5-2007

Distribution supply chain optimization.

Elizabeth A. Forney
University of Louisville

DISTRIBUTION SUPPLY CHAIN OPTIMIZATION

By

Elizabeth A. Forney
B.S.I.E. University of Louisville, 2006

A Thesis
Submitted to the Faculty of the
University of Louisville
J.B. Speed School of Engineering
in Partial Fulfillment of the Requirements
for the Professional Degree

MASTER OF ENGINEERING

Department of Industrial Engineering

May 2007

DISTRIBUTION SUPPLY CHAIN OPTIMIZATION

Submitted by: _____
Elizabeth A. Forney

A Thesis Approved On

_____
(Date)

by the Following Reading and Examination Committee:

_____
Dr. Gerald W. Evans, Thesis Director

_____
Dr. Gail W. DePuy

_____
Dr. Dar-Jen Chang

ACKNOWLEDGEMENTS

I would first like to thank the Industrial Engineering professors of the University of Louisville for their dedication to the students. I would also like to thank Dr. Gerald Evans, who created the simulation model discussed in this thesis. As my advisor for this thesis, he provided the knowledge and support for the entire project, and this would not have been possible without his assistance. I also thank Dr. Gail DePuy for her previous work with the company and her technical assistance. And I would like to thank Dr. John Usher, Chairman of the Industrial Engineering Department, for his commitment to fund a portion of the project, and for his passion for IE that has inspired so many of his students. And finally, I thank Dr. Dar-Jen Chang for agreeing to be on my committee and offer his comments.

Of course, I am also forever grateful to my parents and friends who encouraged me for the last two semesters.

ABSTRACT

The purpose of this project was to analyze the ordering policies for a distribution system. The company has one central distribution center and nineteen branch warehouses spread throughout the country. Each branch warehouse can order product lines either through the local Distribution Center or directly from the vendor. The problem is that the current order policies are ineffective, and it was hypothesized that changes to these policies could result in a significant improvement over the current operations. Therefore, the distribution system was analyzed using an Arena computer simulation model that was developed previously. This project focused on comparing the current ordering policies for the product lines to the optimal policies as determined by the simulation model.

Before the simulation runs could be completed, eleven input files needed to be created for each vendor. This process was completed using Visual Basic macros to ease future input file generation. These macros search and compute data from the spreadsheets provided by the company.

The results of the analysis show that in general, the company should order more products through the vendor than they currently plan. Changing the ordering policies for all 25 of the studied vendors could potentially yield a profit increase of $629,500 over a nine month steady-state period.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

Manufacturing traditionally refers to the processes involved in transforming raw materials into finished goods. Industrialized countries rely on various types of manufacturing to produce everything from food and packaging materials to furniture and electronics. Annually, manufacturing makes up about 12% of the U.S. Gross Domestic Product, or $12.5 billion (National Association of Manufacturers, 2005).

Conversely, logistics refers to organizing the transportation and storage of the raw materials and manufactured goods among various suppliers, manufacturers, distributors, retailers, and customers (Shapiro, 2001). Logistics companies do not make money by the typical methods of manufacturing products; they are simply transporting and storing the products in between the manufacturer and the end customer. Although these logistics companies are not directly adding value to the product itself, the transportation and storage of materials is a critical step in the supply chain. According to Chopra (2007), "distribution-related costs make up about 10.5 percent of the U.S. economy and about 20 percent of the cost of manufacturing." As a result, logistics management is a broad and growing field that affects every business and consumer in some way. Due to the far-reaching consequences of logistics, reducing logistics costs is one of the top priorities for many companies striving to increase revenue.

For the purposes of this paper, a supply chain or logistics system will refer to the high-level stages from raw materials to the end customer, including all levels of

manufacturing, storage, and transportation (Jain, et al, 2001). One aspect of a logistics system is the distribution system, which refers only to the transportation and warehousing of products between the manufacturer and the retailer or customer. The company analyzed in the following report operates as a logistics provider; managing the distribution system from vendors to customers.

Companies typically make logistics decisions on three levels: strategic, tactical and operational. The strategic decisions are long-term decisions such as the number of facilities and their locations. Tactical decisions aren't as permanent, but still have significant effects on the system. These would include decisions made on a monthly or yearly basis such as product mix, transportation methods, or sources. Operational level decisions refer to daily decisions, including daily shipping schedules or ordering decisions (Simchi-Levi, 2005). The analyses detailed later in this report focus on the tactical level logistics decisions made by the company, namely the optimal supply sources for their products.

The main goal of any logistics system is to increase revenue by reducing costs and increasing profit. However, there are several secondary goals that must be considered in terms of appropriate trade-offs between priorities. For example, maintaining minimal inventory will save money in holding costs, but will hurt a company in the long run in terms of the high incidence of lost sales. These priorities vary between companies, and even between the decision makers in the same company. Therefore, tradeoffs must be evaluated between revenue, service level, inventory investment, quality, flexibility, responsiveness, and many other possible objectives. While the analysis detailed in this

paper focuses only on net profit, future research could be done with constraints on other performance measures, depending on the priorities of the company.

A. Simulation Background

To analyze changes to their logistics systems, companies can choose between two diverse strategies. The first option is to make changes to the system and wait, usually for a significant period of time, to evaluate the results of the changes. The other alternative is to create a computer simulation model of the system to quickly analyze the impacts of possible modifications before physically implementing the changes. In general, computer simulation is a superior option in cases where an accurate simulation model can be developed, since the cost of simulation development is relatively low in comparison to the high costs of making poor decisions. Simulation also gives an analyst the tools needed to explore "what-if" scenarios and eventually approach the optimal solution, allowing full consideration of all alternatives before any changes are made to the real-world system (Chang and Makatsoris, 2002).

The application of simulation to supply chain and logistics problems is becoming more common as supply chains grow more complex (Terzi and Cavalieri, 2004). Computer simulation is extremely helpful in making logistics decisions because simulation can analyze operational problems that cannot be solved by traditional analytic methods (Schunk and Plott, 2000). According to Vieira (2004), simulation can generally be defined as "the activity of imitating the behavior of a system through the creating of a computer mathematical model." The following report details the analysis of a computer simulation model designed to mimic the activities of a particular company's distribution system.

B. Company History

This report focuses on a nationwide distribution company. The company has 19 branches warehouses spread throughout the country and one central distribution center (DC) located in Louisville, KY as shown in Figure 1. Currently, the company maintains over 250,000 square feet of warehouse space to store 9000 different products.

FIGURE 1 – Warehouse Locations

The distribution system consists of approximately 100 active vendors, 19 branch warehouses, and one central distribution center, which can also act as a branch warehouse. When a customer places an order at a branch, that branch either supplies the SKUs (stock keeping units) from existing inventory or places an order for that product if it is not available. Inventory is monitored by a computerized tracking system, which records all orders and replenishments in real time.

Products are replenished by ordering either from the vendor or from the distribution center. Each SKU at each branch has a preset reorder point, so that when the

inventory level for that SKU dips below the reorder point, an order is placed with the vendor or the DC. When a SKU is ordered from the vendor, a Class Method is used. This method basically categorizes each SKU into one of six classes based on their expected demand. Since vendors typically have a minimum total order size, additional SKUs from the product line are normally added on to the order based on their Class number. In general, lower class numbers mean that the buyer is allowed greater flexibility in adding those SKUs to an order.

On the other hand, when a SKU is ordered from the DC, an (*s,S*) inventory policy is used. For this policy when the inventory level dips down to or below its reorder point "*s*", an order is placed of an appropriate size so that the total inventory level will reach the "order up to" level, or *S*. There are no minimum order quantities when ordering from the DC, so adding additional SKUs is not required. It should be noted that the DC always uses the Class method, since it can only order from the vendors. In most cases, the class method results in higher inventory levels than the (*s,S*) method since the lead times from the vendors are typically much larger than the lead times from the DC.

One of the main issues for the company was the branches' policies of which products to order from the DC and which to order directly from the vendor. Each branch loosely follows their current Authorized Replenishment Plan (ARP) when making the decisions about which products to order from which source. The ARP is a comprehensive listing for each branch and each product line, based on historical data and general preferences. However, the ARPs are not always reliable and the branches do not necessarily follow the ARP, so it was hypothesized that considerable improvements could be made if the Authorized Replenishment Plans are fully analyzed. This report will

evaluate the current ARPs as well as propose new replenishment plans that will significantly increase the company's annual profit.

The purpose of this report is to use computer simulation to suggest the optimal ordering strategies for the company's distribution system. The remainder of the paper is organized as follows: The next chapter describes the simulation model of the distribution system. Chapter three details the procedures for generating the input data files required for the simulation. Analysis and results of the simulation runs are contained in the fourth chapter. Finally, the last chapter includes conclusions and future research topics.

II. SIMULATION DESCRIPTION

The analysis process began by gathering data and information about the current operation of the distribution network. It quickly became apparent that simple formulas and methods would not be enough to adequately represent the complexities of the system. At that point, Dr. Gerald Evans of the University of Louisville developed a computer simulation model to represent the system, using the Arena Software Package (Kelton, Sadowski and Sturrock, 2007).

Computer simulation has a number of advantages over traditional analysis, namely its ability to quickly and accurately compute the performance measures of the real-world system with any desired inputs. "Simulation is the process of designing and creating a computerized model of a real or proposed system for the purpose of conducting numerical experiments to give us a better understanding of the behavior of that system" (Kelton, Sadowski and Sturrock, 2007, 7). For this particular company, a complete simulation model will allow them to study the potential effects of small changes on the entire distribution system.

A. Simulation Submodels

The simulation model consists of nine submodels, each of which performs a separate function of the simulation. The variables and attributes are then passed between the different submodels. The nine submodels are described briefly below, but the reader

is referred to the CELDi Project Report (Evans, 2006) for a comprehensive description of the simulation model.

1. Read Input Data – before the other submodels begin execution, this submodel reads the Excel input files and translates the data into the appropriate variables and attributes.
2. Compute Initial Variable Values – immediately after the input data are read, this submodel uses these variables to compute the initial values for many of the variables such as reorder points, usage rate for the DC, and initial inventory levels. Since a warm-up period of 90 days is allowed, these initial values are recalculated 90 days into the simulation to represent the system at steady-state. All other summary variables are calculated using these steady-state values.
3. Day of Week and Day of Simulation Update – increments the day of the week and the day of the simulation variables at the beginning of each 8-hour simulated day.
4. Decrease Inventory Levels by Customer Demand and Modify Order to DC and Order to Vendor Variables – updates the inventory levels and order variables based on customer demands to simulate actual inventory decreases and order replenishment. This submodel also decides which SKUs are below their reorder points and must be ordered.
5. Generate Order Entities to DC and Vendors – this submodel creates the orders from each branch to the vendor and the DC immediately after the previous submodel concludes. The submodel also calculates the associated costs and decides if the Algorithm to Form Minimum Size Vendor Order submodel needs to be executed.

6. Algorithm to Form Minimum Size Vendor Order – this is a submodel contained within the Generate Entities to DC and Vendors submodel. When an order is placed with the vendor that does not meet minimum order requirements, this submodel determines which additional SKUs should be added to the order. The decisions on which SKUs to add are based on their class numbers and the Fraction Extra Order variable.
7. Schedule Shipments to Depart DC and Arrive at Branches – schedules when shipments should be sent from the DC according to the orders placed and the preset shipping schedule. This submodel also keeps track of whether there is sufficient inventory to fulfill the orders.
8. Schedule Shipments to Depart Vendors and Arrive at Branches – updates the costs for shipments and schedules when shipments should be sent from the vendors. The submodel also updates inventory when an order is received.
9. Output Variable Computation – This submodel is executed at the end of the simulation run to calculate all of the output variables such as shipping costs, inventory costs, ordering costs, and total profit. These output variables are the performance measures that are evaluated to compare different scenarios against one another.

The simulation model consists of both controllable and uncontrollable variables, as well as the output variables. The most important independent variables which can be changed include: Fraction of Orders Made to the DC (where vendor orders =0, and DC orders =1), Fraction Extra Order, and Safety Factor. Many of the other parameters are kept constant throughout all of the simulation runs, including lead times, the DC to

branch shipping schedule, the reorder points and the order quantities for each SKU. While the main performance measure used to gauge the effectiveness of each combination of inputs is the net profit, some of the other outputs of interest include ordering cost, shipping cost, total sales, lost sales, ending inventory at each branch, and inventory holding cost. Each of these performance measures are computed at the end of every simulation run, and saved for later comparisons.

B. Vendor Selection

The company provided data on the total annual demand for all current SKUs. Any specific SKU is ordered from only one particular vendor, so all SKUs for a particular vendor can be grouped together and summed to find the annual demand for each vendor. Since it is impractical to perform a full analysis on all 360 available vendors, the vendors with the highest amount of annual demand were chosen for further consideration. Figure 2 below shows a graph of only the top 100 vendors, which clearly demonstrates that the Pareto principle is in effect.

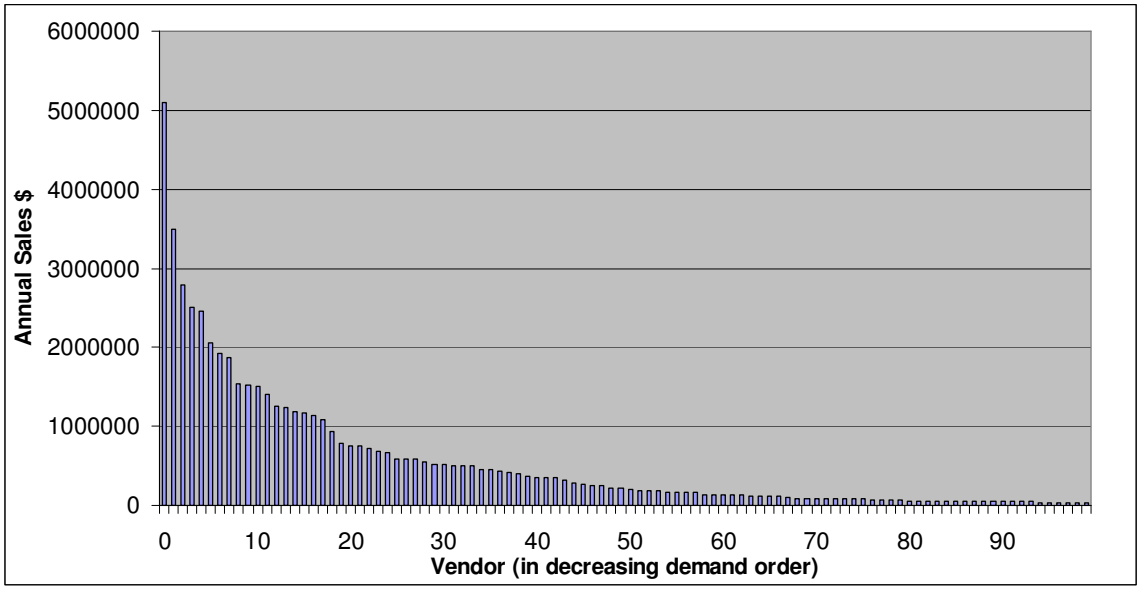

FIGURE 2 – Top 100 Vendors Annual Sales

Using this theory that 80% of the demand is attributed to only 20% of the vendors, it was decided to focus the analysis on the top 25 vendors, as determined by their annual demand. The remaining vendors contributed less than 1% of the total demand each. Table I shows the vendors chosen with their respective demand dollars and percent of total demand.

