warehouse grid location. The output page also provides the user with a profile view of the drawer unit in which the material is to be stored, identifying the cabinet identification number in the installation. This profile includes a set of arrows pointing to the cabinet drawer in which the material is to be stored. Finally, a layout of the drawer is provided indicating the exact compartment in which each part number is to be stored.

INTRODUCTION

Effective communication of technical information to non-technical personnel is essential to the success of implementing modern storage equipment in warehousing. Using base SAS® software, a mixture of graphics and tabular output may be generated to provide warehouse personnel with a schedule for moving material from existing equipment into new installations of high density storage equipment.

High density storage systems have become increasingly popular as the costs of land and building construction continue to rise. The trend in material handling has been to replace old inefficient storage systems with new, space efficient equipment. This equipment replacement cycle involves five major phases:

I. Identify candidate material for storage in the new system.
II. Conduct an engineering feasibility study to determine the most effective type of system, considering both the storage requirements of the candidate items and the desired space savings goals of the replacement project.
III. Procure the new system, allowing potential vendors to conduct surveys of the candidate items. Based on these professional surveys, the system equipment requirements are determined, the system layout is designed, and the material loading schedule is formulated.
IV. Install the equipment.
V. Transfer the material from the old system into the new system based on the loading schedule formulated during procurement.

The research conducted during the feasibility study provides the decision of what type of system is required. However, the planning activities that occur during procurement can be the deciding factor of whether or not a high density storage system succeeds. The software package presented was designed for a specific storage system type (modular drawers), but the programming technique is adaptable to other types of equipment.

DATA REQUIREMENTS

The loading schedule software is dependent upon collection of data from three sources:
1) the identification of candidate material,
2) the engineering feasibility study, and
3) the vendor surveys.

In the scheduling system test case, warehouse personnel at the project site were responsible for identifying the material to be included in the system. Decisions were based on item size, frequency of issue, and security needs. A preliminary list of material formed the basic database for development of the implementation program package.

The engineering feasibility study consisted of a random sample of the material to determine factors such as the average, minimum, and maximum size of the parts; the most likely maximum stock quantity for each part number; and the average storage volume required for the maximum stock quantity of each part number. (See Table 1.) Most likely maximum stock quantities are theoretical values based on a warehouse's inventory control system order rules. Modular drawer cabinets were chosen as the replacement system because of the small to medium average item size, and the potential for reducing storage volume requirements by two-thirds.

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>MEAN VALUE</th>
<th>MINIMUM VALUE</th>
<th>MAXIMUM VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>1.304</td>
<td>0.063</td>
<td>6.500</td>
</tr>
<tr>
<td>W</td>
<td>3.157</td>
<td>0.250</td>
<td>14.000</td>
</tr>
<tr>
<td>D</td>
<td>4.654</td>
<td>0.250</td>
<td>28.000</td>
</tr>
<tr>
<td>VOLUME</td>
<td>1620.429</td>
<td>0.422</td>
<td>63000.000</td>
</tr>
</tbody>
</table>

Table 1

PROC MEANS of Size and Volume Requirements of Candidate Material for New Storage System

During procurement, three potential vendors surveyed the project site to determine the quantity of drawer units, the drawer sizes, and partition arrangements required. The surveys consisted of 100% sampling of the candidate items to insure a high percentage of success in placing the items selected in Phase I. Vendors were supplied with printed output from the database including the part number, quantity on hand on the vendor survey, and expected maximum stock quantity. (See Appendix A.)
The equipment recommendations made by the vendors were for a variety of cabinet sizes; consequently, evaluation of the proposals could not be based on a price per equipment unit. Evaluations were based on the percent utilization measured by dividing the volume required for maximum stock quantities by the volume the vendor allowed:

\[
\text{Utilization} = \frac{\text{Volume Required}}{\text{Volume Allowed}} \times 100
\]

Another evaluation measure was the failure rate - the number of part numbers from the database either not placed or placed in insufficient space (Utilization < 100). Time constraints necessitated manual calculation of this measure using a bid evaluation worksheet generated from the database. (See Appendix B.) The contract was awarded to the vendor providing the highest percent utilization and lowest failure rate.