TABLE I

TOP 25 VENDORS BY TOTAL DEMAND

Vendor ID	demand by vendor	% of total demand
82	$5,099,462.85	8.9412%
130	$3,488,097.56	6.1159%
264	$2,783,090.96	4.8798%
217	$2,511,593.69	4.4038%
26536	$2,461,254.89	4.3155%
206	$2,047,851.89	3.5906%
235	$1,879,517.98	3.2955%
18	$1,533,588.69	2.6889%
234	$1,518,224.99	2.6620%
88	$1,498,201.61	2.6269%
227	$1,402,965.68	2.4599%
157	$1,260,850.83	2.2107%
10	$1,230,713.86	2.1579%
278	$1,189,517.63	2.0857%
71	$1,163,387.35	2.0398%
244	$1,141,587.26	2.0016%
220	$1,082,202.47	1.8975%
22	$789,899.38	1.3850%
242	$754,290.64	1.3226%
48	$744,177.99	1.3048%
125	$722,599.83	1.2670%
27511	$680,790.77	1.1937%
261	$591,842.15	1.0377%
27692	$579,581.18	1.0162%
11	$578,905.75	1.0150%

For each of the selected vendors, further analysis was performed to identify the most important product line for each vendor and the top-selling SKUs within that product line. Each vendor can supply between one and eight different product lines, with one product line typically having significantly higher demand than the others. Thus, one product line was selected for each vendor for additional research. Additionally, each product line may contain up to 830 different SKUs, many of which are seldom sold and therefore will have little effect on the final simulation results. As a result, only SKUs which had a total demand greater than $1000 were included in the simulation analysis described later in the report. These modifications to the initial raw data gave a total of 25

vendors, 25 product lines, and almost 1500 SKUs between the vendors. Analysis of only this modified data set will greatly reduce the time required to generate the input files and run the simulation model, while still showing an accurate picture of the current operations at Glantz. Analyzing this set of data will also demonstrate the magnitude of improvement that will be possible upon implementing the changes recommended later in this report.

III. INPUT FILE PROCEDURES

The simulation model was designed for maximum flexibility, so that all possible scenarios can be accurately modeled. To ease repetition of subsequent runs, the simulation uses 11 input data files, which contain most of the controllable inputs to the model that are kept constant over the simulation run. The model reads in each spreadsheet file at the start of the run, and assigns the data to the appropriate variables and attributes within the model as described in the submodels above. Using independent files allows the analyst to make changes to one or more data points, and then quickly generate updated input files for all vendors.

A. Input File Methodology

After evaluating the input file requirements, it was determined that Microsoft Excel Visual Basic for Applications would be the most appropriate tool for generating the input files. Excel VBA macros allow easy spreadsheet manipulation so that the file generation process can be automated. Of the 11 required input files, four stay the same for all vendors for the purposes of the current analysis and can therefore be copied for each vendor and do not require any further programming. The remaining seven files each required a separate macro, which will be referred to as the "Master" file used to generate the input files for each vendor. The macros search, modify, and perform calculations on raw data provided by the company in order to construct the proper file formats. The

input files for each vendor, as well as the macros that can be used to generate input files for additional vendors, were provided to the company with documentation on their use.

A similar structure was used for each of the macros, so that the user simply needs to select the input files to create, enter the desired vendor number, and start the program. Each macro is designed to automatically create the input file, save it in the corresponding vendor folder, and close the newly created file(s) so that the user can select another vendor and begin the process again. Ten of the eleven macros were combined into a single master input file to facilitate input file generation. As seen in Figure 3, the user can select any of the input files on the left, which will all be created automatically. The Customer Demand file was not included in this group because of file size considerations.

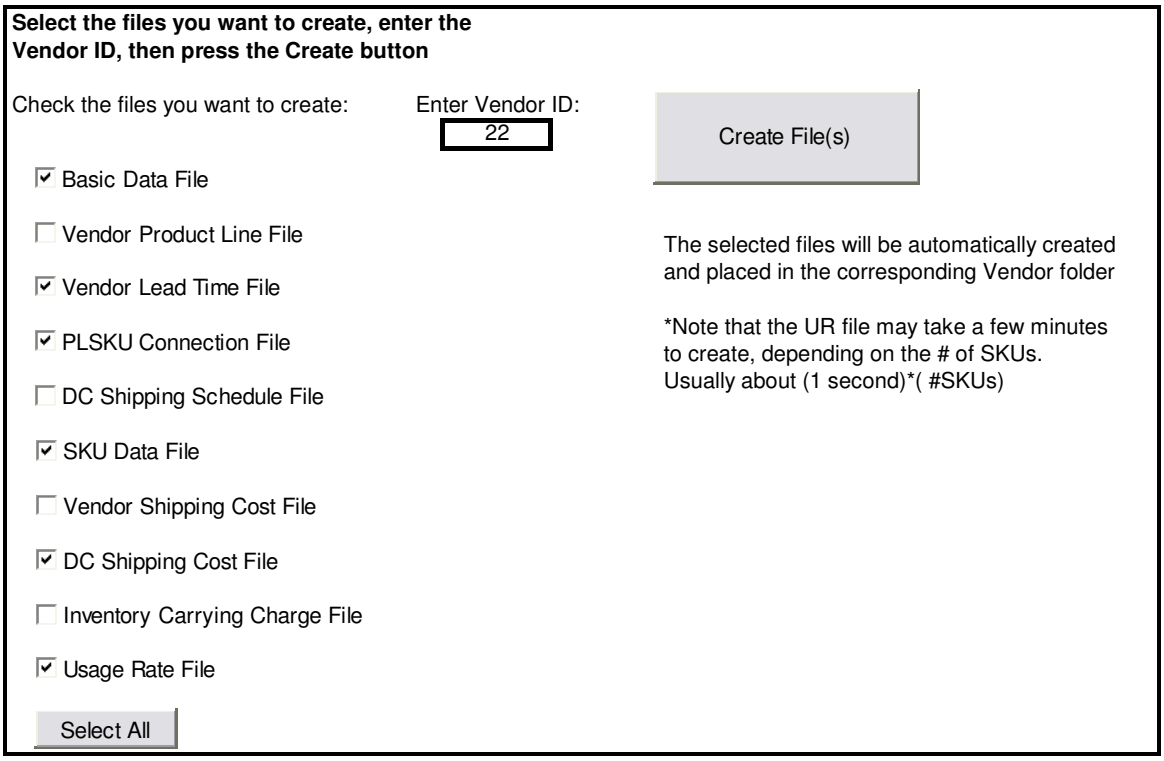

FIGURE 3 – Master Input File User Interface

The following descriptions show the input file structure for all input files for one vendor. These descriptions apply to all vendors used in the analysis, but do not apply to all possible scenarios that the company could need to model in the future. For instance, the input files were designed for only one product line for each vendor, so the company may need to alter the macros in the future if multiple product lines for a vendor are necessary. All of the input file macros can be seen in Appendix A.

B. Input File Descriptions

First, each vendor requires a Key Number Identification File before the other files can be generated. It is not technically an input file since it is not read directly by the simulation model, and must be created manually after the analyst makes the decisions on which product line and SKUs should be included. The file contains a list of the SKUs, product lines, and branches associated with that vendor. Each SKU is given a "key number", which is used throughout the input files and within the simulation to refer to that SKU instead of repeatedly using the full SKU name. The key number files are also used in the generation of several of the other input data files.

Each of the next three files uses a manually created data spreadsheet in order to generate the input files. The data was combined into a single spreadsheet because there is some overlap between the three files. A sample of the beginning of the sheet can be seen below in Figure 4, and the entries are explained in the input file sections to follow.

Vendor ID	# SKUs	Lead Time	Min Buy Amt	Min Type
10	64	8	3000	1
11	17	20	1200	1
18	203	6	1750	1
22	12	8	1500	1
48	27	14	2500	1

FIGURE 4 – Reference Data Spreadsheet

1. The first file is the Basic Data file (BD.xls), which gives the parameters on how the simulation should run and fundamental data about the vendor such as number of SKUs and basic costs. The Basic Data macro looks at the data spreadsheet to determine the number of SKUs for that vendor; all other values are kept constant for the purpose of this analysis. The basic functions of the macro are to save the # of SKUs, copy the BD template into a new workbook, and enter the appropriate # of SKUs. The company can easily modify this macro in the future if additional variables are needed.
2. Next is the Vendor Lead Time file (VLT.xls), which is a set 20 entries, each corresponding to the lead time from the vendor to one of the branches. For this analysis, the lead times are assumed to be the same for all branches. The lead times for each vendor were gathered previously (Chiodi, 2006, 64), and are stored in the data spreadsheet. The VLT macro reads the lead time from the Master file then creates a new workbook. In the new workbook, the vendor key number (1 for this analysis) is entered in the first cell, and then the lead time is entered in the next 20 cells.
3. The Product Line SKU Connection file (PLSKUCon.xls) is slightly different from the previous two files in that the number of entries is not constant. The number of entries is equal to (Number of SKUs + 1). The first line refers to the Product Line Number (always 1), the Minimum Buy Amount, the Minimum Buy Type, and the Number of SKUs in the product line. The remainder of the first column is then the key numbers of the SKUs from 1 to # of SKUs + 1. Although the simulation

model has the capability to make ordering decisions based on weight, that data was not available for all SKUs in any of the product lines, so for this analysis all of the Min Buy Types are equal to 1 (where 1 corresponds to dollar value, and 2 corresponds to weight of the SKU). For vendors that were not identified in any of the raw data spreadsheets, the Min Buy Amt was assumed to be $1000 and the Min Type set to 1. The macro essentially copies each of the variables from the master spreadsheet, enters them into the appropriate cells on a new workbook, and then creates a series from 1 to # of SKUs. Before saving and closing the new workbook, the macro creates a variable range up to the number of SKUs and adds this range as a named range.

4. The SKU Data file (SKU D.xls) uses a company-provided spreadsheet with the complete list of stocked products and the Key Number Identification File in order to output the required data for each SKU. The SKU D file includes the following variable information on each SKU: Key Number, Class Number, Lost Sales Cost (set to 0.01 * purchase price), Sale Price (set to 1.5 * purchase price), and Purchase Price. The remaining values in the SKU D file are kept constant for the purposes of this analysis.

   At the beginning of the SKU D macro, the program checks if the stocked product spreadsheet has already been modified. If so, the program jumps down to begin generating the SKU D file, but if not the spreadsheet must be formatted and modified. These modifications only need to be performed when the raw data spreadsheet is updated or changed, and can then be saved in the proper format for later use. To begin the modifications, this macro first deletes any references to

the Atlanta branch, because that branch has recently opened and sufficient data is not yet available for analysis. Next, the macro adds a new column for a "pseudo-Class" number because many of the SKUs do not have the same Class number listed for all branches. Since the simulation model assumes that a particular SKU will have the same Class number at all branches, the lowest Class number found in the raw data spreadsheet should be used for that SKU. The macro sets the pseudo-Class number for each entry equal to the minimum of the Class number of the entry below (if that entry corresponds to the same SKU) and its own Class number. The end result is that the first entry for each SKU has a pseudo-Class equal to the lowest Class number listed for that SKU over all branches.

Once these modifications are completed, the macro moves to the Key Number Identification sheet for that vendor and deletes the references to the branches and then identifies how many SKUs are in that product line. To accommodate the final format of the SKU D file, a blank line is added between each SKU. Next, the lookup formulas for Class Number and Purchase Price, as well as the computed formulas are entered for the first SKU. For any SKUs that are not found in the raw data spreadsheet, the Class Number is set at a default of 6. Then a slightly altered set of formulas is entered for the second SKU so that the Purchase Price for any SKUs not found in the raw data spreadsheet will default to the Price value of the preceding SKU. The formulas are copied down for all SKUs, and the entire range is copied and pasted using PasteValues so the lookup formulas will not error when the SKU names are deleted. After the SKU

names column is deleted, the sheet is copied to a new workbook, a named range is added, and the workbook is saved and closed.

5. The DC Shipping Cost file (DCSC.xls) essentially looks at the company's sales records and averages the percent shipping costs over all SKUs for each branch. Since the number of data entries on the company-provided spreadsheet exceeds the capacity of one sheet in Excel, the raw data is split into three separate sheets and grouped by vendor number. The macro first navigates to the appropriate sheet based on the vendor number entered. Next, the macro creates a small table on that sheet with the vendor number and branch names. To find the average shipping cost percent, a formula is entered using the *sumproduct()* function twice to add up all of the shipping percentages for all entries on the sheet which refer to the relevant vendor and the specific branch, then divide by the number of such entries that are found. This formula is copied down for all 19 branches (there is no shipping cost to branch 20 since this refers to the DC acting as a branch, and therefore cannot ship to itself). After this, the small table is cut and pasted into a new workbook and the columns containing the vendor number and warehouse names are deleted. Finally, the named range is set and the new workbook is saved and closed.
6. Next is the Usage Rate file (UR.xls), which details the average usage of each SKU at each branch over the last six months. This macro again uses the stocked product spreadsheet provided by the company and the vendor Key Number Identification File when creating the input files. To begin, this macro copies the Key Number Identification sheet to a new sheet and deletes the branch names as

well as the Key Numbers themselves. Next, each SKU name is copied and inserted 20 times between each SKU reference, corresponding to the 20 branches. Then the formulas are entered using the *sumproduct()* function of Excel to find the Usage Rate for each SKU at each branch by searching the raw data spreadsheet. Since copying the formulas down for all SKUs at once requires more computational capacity than is available, the formulas for the SKUs are copied down one at a time with PasteValues being used after each section is copied. While this technique does require more time, it prevents the macro from crashing when there are a large number of SKUs in the product line. Finally, the SKU names are deleted, the sheet is moved to a new workbook, a named range is set, and the file is saved and closed.

7. The last input file macro is for the Customer Demand file (CD.xls). This is the only file not included in the Master input file described above. Generating this file requires a complete one year history of the customer demands from 11/1/04 to 10/31/05, which is much too large to fit on a single sheet. Due to the large file size and the time required to generate each file, it was decided to keep the CD master file separate.

   The Customer Demand input file is a complete listing of all customer demands for a one year period. This one year history is used in the simulation model instead of stochastic demand distributions due to the high variability associated with many of the slow-moving SKUs. To create the CD file, the macro basically sorts the demand data and totals the demand for each branch on any given day, then repeats this for every day. First, the macro searches through the

ten sheets of raw demand data to compile a new sheet with all references to the relevant vendor.

The CD file format has two sections for each day of the simulation, in addition to the date key. The first section is a list for each of the 20 branches showing the total number of SKUs ordered at each branch. The second section is a list in order by branches, with the SKU key number and the quantity of that SKU ordered.

The macro begins by deleting the unnecessary columns from the demand file. Then it creates a reference table for the warehouses by key number, so that a new column can be inserted for the warehouse keys for all of the demand entries. This allows the worksheet to be sorted by warehouse key, which is necessary for the final format of the CD file. As before, any references to the Atlanta branch are deleted and are not considered in the analysis. Next, columns are made for the date keys and SKU keys; where a date key of 1 is equivalent to 11/1/04 and a date key of 10/31/05 is equivalent to 365; and the SKU keys are found in the Key Number Identification sheet. After sorting the sheet by SKU key, any entries that do not have a valid SKU key are deleted, meaning that any SKUs not selected for analysis will not be part of the customer demand file. The macro then reformats the sheet by copying and pasting using PasteValues, deleting unnecessary columns and rearranging the order of the columns.

The next portion of the macro begins entering the formulas to calculate the total demand by branch for each day. This section loops through each day, up to 365. Within the loop, the macro finds the first entry corresponding to the current

date key, and inserts 21 rows above that entry. In the first of those 21 new rows, it enters the current date key. Then the macro fills down the warehouse keys from 1 to 20. Next the formulas are entered to total up the number of SKUs ordered at each branch using the *sumproduct()* function in Excel. This function allows the macro to search through the sheet to find all entries for that day and branch combination. Since there is no set number of entries for any date, the macro searches the next 300 entries, which is sufficiently larger than the greatest number of orders for any one day. Before going back to the beginning of the loop, all of the formulas entered are copied and pasted using PasteValues to prevent Excel from re-computing the values whenever more formulas are added, thus preventing a program crash. The date key is increased by one, and the program loops back to the start of the formulas portion. Finally, the date key and quantity columns are deleted, the sheet is moved a new workbook, a named range is set, and the new workbook is saved and closed.

The remaining four input files are simply copied over for each vendor as stated previously; therefore the macros for these files only copy and save. All of the input files were successfully generated for the 25 vendors to be studied.

IV. ANALYSIS AND RESULTS

With the input file generation completed, the next step in the analysis process was to decide on appropriate values for the variables that would be held constant throughout the simulation runs. Since a full validation of the simulation model for one vendor was completed by Dr. Gerald Evans, most of this process was already complete. To check the suitability of the chosen constants, two more vendors were selected for a complete analysis. A full analysis was run on both vendors while varying some of the key parameters such as Safety Factor, % Markup from Purchase Price, and Fraction Extra Order. The results of these analyses were compared with the expected values and appropriate adjustments were made to these parameters.

A. Simulation Methodology

The main decision variable for this analysis was the Fraction of Orders Made to the DC. The goal was to provide the company with a new Replenishment Plan for each vendor that determines which branches should order from the vendor and which should order from the DC. To get an accurate picture of the current state of the company's distribution system and the potential improvements, four different scenarios were run for each vendor:

1. All branches order from the DC
2. All branches order from the vendor
3. Authorized Replenishment Plan

4. Optimal Solution, as determined by the OptQuest optimization package (Kelton, Sadowski, and Sturrock, 2007).

The purpose of the first two scenarios was to give the company baseline values for comparing the other scenarios. The ARP scenario should correspond to the current state of operations, and the final scenario is the optimal solution as determined by OptQuest for Arena. The following analysis focuses on the differences between the ARP scenario and the Optimal solution, as a means of gauging the potential improvements that can be realized if the company implements the proposed changes.

For large simulations, the number of variables often makes it infeasible to search all of the possible combinations of variables, and inefficient to randomly search the sample space. Analysts therefore turn to specially designed optimization tools to intelligently search large sets of alternatives for the optimal, or near optimal, solution. OptQuest is the optimization package included in Arena, and is used in the analysis to follow. QptQuest employs the metahueristics of Tabu search, neural networks, and scatter search to quickly converge on the best solution (Bapat & Sturrock, 2003). With 19 Boolean decision variables, there are $2^{19} = 524{,}888$ possible solutions. At one second to explore each solution, it would take more than 6 days to consider all possibilities. Alternatively, the efficient techniques used by OptQuest allow you to find a near optimal solution within two minutes in most cases. OptQuest is a very powerful tool, allowing the analyst to change input variables and performance measures, add constraints, or suggest starting solutions. For the remainder of this report, the solutions referred to as "optimal" should be interpreted as "optimal, or near optimal" since the OptQuest solution does not guarantee the single optimal answer.

B. Simulation Results

Net profit was chosen as the performance measure used to compare the results of the simulation runs. The net profit is a calculated value based on Sales dollars, Inventory Value, Lost Sales Cost, Ordering Cost, Shipping Cost, Purchasing Cost, and Inventory Carrying Charges. Since net profit is calculated at the end of the simulation run and combines all of these parameters of interest, it is the best measure of the actual performance of the system.

As stated previously, four different scenarios were run for each vendor using an identical simulation model and the chosen constants. Appendix B shows the full results of the simulation runs for each vendor. If the company were to implement the optimal policy instead of the current ARP for all 25 vendors, the anticipated nine month profit increase would be $629,500, which corresponds to an 11% increase in net profit. If these results are extrapolated for a full year, the estimated profit increase would be $839,300. Table II shows the differences between the ARP and the proposed optimal solution for each vendor.

TABLE II

NET PROFIT COMPARISON

Vendor	Net Profit: ARP	Net Profit: Optimal	Difference: Optimal - ARP	% Difference from ARP
10	$ 140,020	$ 143,270	$ 3,250	2.32%
11	$ 67,313	$ 73,098	$ 5,785	8.59%
18	$ 26,592	$ 32,885	$ 6,293	23.67%
22	$ 155,680	$ 162,160	$ 6,480	4.16%
48	$ 210	$ 33,703	$ 33,493	15916.25%
71	$ 103,690	$ 115,170	$ 11,480	11.07%
82	$ 694,730	$ 792,050	$ 97,320	14.01%
88	$ 30,558	$ 41,624	$ 11,066	36.21%
125	$ 134,640	$ 135,920	$ 1,280	0.95%
130	$ 734,090	$ 753,350	$ 19,260	2.62%
157	$ 215,530	$ 237,710	$ 22,180	10.29%
206	$ 209,150	$ 238,060	$ 28,910	13.82%
217	$ 228,040	$ 252,600	$ 24,560	10.77%
220	$ 144,620	$ 152,930	$ 8,310	5.75%
227	$ 67,887	$ 94,890	$ 27,003	39.78%
234	$ 370,560	$ 375,920	$ 5,360	1.45%
235	$ 187,450	$ 218,070	$ 30,620	16.34%
242	$ 123,750	$ 157,350	$ 33,600	27.15%
244	$ 74,565	$ 81,990	$ 7,425	9.96%
261	$ 75,843	$ 78,226	$ 2,383	3.14%
264	$ 485,960	$ 529,500	$ 43,540	8.96%
278	$ 114,630	$ 189,820	$ 75,190	65.59%
26536	$ 399,800	$ 399,800	$ -	0.00%
27511	$ 715,670	$ 811,760	$ 96,090	13.43%
27692	$ 182,100	$ 210,720	$ 28,620	15.72%
Total:	$ 5,683,078	$ 6,312,576	**$ 629,498**	11.08%

These results were found to be not strongly correlated with the approximate annual demand from Table I. Since this analysis considers only the top 25 vendors, the total profit change would be increased slightly by considering many of the other active vendors.

C. Ordering Policy Analysis

Based on the initial experimentation with one vendor, the previous hypothesis was that in most cases, all branches should order from the DC except for the California branch

and the DC itself. The results below prove that there are no absolute rules that apply for all product lines, but that in general more branches should order directly from the vendor than is currently done using the ARP. Table III shows the top-level trends for both the ARP and the optimal policies. Note that this table shows, for each branch, the number of product lines ordered through the DC, up to the maximum of all 25 vendors ordered through the DC. The sum of the values over all branches represents the total number of product lines ordered from the DC for each policy.

TABLE III

TOTAL PRODUCT LINES ORDERED THROUGH DC

Branch		Number of Product Lines Ordered through DC	
		ARP	Optimal
1	Buffalo	18	17
2	Tulsa	17	12
3	NC	18	16
4	Cleve	20	12
5	Pitt	19	13
6	Dallas	15	8
7	KC	18	7
8	Philly	16	11
9	Indy	18	12
10	Cal	9	4
11	Milw	17	10
12	Minn	17	14
13	Md	19	12
14	SA	17	11
15	Norf	18	18
16	NJ	14	9
17	Orl	14	8
18	Bkln	14	8
19	Phnx	19	12
20	Louisville	0	0
	Sum:	317	214

As Table III shows, no branch orders from the vendor for all product lines or none of the product lines, but there are general patterns in the ordering policies between ARP and optimal. The overall trend is that generally ordering from the vendor more often than

was planned using the ARP will lead to increased profits. In fact, the optimal policies itemized by vendor suggest that 21 of the 25 vendors show a decrease in the number of branches that order from the DC when comparing the optimal solutions to the ARPs. Full enumeration of the ordering policy results can be seen in Appendix C.

Considering the combined policies for all vendors gives a total of 500 binary variables: one for each branch/vendor combination. Of these variables, 63% (317) correspond to ordering from the DC for the ARP, but only 43% (214) for the optimal policies (Table IV). Although one cannot draw conclusions from these results about the exact proportion of orders that should be made to the DC, it does show a significant trend towards ordering more product lines from the vendor.

TABLE IV

CHANGES FROM ARP TO OPTIMAL

	Number of Branches Ordering through DC		Number Changed from ARP to Optimal	
Vendor	ARP	Optimal	1 to 0	0 to 1
10	15	13	2	0
11	17	10	7	0
18	17	5	12	0
22	18	14	5	1
48	16	11	6	1
71	17	4	13	0
82	15	8	8	1
88	19	10	9	0
125	6	6	4	4
130	6	2	5	1
157	15	9	8	2
206	18	11	7	0
217	16	9	8	1
220	18	15	3	0
227	19	8	11	0
234	4	6	1	3
235	12	9	6	3
242	9	6	5	2
244	7	9	4	6
261	11	5	6	0
264	0	13	0	13
278	8	13	2	7
26536	2	2	0	0
27511	14	10	5	1
27692	18	6	12	0
Total:	317	214	149	46

Further analysis of the ordering policies for each vendor/branch combination show that of the 195 individual variables that changed from the ARP to the optimal, 149 (76%) were changed from 1 (DC order) to 0 (vendor order) instead of the opposite, as shown in Table IV. This means that only 24% of the time does the optimal ordering policy recommend a change from ordering from the vendor to ordering from the DC. This analysis also supports the previously stated conclusion that the company should place more orders through the vendors than they currently plan.

However, one aspect of the initial hypothesis was validated since the California branch (branch 10) had the lowest number of vendor orders in both the ARP and the optimal plan. This seems logical since that branch is farthest away from the DC, but still it is not an absolute rule because the optimal policies dictate that California should still order through the DC for 4 of the 25 vendors.

As a whole, these results are surprising since previous experimentation with one vendor showed that 18 of the 19 branches should order through the DC. Real-world tests are needed to fully validate the results, so it is recommended that the company implement the new optimal replenishment plan for a single vendor to compare the actual performance of the policies. Further research can also be performed to explore other alternatives, as discussed in the following chapter.

V. CONCLUSIONS

Several conclusions can be drawn based on the analysis discussed above. First, the company has the potential to significantly increase their profits by changing their ordering policies to the optimal solutions. Aside from the time required to distribute the revised replenishment plans to all branches, there is no cost associated with implementing these changes. Therefore, the company will gain the entire net profit of $629,500 with no initial investment.

A second conclusion is that in general, the company should order more product lines from the vendors, while still striking a balance between vendor orders and DC orders. There are no absolute rules that apply for ordering policies at any branch or for any vendor; each individual branch/vendor combination must be considered.

Another strong conclusion is that this level of analysis would have been impossible without a detailed simulation model. Previous analysis without a simulation model suggested inconclusively that ordering from the DC was superior to ordering from the vendor for most cases (Chiodi, 2006). The distribution system is such a complex and dynamic system that a mathematical analysis could never properly capture the full scope of the system.

There are several areas that could be researched in the future to provide the company with new strategies that could improve their operation. One topic could be experimentation with alternative ordering policies and/or drastic changes to the current

policies. For instance, an (s,S) policy could be implemented for both orders to the vendor and orders to the DC. It may be possible to aggregate SKUs together to create subgroups within a product line that are always ordered in the same ratios so that minimum order size constraints of the vendors are met.

Another future consideration may be revising the shipping schedule from the DC to the branches. Changes to the frequency and sequence of shipments could help the company improve service level and decrease inventory. Additionally, more complete data collection could be done by the company to better refine the simulation model in areas such as lost sales, lead times, inventory carrying charges, and ordering costs where assumptions were made in the current model for parameters where the company does not currently keep records.

With such a large system, it is rare for one person to have a thorough understanding of the entire system; so simply the exercise of gathering the required data can open the company's eyes to the wide range of parameters that affect their distribution system. In this respect, the exercise of helping the analysts create the simulation model itself helped the company understand their own distribution system, even before the analysis was completed. With the growing competition among logistics providers, companies can no longer afford to be uneducated about their own systems (Vieira, 2004).