OUTPUT DEVELOPMENT

The loading schedule received from the contract winning vendor was technically a very good schedule. However, to provide for a smooth transition of material from the old system to the new, additional information was needed on the schedule. The vendor schedule identified exactly which cabinet and drawer in the shipment each candidate item was intended to fit into, but it deviated slightly from the material location system the warehouse personnel were accustomed to using. Consequently, a grid location system was developed for the warehouse.

To provide training in the use of the locator system, the warehouse grid location needed to be included on the material loading schedule. As a result of this need, a new schedule was designed.

The schedule developed provides a bridge between the manufacturer's packing list, the vendor's loading schedule, the equipment layout, and the warehouse grid location. Each of these sources contribute information about the equipment to the final output report. Each page of output is broken into three parts (See Appendix C).

Part A provides the user with a profile view of the basic equipment unit - in this application the modular drawer cabinet. The profile includes the vendor shipping identification number and two arrows pointing to the equipment sub-unit (drawer) for which Parts B and C provide detail. The letter printed on the drawer corresponds to the seventh digit of the warehouse grid location. This letter uniquely identifies the level above the warehouse floor at which the material is to be stored.

Part B displays a layout of the interior of the drawer. Each part number being stored in the drawer is placed in the corresponding compartment for which it is scheduled. The numbers printed in the lower left corner of each compartment on the layout correspond to the eighth digit of the grid location. This digit identifies the horizontal space number from the left of the sub-unit at which the material is to be stored, e.g. 2 identifies the second compartment from the left. The letters printed at the far right of the layout identify the location of the compartment where the material is to be stored. For instance, the letter "B" identifies the second compartment from the front of the drawer. This letter makes up the tenth digit of the grid location. Beneath the layout, a summary of the drawer's attributes is printed, including the level letter, manufacturer's loading schedule, the equipment layout, and compartment volume.

Part C provides the user with a table of all part numbers scheduled to be stored in the drawer, cross-referencing them to the grid location of the compartment in which they are placed in the layout given in Part B. The location identifies the zone of the warehouse where the material is located. The second and third digits identify the aisle of the equipment layout in which the material is located. The aisle is further broken down into sections represented by digits four through six. This section number, when associated with the zone and aisle, identifies the exact basic equipment unit in which the material is located. The remaining digits of the grid location have already been discussed in detail. The cross-reference table is included as an operational aid in pulling material from the old system and as a teaching aid in developing an association of the grid location with its corresponding physical location in the new equipment.

PROGRAMMING TECHNIQUE

The scheduling program uses eight primary attributes. The first two attributes make up the item's part number. The part number is broken into two parts because of the configuration of H.L.& P. A part number for H.L.& P. is obtained by assigning a two-part number called a class-bin. The first part of the number is called the class and consists of no more than three digits. The class identifies a particular type of material, i.e., tools, pipe fittings, insulators, etc. The second part, called the bin, is a sequential number identifying an exact part within the class, i.e., a 3/4" coupling. The bin number may be up to five digits in length.

The next three attributes correspond to the height, width, and depth of the drawer in which the class-bin is to be stored. The drawer and compartment volumes appearing in the summary table of Part B are calculated using these three codes.

The next two attributes identify the number of compartments in the drawer. One of these codes, labeled CIOW, identifies the number of compartments spanning the width of the drawer. The other code, CMOO, identifies the number of compartments...
The last attribute is the material location, a sixteen character alpha-numeric value identifying the cabinet and aisle in which the material is scheduled to be located. This location is designed around five secondary attributes as shown below:

<table>
<thead>
<tr>
<th>Shipment Cabinet No.</th>
<th>Drawer Model No.</th>
<th>Drawer Level Letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>051756</td>
<td>150DR01026</td>
<td></td>
</tr>
<tr>
<td>Partition Model No.</td>
<td>Cabinet Model No.</td>
<td></td>
</tr>
</tbody>
</table>

These secondary codes contribute to the cabinet identification in Part A and the layout summary in Part B of the output.