REFERENCES

Bapat, V. and Sturrock, T. 2003. The Arena Product Family: Enterprise Modeling Solutions. *Proceedings of the 2003 Winter Simulation Confernce.* 210-217.

Chiodi, Maria D. 2006. An Integrated Analysis of a Third Party Logistics Company's Ordering Policies and Operations. Master of Engineering thesis, University of Louisville.

Chang, Y. and Makatsoris, H. 2002. Supply Chain Modeling Using Simulation. *International Journal of Simulation.* 2(1): 24-30.

Chopra, S. and Meindl, P. 2007. *Supply Chain Managament, Third Edition*. New Jersey: Pearson Education.

Evans, G.W. 2006. CELDi Project Report. Department of Industrial Engineering, University of Louisville.

Evans, G.W., DePuy, G.W., Usher, J.S., Chiodi, M. 2006. Distribution System Operation Using Simulation and Criterion Modeling. *Proceedings of the 2006 Industrial Engineering Research Conference.*

Evans, G.W., DePuy, G.W., Gupta, A. 2007. Simulation and Optimization Methodologies to Determine Distribution System Inventory Policies. *Proceedings of the 2007 Industrial Engineering Research Conference.*

Jain, S., Workman, R.W., Collins, L.M., Ervin, E.C., Lathrop, A.P. 2001. Development of a High Level Supply Chain Simulation Model. *Proceedings of the 2001 Winter Simulation Conference.* 1129-1137.

Kelton, W.D., Sadowski, R.P., Sturrock, D.T. 2007. *Simulation with Arena, Fourth Edition.* New York: McGraw-Hill.

National Association f Manufacturers. "U.S. National Data." NAM, February 2007, available from http://www.nam.org/s_nam/sec.asp?CID=202325&DID=234949; accessed April 2007.

Schunk, D., and Plott, B. 2000. Using Simulation to Analyze Supply Chains. *Proceedings of the 2000 Winter Simulation Conference.* 1095-1100.

Shapiro, J.F. 2001. *Modeling the Supply Chain*. California: Wadsworth Group.

Simchi-Levi, D., Chen, X., Bramel, J. 2005. *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management.* New York: Springer Science+Business Media

Suwanruji, P. and Enns, S.T. 2004. Evaluating the Performance of Supply Chain Simulations with Tradeoffs Between Multiple Objectives. *Proceedings of the 2004 Winter Simulation Conference.* 1399-1403.

Terzi, S and Cavalieri, S. 2004. Simulation in the Supply Chain Context: A Survey. *Computers in Industry.* 53(1): 3-16.

Vieira, G.E. 2004. Ideas for Modeling and Simulation of Supply Chains with Arena. *Proceedings of the 2004 Winter Simulation Conference.* 1418-1427.