The last manipulation performed in the input data step is the determination of the line number at which the drawer corresponding to the location will appear on the output report. This information is necessary so that when the profile of the cabinet is printed on the output, a set of arrows may be printed pointing to the drawer for which the output page gives detailed information. The drawer level letter is used to determine the line number as follows:

\[
\text{IF LEVEL} = \text{A' THEN LVLOI} = 34; \\
\text{IF LEVEL} = \text{B' THEN LVLOI} = 32; \\
\text{IF LEVEL} = \text{C' THEN LVLOI} = 30; \\
\text{ETC.}
\]

One further manipulation is required before the schedule may be printed. The number of drawers in each individual cabinet must be determined so that the print routine will print the correct profile in the profile section of the schedule. To determine this number the data set must be sorted by cabinet number and descending level letter.

The next data step takes the sorted input data and determines the number of drawers in each cabinet based on the level letter of the first observation of each cabinet number in the data set. This determination is accomplished using a PUT function and a PROC FORMAT translating the sequence of the alphabet into a numeric sequence:

\[
\text{PROC FORMAT;} \\
\]

\[
\text{DATA DEPPM; SET CBLOCATE; BY CABNO; IF FIRST.CABNO THEN 1 = PUT(LEVEL,$REDO.1); RETAIN 1;}
\]

The print routine uses a NULL data set containing the data set DEPPM to print the leading schedule. The data set DEPPM is set by location (LOC) using a NOTSORTED option. Because the data set DEPPM is sorted indirectly from the input data set (CBLOCATE) by cabinet number and descending level, all class-bins are grouped by drawer. This indirect sorting allows the output to be grouped one complete drawer per page.

Two increment variables are used throughout the routine to index the pointer control to each of the drawer compartments. The first of these increment variables is a depth increment value and is based on the number of compartments spanning the depth of the drawer (CNOD). The report uses
To output lines for the depth of the drawer.

Similarly, the compartment width increment value is calculated based on 60 output columns for the width of the drawer. Using these specifications, the increment values are calculated as follows:

\[
\begin{align*}
\text{DINC} &= \text{INT}(30/\text{CNOD}); \\
\text{WINC} &= \text{INT}(60/\text{CNOW});
\end{align*}
\]

After calculating the increment values, the routine begins the FILE PRINT. Options used on the FILE PRINT include N=PS to allow the routine to use the full page when writing the three parts of the schedule. A HEADER option is also used to print the summary table labels and the outline of the drawer.

On the first location, the routine performs four tasks: 1) write the profile view of the cabinet, 2) write the partition layout in the drawer, 3) write the class-bin vs grid location table outline, and 4) initialize line and column counters for printing class-bins in the drawer layout.

The profile of the cabinet is written by repeating a two line PUT statement that draws one drawer. The number of times this PUT statement is repeated is determined by the number of drawers in the cabinet (I from data step DEFPM). A DO loop is used to repeat the PUT:

\[
\text{DO T=1 TO I;}
\]

\[
\begin{align*}
\text{XI} &= 36 - T*2; \\
\text{PUT #X1 05 '1' 25*=' '1' @6 'T' LVLFMT. ' 039 ' / 05 '1' 33*';} \\
& \text{IF T=1 THEN DO;} \\
& \text{R} = \text{XI} - 1; \\
& \text{PUT #R 06 33*';} \\
& \text{END;} \\
& \text{END;} \\
\text{PUT #42 00 '1' 59*'}
\end{align*}
\]

The format LVLFMT. performs the inverse of format #REDU. used to calculate 1. For example, if T = 1 then LVLFMT. writes an 'A', if T = 2 then LVLFMT. writes a 'B', etc. The nested DO finishes the top of the cabinet when the top drawer has been written. The arrows pointing to the drawer in the profile for which the output detail is related are printed on line number LVL01 calculated in the input data step. The cabinet number and drawer summary details, such as drawer model number, partition number, and compartment dimensions are written in the drawer summary.