Appendix A - Input File Generation Macros

```
Public Sub cmdMake_Click()
'Runs the subroutines for the files selected

    If chkBD = True Then
        BD_make
    End If
    If chkVPL = True Then
        VPLcopy
    End If
    If chkVLT = True Then
        VLT_make
    End If
    If chkPLSKUCon = True Then
        PLSKUCon_make
    End If
    If chkDCSS = True Then
        DCSScopy
    End If
    If chkSKUD = True Then
        SKUDmake
    End If
    If chkVSC = True Then
        VSCcopy
    End If
    If chkDCSC = True Then
        DCSCmake
    End If
    If chkInvCC = True Then
        InvCCcopy
    End If
    If chkUR = True Then
        URmake
    End If
End Sub
```

```
Public Sub cmdSelectAll_Click()
'selects all check boxes

    chkBD = True
    chkVPL = True
    chkVLT = True
    chkPLSKUCon = True
    chkDCSS = True
    chkSKUD = True
    chkVSC = True
```

```
    chkDCSC = True
    chkInvCC = True
    chkUR = True
End Sub
```

```
Sub BD_master()
'Creates the Basic Data file
Dim strVendor As String
Dim strPath As String
Dim intStop As Integer
Dim intNumSKU As Integer

'Save vendor number, copy corresponding sheet
    strVendor = Range("E5").Value
    Sheets("data").Select

'Save current path
    strPath = ThisWorkbook.Path

'Find the Vendor ID on the "data" sheet
    Range("A2").Select
    intStop = 0
    Do Until intStop = 1
        If ActiveCell.Value = strVendor Then
            intStop = 1
        Else: ActiveCell.Offset(1, 0).Select
        End If
        If ActiveCell.Value = "" Then
            MsgBox ("Vendor ID not found.  Please add information on ""data"" sheet")
            Exit Sub
        End If
    Loop

'Copy # of SKUs to variable
    intNumSKU = ActiveCell.Offset(0, 1).Value

'Make sure first row is named range "Basic_Data"
    ThisWorkbook.Names.Add Name:="Basic_Data", _
        RefersTo:="=BD!$A$1:$K$1", Visible:=True

'Copy template to new workbook and change cell G1 to be # SKUs
    Sheets("BD").Copy
    Application.CutCopyMode = False
    Range("G1").Value = intNumSKU
    Range("G1") = Range("G1").Value
```

```
'Save workbook in corresponding Vendor ID folder and close
    ActiveWorkbook.SaveAs strPath & "\" & strVendor _
        & "\" & "BD.xls"
    ActiveWindow.Close

    Sheets(1).Select
End Sub
```

```
Sub VLT_master()
'Creates the Vendor Lead Time file
Dim strVendor As String
Dim intStop As Integer
Dim intLead As Integer
Dim strPath As String

'Save vendor number, copy corresponding sheet
    strVendor = Range("E5").Value
    Sheets("data").Select

'Find the Vendor ID on the "data" sheet
    Range("A2").Select
    intStop = 0
    Do Until intStop = 1
        If ActiveCell.Value = strVendor Then
            intStop = 1
        Else: ActiveCell.Offset(1, 0).Select
        End If
        If ActiveCell.Value = "" Then
            MsgBox ("Vendor ID not found.  Please add information on ""data"" sheet")
            Exit Sub
        End If
    Loop

'Store Lead Time and file path for later use
    intLead = ActiveCell.Offset(0, 2).Value
    strPath = ThisWorkbook.Path

'Create a new workbook and add Vendor key number and Lead Time
    Workbooks.Add
    Range("A1").Formula = 1
    Range("A2:A21").Formula = intLead
    Range("A1").Select

'Save in corresponding vendor folder and create named range VendorLT
```

```
    ActiveWorkbook.SaveAs strPath & "\" & strVendor _
       & "\" & "VLT.xls"
    Workbooks("VLT.xls").Names.Add Name:="VendorLT", _
    RefersTo:="=$A$1:$A$21", Visible:=True
    ActiveWorkbook.Save
    ActiveWindow.Close

    Sheets(1).Select
End Sub
```

```
Sub PLSKUCon_master()
'Creates Product Line SKU Connection file
Dim strVendor As String
Dim intStop As Integer
Dim strMinBuyAmt As String
Dim strMinType As String
Dim intNumSKU As String
Dim strPath As String
Dim intLastSKU As Integer
Dim rngNamerange As Range

'Save vendor number, copy corresponding sheet
    strVendor = Range("E5").Value
    Sheets("data").Select

'Find the Vendor ID on the "data" sheet
    Range("A2").Select
    intStop = 0
    Do Until intStop = 1
       If ActiveCell.Value = strVendor Then
          intStop = 1
       Else: ActiveCell.Offset(1, 0).Select
       End If
       If ActiveCell.Value = "" Then
          MsgBox ("Vendor ID not found.  Please add information on ""data"" sheet")
          Exit Sub
       End If
    Loop

'Save values as variables, create new workbook, enter variables
    intNumSKU = ActiveCell.Offset(0, 1).Value
    strMinBuyAmt = ActiveCell.Offset(0, 3).Value
    strMinType = ActiveCell.Offset(0, 4).Value
    strPath = ThisWorkbook.Path
    Workbooks.Add
```

```
    Range("A1").Value = 1
    Range("B1").Value = strMinBuyAmt
    Range("C1").Value = strMinType
    Range("D1").Value = intNumSKU

'Create series up to #SKUs
    intLastSKU = intNumSKU + 1
    Range("A2").Select
    ActiveCell.FormulaR1C1 = "1"
    Range("A3").Select
    ActiveCell.FormulaR1C1 = "2"
    Range("A2:A3").Select
    Selection.AutoFill Destination:=Range("A2:A" & intLastSKU), Type:=xlFillDefault
    Range("A1").Select

'Save and close the workbook
    ActiveWorkbook.SaveAs strPath & "\" & strVendor _
       & "\" & "PLSKUCon.xls"
    Set rngNamerange = Range("A1:D" & intLastSKU)
    Workbooks("PLSKUCon.xls").Names.Add Name:="PL_SKU_Connection", _
    RefersTo:=rngNamerange, Visible:=True
    ActiveWorkbook.Save
    ActiveWorkbook.Close

    Sheets(1).Select
End Sub
```

```
Sub SKUDmaster()
'Creates the SKU Data file
Dim blnAtlantaCheck As Boolean
Dim dblLastAtlanta As Double
Dim dblEnd As Double
Dim strVendor As String
Dim intNumSKU As Integer
Dim intNumRow As Integer
Dim blnCheck As Boolean
Dim strPath As String
Dim rngNamerange As Range

'Save vendor number
    strVendor = Range("E5").Value

'Check if icswlstflds sheet has already been prepped, if not, reformat it
    Sheets("icswlstflds").Select
    If Range("Q1").Value = "" Then
```

```
'Save the number of entries for later use
    Range("A5").Select
    Selection.End(xlDown).Select
    dblEnd = ActiveCell.Row

'Sort by Warehouse, then delete any entries for Atlanta
    Range("A3:P" & dblEnd).Select
    Selection.Sort Key1:=Range("A3"), Order1:=xlAscending, Header:=xlNo, _
        OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, _
        DataOption1:=xlSortNormal, DataOption2:=xlSortNormal
    Range("A3").Select

    Do While blnAtlantaCheck = False
        If ActiveCell.Value = "ATL" Then
            ActiveCell.Offset(1, 0).Select
        Else: blnAtlantaCheck = True
            ActiveCell.Offset(-1, 0).Select
            If ActiveCell.Value = "ATL" Then
                dblLastAtlanta = ActiveCell.Row
                Range("A3:P" & dblLastAtlanta).Delete
            End If
        End If
    Loop

'Recheck the number of entries for later use
    Range("A5").Select
    Selection.End(xlDown).Select
    dblEnd = ActiveCell.Row

'Sort the sheet by SKU number, then by Warehouse
    Range("A3:P" & dblEnd).Select
    Selection.Sort Key1:=Range("B3"), Order1:=xlAscending, Key2:=Range("A3") _
        , Order2:=xlAscending, Header:=xlNo, OrderCustom:=1, MatchCase:=False _
        , Orientation:=xlTopToBottom, DataOption1:=xlSortNormal, DataOption2:= _
        xlSortNormal

'Insert a new column after Class column
    Columns("N:N").Select
    Selection.Insert shift:=xlToRight

'Fills in the formula so that the Class # of the last SKU listing for each SKU is
'  the lowest class number of all Warehouses for that SKU.
'When the macro searches for a SKU later, it will always return the first entry
'  found for that SKU, so this ensures that it will be the lowest class # listed.
'Also, if this class number is greater than 6, it is changed to equal 6.
```

```
        Range("N3").Formula = _
              "=IF(NOT(B3=B4),MIN(M3,6), MIN(M3,N4,6))"
        Range("N3").Select
        Selection.AutoFill Destination:=Range("N3:N" & dblEnd)
        Range("A1").Select
    End If


'Copy corresponding vendor sheet
    Sheets(strVendor).Select
    ActiveSheet.Copy before:=Sheets(1)

'Delete last 22 rows of sheet (locations and descriptors)
    Range("B1").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(-1, 0).Select
    If ActiveCell.Value = "Louisville" Then
        ActiveCell.Offset(-20, -1).Select
        ActiveCell.Range("A1:B22").Delete
    Else: MsgBox ("Key Number sheet not properly formatted. " & _
            "Please check and try again")
        Exit Sub
    End If

'Count number of entries for later use, stores as "intNumSKU"
    Range("A1").Select
    Selection.End(xlDown).Select
    intNumSKU = ActiveCell.Row

'Insert blank row then move up a row
    Do Until ActiveCell.Row = 1
        ActiveCell.EntireRow.Insert shift:=xlDown
        ActiveCell.Offset(-1, 0).Select
    Loop

'Enter formulas for first SKU
    Range("C1").Formula = "=1"
    Range("D1").Formula =
        "=IF(ISERROR(VLOOKUP(B1,icswlstflds!B$3:N$50000,13, FALSE)),6," & _
        "VLOOKUP(B1,icswlstflds!B$3:N$50000,13, FALSE))"
    Range("E1").Formula = 1
    Range("F1").Formula = 0
    Range("G1").Formula = "=0.01*A2"
    Range("H1").Formula = "=1.5*A2"
    Range("I1:J1").Formula = 1
    Range("A2").Formula = "=IF(ISERROR(VLOOKUP(B1, icswlstflds!B$3: " & _
```

```
        "F$50000, 5, FALSE)),50, VLOOKUP(B1, icswlstflds!B$3:F$50000, 5, FALSE))"

'Copy formulas down for 2nd SKU (this is different than for 1st SKU so that the cost
'values for any SKUs not found in the icswlstflds sheet will default to the cost of the
'previous entry)
    Range("A1:A2").Select
    Selection.AutoFill Destination:=Range("A1:A4"), Type:=xlFillDefault
    Range("A4").Formula = "=IF(ISERROR(VLOOKUP(B3, icswlstflds!B$3: " & _
        "F$50000, 5, FALSE)),A2,VLOOKUP(B3, icswlstflds!B$3:F$50000, 5, FALSE))"

'Copy formulas down to max # SKUs
    intNumRow = intNumSKU * 2
    Range("A3:A4").Select
    Selection.AutoFill Destination:=Range("A3:A" & intNumRow), Type:=xlFillDefault
    Range("C1:J2").Select
    Selection.AutoFill Destination:=Range("C1:J" & intNumRow), Type:=xlFillDefault

'Copy and "paste values" of all cells so SKU names can be deleted
    Range("A1:J" & intNumRow).Select
    Selection.Copy
    Range("A1").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Application.CutCopyMode = False

'Delete SKUs column to get in final format
    Columns("B:B").Select
    Application.CutCopyMode = False
    Selection.Delete shift:=xlToLeft
    Range("A1").Select

'Move sheet to new workbook, save, add named range SKUD and close
    strPath = ThisWorkbook.Path
    Sheets(1).Move
    Sheets(1).Name = "Sheet1"
    ActiveWorkbook.SaveAs strPath & "\" & strVendor & "\" & "SKU D.xls"
    Set rngNamerange = Range("A1:I" & intNumRow)
    Workbooks("SKU D.xls").Names.Add Name:="SKUD", _
        RefersTo:=rngNamerange, Visible:=True
    Sheets(1).Name = strVendor
    ActiveWorkbook.Save
    ActiveWorkbook.Close

    Sheets(1).Select
End Sub
```

```
Sub DCSCmaster()
'Creates DC Shipping Cost file
Dim intVendor As String
Dim strPath As String

'Save Vendor ID and current path
   intVendor = Range("E5").Value
   strPath = ThisWorkbook.Path

'Copy Warehouse names
   Sheets("DCSC").Select
   Range("A1:A19").Copy

'Check which sheet is needed for that Vendor
   If intVendor > 0 Then
      If intVendor <= 99 Then
         Sheets("DCSC1to99").Select
      Else:
         If intVendor <= 249 Then
            Sheets("DCSC100to249").Select
         Else: Sheets("DCSC250+").Select
         End If
      End If
   End If

'Input formulas for calculations
   Range("Q1").Select
   ActiveSheet.Paste
   Range("P1:P19").Value = intVendor
   Range("R1").Formula =
   "=IF(SUMPRODUCT(($A$2:$A$60000=$P1)*($B$2:$B$60000=$Q1))=0, 0," & _
   "SUMPRODUCT(($A$2:$A$60000=$P1)*($B$2:$B$60000=$Q1)" & _
   "*($J$2:$J$60000))/" & _
   "SUMPRODUCT(($A$2:$A$60000=$P1)*($B$2:$B$60000=$Q1)))"
   Range("R1").Select
   Selection.AutoFill Destination:=Range("R1:R19")

'Cut and paste formulas into new workbook
   Range("P1:R19").Cut
   Range("A1").Select
   Workbooks.Add
   ActiveSheet.Paste
   Application.CutCopyMode = False

'Paste values and delete first 2 columns
```

```
    Range("C1:C19") = Range("C1:C19").Value
    Range("A:B").Select
    Selection.Delete shift:=xlToLeft

'Create named range DCSCost and save workbook in vendor folder
    Range("A1").Select
    ActiveWorkbook.Names.Add Name:="DCSCost", _
       RefersTo:="=$A$1:$A$19", Visible:=True
    ActiveWorkbook.SaveAs strPath & "\" & intVendor _
       & "\" & "DCSC.xls"
    ActiveWorkbook.Close

    Sheets(1).Select

End Sub
```

```
Sub URmaster()
'Creates the Usage Rate file
Dim strVendor As String
Dim intCount As Integer
Dim strCurrent As String
Dim dblEnd As Double
Dim i As Double
Dim j As Double
Dim intNumSKU As Integer
Dim strPath As String
Dim rngNamerange As Range

'Save vendor number, copy corresponding sheet
    strVendor = Range("E5").Value
    Sheets(strVendor).Select
    ActiveSheet.Copy before:=Sheets(1)

'Delete last 22 rows of sheet (locations and descriptors)
    Range("B1").Select
    Selection.End(xlDown).Select
    ActiveCell.Offset(-1, 0).Select
    If ActiveCell.Value = "Louisville" Then
       ActiveCell.Offset(-20, -1).Select
       ActiveCell.Range("A1:B22").Delete
    Else: MsgBox ("Key Number sheet not properly formatted. " & _
          "Please check and try again")
       Exit Sub
    End If
```

```
'Delete 1st column
    Columns("A:A").Select
    Application.CutCopyMode = False
    Selection.Delete shift:=xlToLeft

'Count number of entries for later use, stores as "intCount"
    Range("A1").Select
    Selection.End(xlDown).Select
    intCount = ActiveCell.Row

'Insert and copy each row 20 times
    Do Until ActiveCell.Row = 1
        strCurrent = ActiveCell.Value
        ActiveCell.Range("A1:A20").Select
        Selection.Insert shift:=xlDown
        ActiveCell.Range("A1:A20").Value = strCurrent
        ActiveCell.Offset(-1, 0).Select
    Loop

    'repeat for top row
    strCurrent = ActiveCell.Value
    ActiveCell.Range("A1:A20").Select
    Selection.Insert shift:=xlDown
    ActiveCell.Range("A1:A20").Value = strCurrent

    Range("A1").Select
    Selection.End(xlDown).Select
    dblEnd = ActiveCell.Row

'Enter Formulas for usage rate lookups
    Range("B1").Formula = 1
    Range("B2").Formula =
          "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""BUFF"")*" & _
          "(icswlstflds!$B$3:$B$32000=A2)*(icswlstflds!$J$3:$J$32000))"
    Range("B3").Formula =
          "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""TULS"")*" & _
          "(icswlstflds!$B$3:$B$32000=A3)*(icswlstflds!$J$3:$J$32000))"
    Range("B4").Formula =
          "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""SD"")*" & _
          "(icswlstflds!$B$3:$B$32000=A4)*(icswlstflds!$J$3:$J$32000))"
    Range("B5").Formula =
          "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""CLEV"")*" & _
          "(icswlstflds!$B$3:$B$32000=A5)*(icswlstflds!$J$3:$J$32000))"
    Range("B6").Formula =
          "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""PITT"")*" & _
          "(icswlstflds!$B$3:$B$32000=A6)*(icswlstflds!$J$3:$J$32000))"
```

```
Range("B7").Formula =
      "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""DALL"")*" & _
      "(icswlstflds!$B$3:$B$32000=A7)*(icswlstflds!$J$3:$J$32000))"
Range("B8").Formula =
      "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""KC"")*" & _
      "(icswlstflds!$B$3:$B$32000=A8)*(icswlstflds!$J$3:$J$32000))"
Range("B9").Formula =
      "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""PHIL"")*" & _
      "(icswlstflds!$B$3:$B$32000=A9)*(icswlstflds!$J$3:$J$32000))"
Range("B10").Formula =
      "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""INDY"")*" & _
      "(icswlstflds!$B$3:$B$32000=A10)*(icswlstflds!$J$3:$J$32000))"
Range("B11").Formula =
      "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""CAL"")*" & _
      "(icswlstflds!$B$3:$B$32000=A11)*(icswlstflds!$J$3:$J$32000))"
Range("B12").Formula =
      "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""MILW"")*" & _
      "(icswlstflds!$B$3:$B$32000=A12)*(icswlstflds!$J$3:$J$32000))"
Range("B13").Formula =
      "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""MINN"")*" & _
      "(icswlstflds!$B$3:$B$32000=A13)*(icswlstflds!$J$3:$J$32000))"
Range("B14").Formula =
      "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""MD"")*" & _
      "(icswlstflds!$B$3:$B$32000=A14)*(icswlstflds!$J$3:$J$32000))"
Range("B15").Formula =
      "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""SA"")*" & _
      "(icswlstflds!$B$3:$B$32000=A15)*(icswlstflds!$J$3:$J$32000))"
Range("B16").Formula =
      "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""NORF"")*" & _
      "(icswlstflds!$B$3:$B$32000=A16)*(icswlstflds!$J$3:$J$32000))"
Range("B17").Formula =
      "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""NJ"")*" & _
      "(icswlstflds!$B$3:$B$32000=A17)*(icswlstflds!$J$3:$J$32000))"
Range("B18").Formula =
      "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""ORL"")*" & _
      "(icswlstflds!$B$3:$B$32000=A18)*(icswlstflds!$J$3:$J$32000))"
Range("B19").Formula =
      "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""BKLN"")*" & _
      "(icswlstflds!$B$3:$B$32000=A19)*(icswlstflds!$J$3:$J$32000))"
Range("B20").Formula =
      "=SUMPRODUCT((icswlstflds!$A$3:$A$32000=""PHNX"")*" & _
      "(icswlstflds!$B$3:$B$32000=A20)*(icswlstflds!$J$3:$J$32000))"
Range("B21").Formula =
      "=.1*SUMPRODUCT((icswlstflds!$A$3:$A$32000=""CENT"")*" & _
      "(icswlstflds!$B$3:$B$32000=A21)*(icswlstflds!$J$3:$J$32000))"
```

```
'Copy formulas down to end, one at a time w/ pastevalues after each
    Application.ScreenUpdating = False
    Range("B1:B21").Select
    i = 23
    j = 42
    intNumSKU = 2
    Do Until intNumSKU > intCount
        Range("B2:B21").Copy
        Range("B" & i).Select
        ActiveSheet.Paste
        Range("B" & i & ":B" & j).Copy
        Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
            :=False, Transpose:=False
        Application.CutCopyMode = False
        ActiveCell.Offset(-1, 0).Select
        ActiveCell.Value = intNumSKU
        intNumSKU = intNumSKU + 1
        i = i + 21
        j = j + 21
    Loop
    Range("B2:B21").Select
    Selection.Copy
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Application.CutCopyMode = False
    Application.ScreenUpdating = True

'Delete SKU names column to get in final format
    Columns("A:A").Select
    Selection.Delete shift:=xlToLeft
    Range("A1").Select

'Move sheet to new workbook and save
    strPath = ThisWorkbook.Path
    Sheets(1).Move
    ActiveWorkbook.SaveAs strPath & "\" & strVendor & "\" & "UR.xls"

'Set named range, save, and close
    Sheets(1).Name = "Sheet1"
    Set rngNamerange =
          Range("OFFSET(Sheet1!$A$1,0,0,COUNTA(Sheet1!$A:$A),1)")
    ActiveWorkbook.Names.Add Name:="UseR", _
        RefersTo:=rngNamerange, Visible:=True
    Sheets(1).Name = strVendor
    ActiveWorkbook.Save
    ActiveWorkbook.Close
```

```
    Sheets(1).Select
End Sub
```

```
Sub VPLcopy()
'Copies the Vendor Product Line file into the new Vendor folder
Dim strVendor As String
Dim strPath As String

'Save vendor number, save current path, copy sheet
    strVendor = Range("E5").Value
    strPath = ThisWorkbook.Path
    Sheets("VPL").Copy
    Application.CutCopyMode = False

'Save workbook in corresponding Vendor ID folder and close
    ActiveWorkbook.SaveAs strPath & "\" & strVendor _
        & "\" & "VPL.xls"
    ActiveWindow.Close

    Sheets(1).Select
End Sub
```

```
Sub DCSScopy()
'Copies the DC Shipping Schedule file into the new Vendor folder
Dim strVendor As String
Dim strPath As String

'Save vendor number, save current path, copy sheet
    strVendor = Range("E5").Value
    strPath = ThisWorkbook.Path
    Sheets("DCSS").Copy
    Application.CutCopyMode = False

'Save workbook in corresponding Vendor ID folder and close
    ActiveWorkbook.SaveAs strPath & "\" & strVendor _
        & "\" & "DCSS.xls"
    ActiveWindow.Close

    Sheets(1).Select
End Sub
```

```
Sub VSCcopy()
'Copies the Vendor Shipping Cost file into the new Vendor folder
Dim strVendor As String
Dim strPath As String

'Save vendor number, save current path, copy sheet
    strVendor = Range("E5").Value
    strPath = ThisWorkbook.Path
    Sheets("VSC").Copy
    Application.CutCopyMode = False

'Save workbook in corresponding Vendor ID folder and close
    ActiveWorkbook.SaveAs strPath & "\" & strVendor _
        & "\" & "VSC.xls"
    ActiveWindow.Close

    Sheets(1).Select
End Sub
```

```
Sub InvCCcopy()
'Copies the Inventory Carrying Cost file into the new Vendor folder
Dim strVendor As String
Dim strPath As String

'Save vendor number, save current path, copy sheet
    strVendor = Range("E5").Value
    strPath = ThisWorkbook.Path
    Sheets("InvCC").Copy
    Application.CutCopyMode = False

'Save workbook in corresponding Vendor ID folder and close
    ActiveWorkbook.SaveAs strPath & "\" & strVendor _
        & "\" & "InvCC.xls"
    ActiveWindow.Close

    Sheets(1).Select
End Sub
```

```
Sub CDmaster()

Dim i As Integer
Dim strVendor As String
Dim blnFirstFound As Boolean
Dim intFirst As Integer
```

```
Dim blnLastFound As Boolean
Dim intLast As Integer
Dim strPath As String
Dim rngNameRange As Range
Dim dblEnd As Double
Dim j As Integer
Dim k As Integer
Dim intDay As Integer
Dim intLastDay As Integer
Dim dblDayCell As Double

'Save vendor number, add a new sheet at the end for this vendor
   Sheets(1).Select
   strVendor = Range("B5").Value
   Sheets.Add
   ActiveSheet.Name = "CD"
   Sheets(strVendor).Copy after:=Sheets(1)
   ActiveSheet.Name = "KNIF"


'Search for demand entries for desired vendor
   For i = 4 To 13
      Sheets(i).Select
      Range("B1").Select
      Application.ScreenUpdating = False
      Do Until blnFirstFound = True
         If ActiveCell.Value = strVendor Then
            intFirst = ActiveCell.Row
            blnFirstFound = True
         End If
         ActiveCell.Offset(1, 0).Select
      Loop
      Do Until blnLastFound = True
         If ActiveCell.Value <> strVendor Then
            intLast = ActiveCell.Row - 1
            blnLastFound = True
         End If
         ActiveCell.Offset(1, 0).Select
      Loop
      Application.ScreenUpdating = True
      Range("A" & intFirst & ":G" & intLast).Select
      Selection.Copy
      Sheets(1).Select
      ActiveSheet.Paste
      Range("A1").Select
      Selection.End(xlDown).Select
```

```
        ActiveCell.Offset(1, 0).Select

        blnFirstFound = False
        blnLastFound = False
    Next i

'Begin creating the format for the CD file
strPath = ThisWorkbook.Path

'Delete 2 price columns and Vendor ID column
    Range("B:B, F:F, G:G").Select
    Selection.Delete shift:=xlToLeft

'Find last entry
    Range("A1").Select
    Selection.End(xlDown).Select
    dblEnd = ActiveCell.Row

'Copy table for warehouse keys
    Sheets(3).Select
    Range("A51:B70").Copy
    Sheets(1).Select
    Range("G1").Select
    ActiveSheet.Paste

'Insert column for warehouse keys, then sort by warehouse key
    Columns("B:B").Select
    Selection.Insert shift:=xlToRight
    Range("B1").Formula = "=INDEX(H$1:I$20, MATCH(A1, I$1:I$20,0),1)"
    Range("B1").Select
    Selection.AutoFill Destination:=Range("B1:B" & dblEnd)
    Range("A1:E" & dblEnd).Sort Key1:=Range("B1"), Order1:=xlAscending, Header:= _
        xlNo, OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, _
        DataOption1:=xlSortNormal

'Delete rows which have errors for WHSE (usually from Atlanta)
    Range("B1").Select
    j = 0
    Selection.End(xlDown).Select
    Do Until j = 1
        If IsError(ActiveCell.Value) = True Then
            ActiveCell.Offset(-1, 0).Select
        Else: j = 1
            ActiveCell.Offset(1, 0).Select
        End If
    Loop
```

```
    Range("A" & ActiveCell.Row & ":E" & dblEnd).Select
    Selection.Delete shift:=xlUp
    Range("A1").Select
    Selection.End(xlDown).Select
    dblEnd = ActiveCell.Row

'Create column for date keys (where 11/1/04 =1, 11/2/04 =2,...,10/31/05 =365)
    Range("F1").Formula = "=E1 - 38291"
    Range("F1").Select
    Selection.NumberFormat = "General"
    Selection.AutoFill Destination:=Range("F1:F" & dblEnd)

'Add column for SKU keys
    Columns("D:D").Select
    Selection.Insert shift:=xlToRight
    Range("D1").Formula = "=INDEX(KNIF!A$1:B$250, MATCH(C1, &" _
       "KNIF!B$1:B$250,0),1)"
    Range("D1").Select
    Selection.AutoFill Destination:=Range("D1:D" & dblEnd)
    Columns("D:D").Select
    Range("A1:G" & dblEnd).Sort Key1:=Range("D1"), Order1:=xlAscending,Header:= _
       xlNo, OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, _
       DataOption1:=xlSortNormal

'Delete rows which have errors for the SKU key (i.e. not found in KNIF)
    Range("D1").Select
    Application.ScreenUpdating = False
    k = 0
    Selection.End(xlDown).Select
    Do Until k = 1
       If IsError(ActiveCell.Value) = True Then
          ActiveCell.Offset(-1, 0).Select
       Else: k = 1
          ActiveCell.Offset(1, 0).Select
       End If
    Loop
    Application.ScreenUpdating = True

    Range("A" & ActiveCell.Row & ":G" & dblEnd).Select
    Selection.Delete shift:=xlUp
    Range("D1").Select
    Selection.End(xlDown).Select
    dblEnd = ActiveCell.Row

'Copy and paste values for whole sheet and delete unnecessary columns
    Range("A1:G" & dblEnd) = Range("A1:G" & dblEnd).Value
```

```
    Range("A:A,C:C,F:F, I:I, J:J").Select
    Selection.Delete shift:=xlToLeft

'Rearrange columns and sort by Date/Whse/SKU
    Columns("D:D").Select
    Selection.Insert shift:=xlToRight
    Columns("A:A").Select
    Selection.Cut Destination:=Columns("D:D")
    Columns("A:A").Select
    Selection.Delete shift:=xlToLeft

    Columns("A:D").Select
    Selection.Sort Key1:=Range("D1"), Order1:=xlAscending, Key2:=Range("C1") _
        , Order2:=xlAscending, Key3:=Range("A1"), Order3:=xlAscending, Header:= _
        xlNo, OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, _
        DataOption1:=xlSortNormal, DataOption2:=xlSortNormal, DataOption3:= _
        xlSortNormal

'Add formulas for totaling orders by branch for each day
    Range("D1").Select
    Selection.End(xlDown).Select
    intLastDay = ActiveCell.Value
    intDay = 1
    Application.ScreenUpdating = False
    Range("D1").Select
    Do While intDay <= 365

        If ActiveCell.Value >= intDay Then
            'If the current entry is the first entry for the next day, insert 21 rows
            ActiveCell.Offset(0, -3).Range("A1:D21").Select
            Selection.Insert shift:=xlDown

            ActiveCell.Formula = intDay
            dblDayCell = ActiveCell.Row

            'On the first newly inserted row, begin entering formulas
            Range("B" & dblDayCell + 1).Formula = "1"
            Range("B" & dblDayCell + 2).Formula = "2"
            ActiveCell.Range("B2:B3").Select
            Selection.AutoFill Destination:=ActiveCell.Range("A1:A20"), _
                Type:= xlFillDefault
            Range("A" & dblDayCell + 1).FormulaR1C1 =
              "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-1]C)*(RC[2]:R[300]C[2]=RC[1]))"
            Range("A" & dblDayCell + 2).FormulaR1C1 =
              "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-2]C)*RC[2]:R[300]C[2]=RC[1]))"
            Range("A" & dblDayCell + 3).FormulaR1C1 =
```

```
        "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-3]C)*RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 4).FormulaR1C1 =
        "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-4]C)*RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 5).FormulaR1C1 =
        "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-5]C)*RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 6).FormulaR1C1 =
        "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-6]C)*RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 7).FormulaR1C1 =
        "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-7]C)*RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 8).FormulaR1C1 =
        "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-8]C)*RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 9).FormulaR1C1 =
        "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-9]C)*RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 10).FormulaR1C1 =
        "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-10]C)*RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 11).FormulaR1C1 =
        "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-11]C)*RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 12).FormulaR1C1 =
       "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-12]C)*(RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 13).FormulaR1C1 =
       "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-13]C)*(RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 14).FormulaR1C1 =
       "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-14]C)*RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 15).FormulaR1C1 =
       "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-15]C)*RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 16).FormulaR1C1 =
       "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-16]C)*(RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 17).FormulaR1C1 =
       "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-17]C)*RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 18).FormulaR1C1 =
       "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-18]C)*RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 19).FormulaR1C1 =
       "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-19]C)*(RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 20).FormulaR1C1 =
       "=SUMPRODUCT((RC[3]:R[300]C[3]=R[-20]C)*RC[2]:R[300]C[2]=RC[1]))"
      Range("A" & dblDayCell + 1 & ":A" & dblDayCell + 20) = _
         Range("A" & dblDayCell + 1 & ":A" & dblDayCell + 20).Value
      ActiveCell.Offset(20, 2).Select
      intDay = intDay + 1
   Else: ActiveCell.Offset(1, 0).Select
   End If
Loop
If intDay < intLastDay Then
   MsgBox ("Error inserting rows. Check File")
End If
```

```
    Application.ScreenUpdating = True
    Range("A1").Select
    Selection.End(xlDown).Select
    dblEnd = ActiveCell.Row

'Delete columns C and D
    Range("C:C, D:D").Select
    Selection.Delete shift:=xlToLeft

'Make named range and save workbook
    Sheets(1).Move
    Set rngNameRange = Range("A1:B" & dblEnd)
    Sheets(1).Names.Add Name:="CustomerDemand", RefersTo:=rngNameRange, _
        Visible:=True
    Range("A1").Select
    ActiveWorkbook.SaveAs strPath & "\" & strVendor _
        & "\" & "CD.xls"
    ActiveWorkbook.Close

    Application.DisplayAlerts = False
    Sheets(1).Delete
    Application.DisplayAlerts = True
End Sub
```