At this point, the routine begins a series of three loops to write the drawer layout, giving the drawer its depth compartments.

IF CNOO > 1 THEN CD = CNOD - 1; ELSE CD = 1; CACC = 5;

\[
\text{DO T=1 to CD;}
\]

\[
\begin{align*}
\text{CACC} &= \text{CACC} + \text{DINC}; \\
\text{PUT #CACC 061 59*'} \\
\text{DALPHA} &= \text{CNOD} - T + 1; \\
\text{PUT #CACC 021 DALPHA LVLFNT.}
\end{align*}
\]

The line counter, CACC, is indexed by the depth increment, DINC, on each pass of the loop, thus evenly spacing the horizontal partitions. The drawer depth letter, DALPHA, is written to the far right of the drawer layout using the LVLFNT. format. This letter corresponds to the tenth digit of the warehouse grid location.

The next loop of the series writes the vertical partitions in a manner similar to the loop that wrote the horizontal partitions.

IF CNOW > 1 THEN CW = CNOW - 1; ELSE CW = 1; DO T=6 TO 35:

\[
\begin{align*}
\text{CACC} &= 0; \\
\text{U} &= 1; \\
\text{DO T=6 TO 35:} \\
& \text{CACC} = \text{CACC} + \text{WINC}; \\
& \text{PUT #CACC '1';}
\end{align*}
\]

The final loop of the series writes the horizontal space number in the left front corner of each compartment. The space number corresponds to the eighth digit of the warehouse grid location.

\[
\text{DO T=1 TO CNOD;}
\]

\[
\begin{align*}
\text{U} &= 35 - (T-1)*\text{DINC;} \\
\text{DO V=1 TO CNOW;}
\end{align*}
\]

\[
\begin{align*}
\text{W} &= 61 + (V-1)*\text{WINC;} \\
\text{PUT IU 00 '1'} \\
& \text{END;} \\
& \text{END};
\end{align*}
\]

At this point the routine prints the outline and labels for the class-bin vs grid location table as follows:

\[
\begin{align*}
\text{PUT #42 061 '1' 59*'} \\
& \text{/ 66 '1' 59*' '0120 '1' / 66 '1' 59*' '0120 '1' / 66 '1' 59*';} \\
& \text{/ 60 '1' 59*';} \\
\text{DO T=1 TO 10;}
\]

\[
\begin{align*}
\text{XI} &= 46 + T; \\
\text{PUT #X1 000 '1' 0120 '1'} \\
& \text{END;} \\
\text{PUT #57 000 '1' 59*';} \\
& \text{0120 '1'};
\end{align*}
\]

After the report layout has been written, the routine initializes four variables:

\[
\begin{align*}
\text{T} &= 0; \\
\text{L} &= 0; \\
\text{LCOLUMN} &= 65; \\
\text{LLINE} &= 40;
\end{align*}
\]

These variables are used as line and column counters in writing the class-bins to their respective locations on the layout. T counts the
horizontal compartment to which the pointer is being directed. L counts the vertical compartment to which the pointer is being directed. When T or L exceeds its limit, CNOW or CNOD respectively, it is reinitialized to zero, thus keeping writing within the boundaries of the drawer layout. LCOLUMN tracks the column at which class-bins and grid locations are being written in the cross-reference table. LLINE tracks the line number to which the pointer is being directed when writing values in the table. When LLINE exceeds 55, it is reinitialized to 46 and LCOLUMN is reinitialized to 90. This formats the table in two columns of output with up to twenty class-bins.