Appendix B – Simulation Run Results

Vendor 10				
Product Line CG				
1	1	0	0	0
2	1	0	1	1
3	1	0	1	1
4	1	0	1	1
5	1	0	0	0
6	1	0	1	1
7	1	0	1	0
8	1	0	1	1
9	1	0	1	0
10	1	0	1	1
11	1	0	1	1
12	1	0	1	1
13	1	0	0	0
14	1	0	1	1
15	1	0	0	0
16	1	0	1	1
17	1	0	1	1
18	1	0	1	1
19	1	0	1	1
20	0	0	0	0
Beginning Inv	363660	287760	321060	318930
End Inv	337480	261690	297620	296840
InvCC	71406	53985	61555	59886
Lost Sales	566.9	1063.9	501.86	495.84
# Insufficient shipments	377	0	80	39
Ordering Cost	2438	1065	1600	1518
Purchased Parts cost	448960	400620	458240	459820
Sales	701560	627000	711320	712220
Shipping cost	36404	20031	25955	25139
Avg Inv @ 10	7558.2	7467.9	7558.2	7558.2
Max Inv @ 10	7574.5	7503.6	7574.5	7574.5
Avg Inv @ 20	187170	7883.2	51931	38440
Max Inv @ 20	220110	10424	61325	43537
Net Profit	$ 115,590	$ 124,150	$ 140,020	$ 143,270

Vendor 11				
Product Line PA				
1	1	0	1	1
2	1	0	1	1
3	1	0	0	0
4	1	0	1	1
5	1	0	1	0
6	1	0	1	0
7	1	0	1	0
8	1	0	1	0
9	1	0	1	1
10	1	0	1	0
11	1	0	1	1
12	1	0	1	1
13	1	0	1	1
14	1	0	0	0
15	1	0	1	1
16	1	0	1	1
17	1	0	1	0
18	1	0	1	0
19	1	0	1	1
20	0	0	0	0
Beginning Inv	92054	100830	89885	86798
End Inv	109440	105380	97162	111070
InvCC	22656	21210	18623	20020
Lost Sales	230.62	359.42	192.33	187.8
# Insufficient shipments	11	0	4	4
Ordering Cost	939	464	862	808
Purchased Parts cost	204710	183240	201550	213410
Sales	289640	270320	295380	296060
Shipping cost	16839	9162.1	14119	12811
Avg Inv @ 10	1111	1111	1111	1111
Max Inv @ 10	1111	1111	1111	1111
Avg Inv @ 20	77572	11363	35966	24201
Max Inv @ 20	97600	17345	47901	33297
Net Profit	$ 61,642	$ 60,426	$ 67,313	$ 73,098

Vendor 18				
Product Line				
1	1	0	1	1
2	1	0	1	0
3	1	0	1	0
4	1	0	1	1
5	1	0	1	0
6	1	0	1	0
7	1	0	1	1
8	1	0	1	1
9	1	0	1	0
10	1	0	0	0
11	1	0	1	1
12	1	0	0	0
13	1	0	1	0
14	1	0	1	0
15	1	0	1	0
16	1	0	1	0
17	1	0	1	0
18	1	0	1	0
19	1	0	1	0
20	0	0	0	0
Beginning Inv	1031900	1007900	1032600	1013400
End Inv	1050100	1010100	1047700	1045500
InvCC	197310	191930	199740	197840
Lost Sales	612.58	347.82	284.41	319.81
# Insufficient shipments	222	0	173	33
Ordering Cost	2058	966	1320	1161
Purchased Parts cost	382030	402110	416990	424200
Sales	598740	641100	651250	645590
Shipping cost	33365	20105	21421	21288
Avg Inv @ 10	47856	64174	64174	64174
Max Inv @ 10	52124	71179	71179	71179
Avg Inv @ 20	63380	9555.8	49459	39821
Max Inv @ 20	84072	14083	53334	50702
Net Profit	$ 1,624	$ 27,895	$ 26,592	$ 32,885

Vendor 22				
Product Line QV				
1	1	0	1	1
2	1	0	1	1
3	1	0	1	1
4	1	0	1	1
5	1	0	1	1
6	1	0	1	1
7	1	0	1	1
8	1	0	1	1
9	1	0	1	0
10	1	0	1	0
11	1	0	1	1
12	1	0	1	1
13	1	0	1	0
14	1	0	1	1
15	1	0	1	1
16	1	0	0	1
17	1	0	1	0
18	1	0	1	1
19	1	0	1	0
20	0	0	0	0
Beginning Inv	128270	141500	141160	140830
End Inv	150160	129970	162200	166110
InvCC	29703	25772	31516	29711
Lost Sales	583.58	994.9	586.73	583.47
# Insufficient shipments	2	0	4	6
Ordering Cost	1216	673	1202	1120
Purchased Parts cost	442610	381440	441880	445100
Sales	645400	583700	644930	645420
Shipping cost	37361	19072	35095	32025
Avg Inv @ 10	6939.5	13524	6905.8	13524
Max Inv @ 10	9441.6	20069	9441.6	20069
Avg Inv @ 20	81478	15011	78180	60150
Max Inv @ 20	121500	22108	109890	90221
Net Profit	$ 155,810	$ 144,220	$ 155,680	$ 162,160

Vendor 48				
Product Line LA				
1	1	0	1	1
2	1	0	1	1
3	1	0	1	1
4	1	0	1	1
5	1	0	0	0
6	1	0	1	0
7	1	0	1	0
8	1	0	0	0
9	1	0	1	1
10	1	0	1	0
11	1	0	1	1
12	1	0	1	1
13	1	0	0	1
14	1	0	1	0
15	1	0	1	1
16	1	0	1	0
17	1	0	1	1
18	1	0	1	0
19	1	0	1	1
20	0	0	0	0
Beginning Inv	451850	133390	403890	267620
End Inv	504720	159490	473540	345390
InvCC	94351	28644	85776	59712
Lost Sales	140.7	931.56	210.01	118.44
# Insufficient shipments	8	0	0	6
Ordering Cost	285	166	231	351
Purchased Parts cost	158150	61148	161740	175930
Sales	198250	71715	187160	201820
Shipping cost	9756.1	3057.4	8646.8	9765
Avg Inv @ 10	7404.3	7404.3	7404.3	7404.3
Max Inv @ 10	7404.3	7404.3	7404.3	7404.3
Avg Inv @ 20	383390	215.85	333870	194390
Max Inv @ 20	423010	423.18	361970	216400
Net Profit	$ (11,560)	$ 3,866	$ 210	$ 33,703