After the line and column counters are initialized, the FIRST.LOC loop is ended. Once this loop has been closed, the routine begins the algorithm used to 1) place each class-bin in a compartment of the layout and 2) write the class-bin and its corresponding grid location to the cross-reference table.

Class-bins are placed in compartments in the following order. First, compartments are filled horizontally. For example, in a drawer containing three compartments, all in the horizontal direction, but only two class-bins, the two compartments to the far left of the drawer would contain the class-bin numbers. Once the horizontal direction is filled, the routine advances to the next compartment and the horizontal counter, T, is reinitialized to the left compartment (T=0). The following statements perform this section of the routine:

```
COLUMN = 60 + TWINC + INT(WINC/2) - 3;
LINE = 35 + LDINC - INT(DINC/2);
PUT LLINE BCOLUMN CLASS 3. '-' BIN:
IF T = (CNOW-1) THEN 30;
T = T + 1;
L = L + 1;
END;
T = T + 1;
RETAIN T;
```

The algorithm to write the class-bin and grid location to the cross-reference table uses a series of column counters to direct the pointer in writing the grid location attributes and characters to the table. These counters are based on the column counter LCOLUMN:

<table>
<thead>
<tr>
<th>Counter</th>
<th>Attribute/Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCALL</td>
<td>AISLE NUMBER</td>
</tr>
<tr>
<td>LCLS</td>
<td>SECTION NUMBER</td>
</tr>
<tr>
<td>LLCL</td>
<td>LEVEL LETTER</td>
</tr>
<tr>
<td>LCLK</td>
<td>SPACE NUMBER</td>
</tr>
<tr>
<td>LCO1</td>
<td>DEPTH LETTER</td>
</tr>
<tr>
<td>LD1</td>
<td>FIRST DASH</td>
</tr>
<tr>
<td>LD2</td>
<td>SECOND DASH</td>
</tr>
<tr>
<td>LD3</td>
<td>THIRD DASH</td>
</tr>
<tr>
<td>LD4</td>
<td>FOURTH DASH</td>
</tr>
</tbody>
</table>

Based on these column counters, the following routine writes the values to the cross-reference table.

```
LINE = LINE + 1;
LI = L + 1;
T1 = T + 1;
PUT LLINE BCOLUMN CLASS 3. '-' BIN BLREI AIaLEI BLOI '-' BLCLK ALISIEI BLD2 '-' BLLCL ALCEI:
IF LAST.LOC THEN PUT _PAGE_;
END;
RETAIN LLINE LCOLUMN;
```

When the observation being written is the last location and, consequently, the last class-bin in the drawer, the page is printed:

```
IF LAST.LOC THEN PUT _PAGE_;
```

The routine continues processing the file, taking next location and writing the next page, or drawer, of the schedule.

**RECOMMENDATIONS FOR FURTHER DEVELOPMENT**

Bid evaluation of the three vendor surveys and proposals for percent utilization and failure rate required 60 man-hours under the manual system. Software to perform these evaluations would be a valuable addition to the equipment replacement cycle.

An added dimension to the programming technique presented is the ability to choose the exact compartment in the equipment sub-unit in which the class-bin is to be placed. The routine used in this application places class-bins as they are encountered in the data set. This method does not allow for empty compartments between full ones. The class-bin placement technique is being developed further at one of H.L.& P.'s larger warehouses. The new development, when completed, will show profiles of all of the warehouse's storage equipment, including the class-bins placed in their proper compartments as dictated by the warehouse grid location. This software system will be used to identify empty or "free" locations in the warehouse.

**ACKNOWLEDGEMENTS**

- Dinicelli, Paula. Houston Lighting & Power Co.
- Porter, Christina. Houston Lighting & Power Co.
- Vaghela, Mansukh. Houston Lighting & Power Co.

For further information contact:
Darryl W. Kolojaco
Houston Lighting & Power Co.
12301 Kurland Dr.
Houston, Texas 77034
(713) 481-7403

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