Vendor 71				
Product Line TE				
1	1	0	1	1
2	1	0	1	0
3	1	0	1	0
4	1	0	1	0
5	1	0	1	1
6	1	0	1	0
7	1	0	1	1
8	1	0	1	0
9	1	0	1	0
10	1	0	1	0
11	1	0	1	0
12	1	0	1	0
13	1	0	1	0
14	1	0	1	0
15	1	0	1	0
16	1	0	0	0
17	1	0	1	0
18	1	0	0	0
19	1	0	1	1
20	0	0	0	0
Beginning Inv	107820	108390	109370	92290
End Inv	137330	118280	128700	112870
InvCC	24417	21876	22990	19512
Lost Sales	188.3	161.19	169.82	158.76
# Insufficient shipments	5	0	8	1
Ordering Cost	1618	1175	1573	1339
Purchased Parts cost	306510	298640	302770	305540
Sales	434000	438070	436780	438430
Shipping cost	27225	14932	24921	17292
Avg Inv @ 10	6945.5	17742	6945.5	17742
Max Inv @ 10	9226.5	21980	9226.5	21980
Avg Inv @ 20	69467	23783	59594	16821
Max Inv @ 20	94342	27295	80901	21959
Net Profit	$ 103,550	$ 111,180	$ 103,690	$ 115,170

Vendor 82				
Product Line QF				
1	1	0	1	0
2	1	0	0	1
3	1	0	1	1
4	1	0	1	0
5	1	0	1	1
6	1	0	1	1
7	1	0	1	0
8	1	0	0	0
9	1	0	1	1
10	1	0	0	0
11	1	0	1	0
12	1	0	1	1
13	1	0	1	0
14	1	0	1	1
15	1	0	1	0
16	1	0	1	1
17	1	0	1	0
18	1	0	1	0
19	1	0	0	0
20	0	0	0	0
Beginning Inv	1011100	843680	1032600	902850
End Inv	1038300	814840	1036300	939720
InvCC	209520	161870	207900	180410
Lost Sales	3108.1	3605.3	2832.8	2043.6
# Insufficient shipments	84	0	78	52
Ordering Cost	4982	3088	4647	4265
Purchased Parts cost	2073200	2007000	2098000	2207900
Sales	3113900	3039300	3155200	3273600
Shipping cost	177140	100350	150750	123840
Avg Inv @ 10	46416	146340	146340	146340
Max Inv @ 10	58868	168720	168720	168720
Avg Inv @ 20	742020	56430	564410	189690
Max Inv @ 20	841770	68513	653770	230460
Net Profit	$ 673,140	$ 734,580	$ 694,730	$ 792,050

Vendor 88				
Product Line PB				
1	1	0	1	1
2	1	0	1	1
3	1	0	1	1
4	1	0	1	0
5	1	0	1	1
6	1	0	1	0
7	1	0	1	0
8	1	0	1	0
9	1	0	1	0
10	1	0	1	0
11	1	0	1	1
12	1	0	1	1
13	1	0	1	1
14	1	0	1	0
15	1	0	1	1
16	1	0	1	0
17	1	0	1	0
18	1	0	1	1
19	1	0	1	1
20	0	0	0	0
Beginning Inv	674880	649260	674880	632210
End Inv	695770	635800	695770	685350
InvCC	129960	119950	129960	126750
Lost Sales	264.43	588.03	264.43	350.45
# Insufficient shipments	63	0	63	71
Ordering Cost	1911	741	1911	1707
Purchased Parts cost	389500	334230	389500	403560
Sales	563340	514800	563340	550440
Shipping cost	32032	16711	32032	29588
Avg Inv @ 10	27750	31357	27750	31357
Max Inv @ 10	28357	34094	28357	34094
Avg Inv @ 20	175760	43618	175760	138770
Max Inv @ 20	201960	48523	201960	167430
Net Profit	$ 30,558	$ 29,119	$ 30,558	$ 41,624

Vendor 125				
Product Line PJ				
1	1	0	0	1
2	1	0	1	0
3	1	0	1	1
4	1	0	0	0
5	1	0	1	1
6	1	0	0	0
7	1	0	1	0
8	1	0	0	1
9	1	0	1	0
10	1	0	0	0
11	1	0	0	0
12	1	0	0	0
13	1	0	0	0
14	1	0	0	1
15	1	0	0	1
16	1	0	0	0
17	1	0	0	0
18	1	0	0	0
19	1	0	1	0
20	0	0	0	0
Beginning Inv	112120	84416	84405	82934
End Inv	107060	85036	85300	85259
InvCC	23387	18481	18491	17918
Lost Sales	460.02	428.55	423.81	396.06
# Insufficient shipments	3	0	0	21
Ordering Cost	1365	1117	1115	1171
Purchased Parts cost	333480	343050	343710	347150
Sales	509230	513950	514660	518830
Shipping cost	29679	17152	17185	18594
Avg Inv @ 10	4960.9	10587	10587	10587
Max Inv @ 10	6638.7	15462	15462	15462
Avg Inv @ 20	64865	2851.7	2903	9020
Max Inv @ 20	86983	4168.6	4513.2	12613
Net Profit	$ 115,810	$ 134,340	$ 134,640	$ 135,920

Vendor 130				
Product Line BM				
1	1	0	1	1
2	1	0	0	0
3	1	0	0	0
4	1	0	1	0
5	1	0	0	0
6	1	0	0	0
7	1	0	1	0
8	1	0	0	0
9	1	0	1	0
10	1	0	0	0
11	1	0	1	0
12	1	0	1	0
13	1	0	0	0
14	1	0	0	0
15	1	0	0	0
16	1	0	0	0
17	1	0	0	0
18	1	0	0	1
19	1	0	0	0
20	0	0	0	0
Beginning Inv	1187100	1023400	1079400	1004400
End Inv	1076800	962070	1018000	973200
InvCC	219830	180620	190460	180340
Lost Sales	1926.9	1115.9	1082.8	1198.4
# Insufficient shipments	360	0	56	80
Ordering Cost	7619	4828	5523	5215
Purchased Parts cost	1950300	2085000	2087900	2090100
Sales	3067100	3188800	3193700	3176400
Shipping cost	180920	104250	113310	114940
Avg Inv @ 10	51673	118330	118330	118330
Max Inv @ 10	65344	139520	139520	139520
Avg Inv @ 20	603060	30390	141000	69847
Max Inv @ 20	699020	34829	162450	87832
Net Profit	$ 596,130	$ 751,640	$ 734,090	$ 753,350

Vendor 157				
Product Line GR				
1	1	0	1	0
2	1	0	1	1
3	1	0	1	1
4	1	0	1	0
5	1	0	1	0
6	1	0	1	1
7	1	0	1	0
8	1	0	1	1
9	1	0	1	0
10	1	0	0	1
11	1	0	1	0
12	1	0	0	0
13	1	0	1	1
14	1	0	1	0
15	1	0	1	1
16	1	0	0	1
17	1	0	1	0
18	1	0	0	0
19	1	0	1	1
20	0	0	0	0
Beginning Inv	313580	219970	316240	276760
End Inv	351530	225000	305420	297110
InvCC	65509	41688	60832	54559
Lost Sales	237.61	1244.1	358.96	376.94
# Insufficient shipments	60	0	39	20
Ordering Cost	3627	1407	3107	2628
Purchased Parts cost	693060	574330	651650	668060
Sales	1007400	856430	989210	986510
Shipping cost	58599	28716	46908	43529
Avg Inv @ 10	14722	48391	48391	14669
Max Inv @ 10	19005	58881	58881	19141
Avg Inv @ 20	237990	14962	158100	101370
Max Inv @ 20	294580	21529	192870	126990
Net Profit	$ 224,320	$ 214,070	$ 215,530	$ 237,710

Vendor 206				
Product Line QT				
1	1	0	1	1
2	1	0	1	1
3	1	0	1	1
4	1	0	1	0
5	1	0	1	1
6	1	0	1	0
7	1	0	1	0
8	1	0	1	1
9	1	0	1	1
10	1	0	0	0
11	1	0	1	1
12	1	0	1	0
13	1	0	1	1
14	1	0	1	1
15	1	0	1	1
16	1	0	1	0
17	1	0	1	0
18	1	0	1	0
19	1	0	1	1
20	0	0	0	0
Beginning Inv	400980	292980	383930	339530
End Inv	487000	307200	406100	383550
InvCC	97692	58237	84399	73598
Lost Sales	604.87	1621.6	580.33	506.8
# Insufficient shipments	28	0	18	24
Ordering Cost	1234	459	1191	1050
Purchased Parts cost	713820	577470	681540	708450
Sales	1004500	851990	1008200	1019200
Shipping cost	63983	28873	53494	41578
Avg Inv @ 10	29001	104880	104880	104880
Max Inv @ 10	41272	130630	130630	130630
Avg Inv @ 20	394280	4994.9	252420	93874
Max Inv @ 20	467630	12278	305590	111120
Net Profit	$ 213,190	$ 199,540	$ 209,150	$ 238,060

Vendor 217				
Product Line R				
1	1	0	1	1
2	1	0	1	0
3	1	0	1	0
4	1	0	1	1
5	1	0	1	1
6	1	0	1	1
7	1	0	1	0
8	1	0	1	0
9	1	0	1	1
10	1	0	0	0
11	1	0	1	0
12	1	0	1	0
13	1	0	1	0
14	1	0	1	1
15	1	0	1	1
16	1	0	1	1
17	1	0	0	1
18	1	0	0	0
19	1	0	1	0
20	0	0	0	0
Beginning Inv	492370	414050	489160	472300
End Inv	596360	372590	524000	505400
InvCC	109330	77840	100660	96557
Lost Sales	1085.1	2515.5	1218.3	964.53
# Insufficient shipments	316	0	325	160
Ordering Cost	3075	1676	2925	2507
Purchased Parts cost	842570	602930	783970	809270
Sales	1160400	945850	1140400	1178500
Shipping cost	70083	30146	58449	49694
Avg Inv @ 10	11543	46090	46090	46090
Max Inv @ 10	15504	62367	62367	62367
Avg Inv @ 20	441760	9947.5	341770	192070
Max Inv @ 20	502530	13988	381210	215360
Net Profit	$ 238,250	$ 189,290	$ 228,040	$ 252,600

Vendor 220				
Product Line NJ				
1	1	0	1	0
2	1	0	1	1
3	1	0	1	1
4	1	0	1	1
5	1	0	1	1
6	1	0	1	0
7	1	0	1	1
8	1	0	1	1
9	1	0	1	1
10	1	0	1	0
11	1	0	1	1
12	1	0	1	1
13	1	0	1	1
14	1	0	1	1
15	1	0	1	1
16	1	0	1	1
17	1	0	1	1
18	1	0	0	0
19	1	0	1	1
20	0	0	0	0
Beginning Inv	217610	106440	220220	209000
End Inv	234680	95954	217730	214320
InvCC	46626	15809	44583	39853
Lost Sales	489.81	2667.8	487.91	516.27
# Insufficient shipments	29	0	22	19
Ordering Cost	2183	386	2119	1936
Purchased Parts cost	454840	222320	440300	446530
Sales	671570	344860	671850	667600
Shipping cost	43789	11116	37263	31156
Avg Inv @ 10	14554	40645	14556	40645
Max Inv @ 10	19920	53512	19920	53512
Avg Inv @ 20	179250	13991	164690	117410
Max Inv @ 20	202720	22050	196620	136700
Net Profit	$ 140,710	$ 82,072	$ 144,620	$ 152,930

Vendor 227				
Product Line DW				
1	1	0	1	1
2	1	0	1	0
3	1	0	1	0
4	1	0	1	0
5	1	0	1	0
6	1	0	1	0
7	1	0	1	1
8	1	0	1	0
9	1	0	1	1
10	1	0	1	1
11	1	0	1	0
12	1	0	1	1
13	1	0	1	1
14	1	0	1	0
15	1	0	1	0
16	1	0	1	0
17	1	0	1	0
18	1	0	1	1
19	1	0	1	1
20	0	0	0	0
Beginning Inv	1160800	1079700	1160800	1054000
End Inv	1227900	1076100	1227900	1107400
InvCC	232540	208380	232540	213370
Lost Sales	1315.8	1186.5	1315.8	1237.9
# Insufficient shipments	65	0	65	55
Ordering Cost	1187	766	1187	1042
Purchased Parts cost	675750	658020	675750	679100
Sales	971260	990660	971260	982940
Shipping cost	59658	32901	59658	46725
Avg Inv @ 10	54464	72056	54464	53966
Max Inv @ 10	63143	85355	63143	62574
Avg Inv @ 20	221240	51116	221240	92429
Max Inv @ 20	260500	54263	260500	112240
Net Profit	$ 67,887	$ 85,768	$ 67,887	$ 94,890

Vendor 234				
Product Line PV				
1	1	0	0	1
2	1	0	0	0
3	1	0	1	1
4	1	0	1	1
5	1	0	0	0
6	1	0	0	0
7	1	0	0	0
8	1	0	0	0
9	1	0	0	0
10	1	0	0	0
11	1	0	0	0
12	1	0	1	1
13	1	0	0	1
14	1	0	0	0
15	1	0	0	1
16	1	0	0	0
17	1	0	0	0
18	1	0	0	0
19	1	0	1	0
20	0	0	0	0
Beginning Inv	254520	165910	166630	174210
End Inv	259790	167680	167590	196140
InvCC	53999	33094	33138	36282
Lost Sales	682.03	829.91	737.07	668.5
# Insufficient shipments	7	0	11	14
Ordering Cost	1455	1221	1223	1240
Purchased Parts cost	906890	893060	901770	920860
Sales	1360000	1337800	1351700	1362000
Shipping cost	76073	44652	45263	48984
Avg Inv @ 10	29395	48909	48909	48909
Max Inv @ 10	42854	72588	72588	72588
Avg Inv @ 20	158800	1311.3	2768.1	20203
Max Inv @ 20	226670	2459.5	4794.1	38863
Net Profit	$ 326,160	$ 366,730	$ 370,560	$ 375,920

Vendor 235				
Product Line LA				
1	1	0	1	0
2	1	0	1	0
3	1	0	0	0
4	1	0	1	1
5	1	0	1	1
6	1	0	0	1
7	1	0	1	0
8	1	0	1	0
9	1	0	1	1
10	1	0	0	0
11	1	0	1	0
12	1	0	1	1
13	1	0	1	0
14	1	0	1	1
15	1	0	1	1
16	1	0	0	1
17	1	0	0	1
18	1	0	0	0
19	1	0	0	0
20	0	0	0	0
Beginning Inv	786930	503980	710630	561700
End Inv	1220000	612690	957010	841390
InvCC	194120	109350	159910	142150
Lost Sales	974.37	804.74	906.4	921.44
# Insufficient shipments	6	0	14	16
Ordering Cost	1897	1311	1717	1598
Purchased Parts cost	913080	714550	797770	818450
Sales	937590	963040	947790	945530
Shipping cost	54944	35727	46429	44040
Avg Inv @ 10	16622	29685	29685	29685
Max Inv @ 10	21386	37220	37220	37220
Avg Inv @ 20	732820	6047.5	483630	307360
Max Inv @ 20	937320	6047.5	560810	400350
Net Profit	$ 205,630	$ 210,000	$ 187,450	$ 218,070

Vendor 242				
Product Line EB				
1	1	0	1	0
2	1	0	0	0
3	1	0	1	1
4	1	0	1	0
5	1	0	1	0
6	1	0	0	0
7	1	0	0	1
8	1	0	0	0
9	1	0	0	0
10	1	0	0	0
11	1	0	0	0
12	1	0	0	0
13	1	0	1	1
14	1	0	0	0
15	1	0	1	1
16	1	0	1	0
17	1	0	0	1
18	1	0	1	0
19	1	0	1	1
20	0	0	0	0
Beginning Inv	605640	318230	496600	362790
End Inv	628460	327030	526000	391380
InvCC	125480	65577	102800	75743
Lost Sales	466.5	1046.1	391.71	212.97
# Insufficient shipments	15	0	1	20
Ordering Cost	1484	633	844	1093
Purchased Parts cost	492900	427010	505790	523650
Sales	718590	631640	729810	756620
Shipping cost	40953	21350	25621	27153
Avg Inv @ 10	18047	108780	108780	108780
Max Inv @ 10	24830	127470	127470	127470
Avg Inv @ 20	564350	472.06	194890	98714
Max Inv @ 20	621050	3842.5	215940	117790
Net Profit	$ 80,133	$ 124,830	$ 123,750	$ 157,350

Vendor 244				
Product Line ME				
1	1	0	0	0
2	1	0	1	1
3	1	0	0	0
4	1	0	0	1
5	1	0	1	1
6	1	0	0	1
7	1	0	0	0
8	1	0	0	0
9	1	0	0	1
10	1	0	0	0
11	1	0	0	0
12	1	0	0	1
13	1	0	1	0
14	1	0	1	1
15	1	0	0	1
16	1	0	1	0
17	1	0	0	1
18	1	0	1	0
19	1	0	1	0
20	0	0	0	0
Beginning Inv	243040	137620	151770	143230
End Inv	243640	189760	206250	240020
InvCC	48950	32661	35494	40826
Lost Sales	3081.3	3399.5	3312.9	3270.4
# Insufficient shipments	143	0	38	90
Ordering Cost	1617	843	1026	1025
Purchased Parts cost	240040	243790	253880	286980
Sales	361220	313500	326490	332850
Shipping cost	14999	12189	12693	15555
Avg Inv @ 10	3646.4	4584.9	4584.9	4584.9
Max Inv @ 10	5257	7174	7174	7174
Avg Inv @ 20	141570	3934.9	22611	70686
Max Inv @ 20	157850	5187.9	25934	99576
Net Profit	$ 53,125	$ 72,753	$ 74,565	$ 81,990

Vendor 261				
Product Line YM				
1	1	0	1	1
2	1	0	0	0
3	1	0	1	1
4	1	0	1	1
5	1	0	1	1
6	1	0	0	0
7	1	0	0	0
8	1	0	1	0
9	1	0	0	0
10	1	0	1	0
11	1	0	0	0
12	1	0	0	0
13	1	0	1	0
14	1	0	0	0
15	1	0	1	1
16	1	0	1	0
17	1	0	1	0
18	1	0	1	0
19	1	0	0	0
20	0	0	0	0
Beginning Inv	141450	127750	143840	132720
End Inv	133250	118800	137260	126740
InvCC	22382	20548	23445	21579
Lost Sales	338.79	390.26	324.11	327.54
# Insufficient shipments	305	0	74	22
Ordering Cost	1753	873	1250	1055
Purchased Parts cost	222840	218090	225740	226330
Sales	343900	336180	346100	345590
Shipping cost	19121	10904	12914	12088
Avg Inv @ 10	1644.3	4478.9	1675	4478.9
Max Inv @ 10	2175.3	6864	2242.9	6864
Avg Inv @ 20	88228	2151.7	35989	16184
Max Inv @ 20	125590	3782.4	42494	21835
Net Profit	$ 69,268	$ 76,423	$ 75,843	$ 78,226

Vendor 264				
Product Line MS				
1	1	0	0	1
2	1	0	0	1
3	1	0	0	1
4	1	0	0	0
5	1	0	0	0
6	1	0	0	0
7	1	0	0	1
8	1	0	0	1
9	1	0	0	1
10	1	0	0	1
11	1	0	0	1
12	1	0	0	1
13	1	0	0	1
14	1	0	0	0
15	1	0	0	1
16	1	0	0	0
17	1	0	0	0
18	1	0	0	1
19	1	0	0	1
20	0	0	0	0
Beginning Inv	690530	407500	407500	515250
End Inv	708300	347540	347540	642590
InvCC	143620	76744	76744	119230
Lost Sales	769.27	1250.4	1250.4	785.39
# Insufficient shipments	34	0	0	37
Ordering Cost	1120	609	609	1030
Purchased Parts cost	1337700	1254400	1254400	1412200
Sales	2013800	1941700	1941700	2011400
Shipping cost	75258	62721	62721	76029
Avg Inv @ 10	27121	53747	53747	27121
Max Inv @ 10	38898	86780	86780	38898
Avg Inv @ 20	544650	8.5265E-14	8.5265E-14	373500
Max Inv @ 20	625780	8.5265E-14	8.5265E-14	432980
Net Profit	$ 473,140	$ 485,960	$ 485,960	$ 529,500

Vendor 278				
Product Line EM				
1	1	0	0	1
2	1	0	0	0
3	1	0	1	1
4	1	0	0	1
5	1	0	1	1
6	1	0	1	1
7	1	0	0	0
8	1	0	0	1
9	1	0	0	1
10	1	0	0	0
11	1	0	0	0
12	1	0	1	1
13	1	0	1	0
14	1	0	1	1
15	1	0	1	1
16	1	0	0	1
17	1	0	0	1
18	1	0	0	1
19	1	0	1	0
20	0	0	0	0
Beginning Inv	525310	267660	267570	419220
End Inv	497270	254490	254490	405920
InvCC	94007	43904	43897	73647
Lost Sales	954.66	3999.4	3999.9	1612.9
# Insufficient shipments	74	0	685	390
Ordering Cost	1666	246	931	1613
Purchased Parts cost	655850	363200	363200	600330
Sales	1014700	557980	557910	915960
Shipping cost	52574	18160	18160	35637
Avg Inv @ 10	13432	31011	31011	31011
Max Inv @ 10	16569	46867	46867	46867
Avg Inv @ 20	344100	1233.5	1196.7	122800
Max Inv @ 20	420880	3078.5	2968.9	145250
Net Profit	$ 181,620	$ 115,300	$ 114,630	$ 189,820

Vendor 26536				
Product Line G				
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0
4	1	0	0	0
5	1	0	0	0
6	1	0	0	0
7	1	0	0	0
8	1	0	1	1
9	1	0	0	0
10	1	0	0	0
11	1	0	0	0
12	1	0	0	0
13	1	0	1	1
14	1	0	0	0
15	1	0	0	0
16	1	0	0	0
17	1	0	0	0
18	1	0	0	0
19	1	0	0	0
20	0	0	0	0
Beginning Inv	935310	898340	905210	905210
End Inv	913040	869220	875840	875840
InvCC	178690	167100	168350	168350
Lost Sales	708.68	722.73	460.66	460.66
# Insufficient shipments	186	0	16	16
Ordering Cost	8812	6131	6259	6259
Purchased Parts cost	1255100	1252700	1278700	1278700
Sales	1910400	1908200	1947600	1947600
Shipping cost	105800	62637	64577	64577
Avg Inv @ 10	33341	57070	57070	57070
Max Inv @ 10	36106	62866	62866	62866
Avg Inv @ 20	349710	25367	32024	32024
Max Inv @ 20	401530	28223	35659	35659
Net Profit	$ 338,960	$ 389,800	$ 399,800	$ 399,800

Vendor 27511				
Product Line QG				
1	1	0	1	1
2	1	0	1	1
3	1	0	0	1
4	1	0	1	0
5	1	0	1	0
6	1	0	0	0
7	1	0	1	0
8	1	0	1	0
9	1	0	1	1
10	1	0	0	0
11	1	0	1	1
12	1	0	1	1
13	1	0	1	1
14	1	0	1	0
15	1	0	1	1
16	1	0	0	0
17	1	0	0	0
18	1	0	1	1
19	1	0	1	1
20	0	0	0	0
Beginning Inv	2672100	2306100	2576700	2324000
End Inv	1532100	1643800	2563700	1661900
InvCC	451040	391090	462730	394310
Lost Sales	33219	25445	29087	24875
# Insufficient shipments	47	0	19	152
Ordering Cost	1578	725	1478	1366
Purchased Parts cost	1089300	2189800	2228800	2246400
Sales	2922200	4166000	3583300	4257300
Shipping cost	124240	109490	132480	116460
Avg Inv @ 10	64354	240400	240400	240400
Max Inv @ 10	101350	306260	306260	306260
Avg Inv @ 20	1730900	276730	1485900	304590
Max Inv @ 20	2435900	278900	1716400	326100
Net Profit	$ 82,766	$ 787,240	$ 715,670	$ 811,760

Vendor 27692				
Product Line QL				
1	1	0	1	1
2	1	0	1	0
3	1	0	1	1
4	1	0	1	0
5	1	0	1	1
6	1	0	1	0
7	1	0	1	0
8	1	0	1	1
9	1	0	1	0
10	1	0	0	0
11	1	0	1	0
12	1	0	1	0
13	1	0	1	0
14	1	0	1	1
15	1	0	1	1
16	1	0	1	0
17	1	0	1	0
18	1	0	1	0
19	1	0	1	0
20	0	0	0	0
Beginning Inv	308180	219020	306030	194170
End Inv	490570	374120	583360	438060
InvCC	78457	40520	83248	50216
Lost Sales	435.35	333.28	443.15	324.15
# Insufficient shipments	56	0	56	76
Ordering Cost	584	507	576	661
Purchased Parts cost	569110	557590	633610	624660
Sales	665830	681140	664660	682510
Shipping cost	38869	27879	42014	39821
Avg Inv @ 10	2704.6	2583.3	2583.3	2583.3
Max Inv @ 10	3112.6	2715.3	2715.3	2715.3
Avg Inv @ 20	332600	29562	358200	163290
Max Inv @ 20	426470	29562	514290	270980
Net Profit	$ 160,760	$ 209,410	$ 182,100	$ 210,720

Appendix C – Comparison of Fraction Orders Made to DC

ARP

Branch	Vendor ID																									
	10	11	18	22	48	71	82	88	125	130	157	206	217	220	227	234	235	242	244	261	264	278	26536	27511	27692	Total
1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	1	0	0	0	1	1	18
2	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	0	1	0	1	0	0	0	0	1	1	17
3	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	0	1	0	1	0	0	1	18
4	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	0	0	1	1	20
5	0	1	1	1	0	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	0	1	0	1	1	19
6	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	1	0	0	1	15
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	1	1	18
8	1	1	1	1	0	1	0	1	0	0	1	1	1	1	1	0	1	0	0	1	0	0	1	1	1	16
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	1	1	18
10	1	1	0	1	1	1	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	9
11	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	0	0	0	0	0	0	1	1	17
12	1	1	0	1	1	1	1	1	0	1	0	1	1	1	1	1	1	0	0	0	0	1	0	1	1	17
13	0	1	1	1	0	1	1	1	0	0	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	19
14	1	0	1	1	1	1	1	1	0	0	1	1	1	1	1	0	1	0	1	0	0	1	0	1	1	17
15	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	0	1	1	0	1	0	1	0	1	1	18
16	1	1	1	0	1	0	1	1	0	0	0	1	1	1	1	0	0	1	1	1	0	0	0	0	1	14
17	1	1	1	1	1	1	1	1	0	0	1	1	0	1	1	0	0	0	0	1	0	0	0	0	1	14
18	1	1	1	1	1	0	1	1	0	0	0	1	0	0	1	0	0	1	1	1	0	0	0	0	1	14
19	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	0	1	1	0	0	1	0	1	1	19
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	15	17	17	18	16	17	15	19	6	6	15	18	16	18	19	4	12	9	7	11	0	8	2	14	18	**317**

Optimal

Branch	Vendor ID																									
	10	11	18	22	48	71	82	88	125	130	157	206	217	220	227	234	235	242	244	261	264	278	26536	27511	27692	
1	0	1	1	1	1	1	0	1	1	1	0	1	1	0	1	1	0	0	0	1	1	1	0	1	1	17
2	1	1	0	1	1	0	1	1	0	0	1	1	0	1	0	0	0	0	1	0	1	0	0	1	0	12
3	1	0	0	1	1	0	1	1	1	0	1	1	0	1	0	1	0	1	0	1	1	1	0	1	1	16
4	1	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	0	1	0	0	0	12
5	0	0	0	1	0	1	1	1	1	0	0	1	1	1	0	0	1	0	1	1	0	1	0	0	1	13
6	1	0	0	1	0	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	0	1	0	0	0	8
7	0	0	1	1	0	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0	0	0	7
8	1	0	1	1	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	1	1	1	0	1	11
9	0	1	0	0	1	0	1	0	0	0	0	1	1	1	1	0	1	0	1	0	1	1	0	1	0	12
10	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	4
11	1	1	1	1	1	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0	10
12	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	1	1	0	1	0	1	1	0	1	0	14
13	0	1	0	0	1	0	0	1	0	0	1	1	0	1	1	1	0	1	0	0	1	0	1	1	0	12
14	1	0	0	1	0	0	1	0	1	0	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	11
15	0	1	0	1	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	18
16	1	1	0	1	0	0	1	0	0	0	1	0	1	1	0	0	1	0	0	0	0	1	0	0	0	9
17	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	1	1	0	0	1	0	0	0	8
18	1	0	0	1	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0	8
19	1	1	0	0	1	1	0	1	0	0	1	1	0	1	1	0	0	1	0	0	1	0	0	1	0	12
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	13	10	5	14	11	4	8	10	6	2	9	11	9	15	8	6	9	6	9	5	13	13	2	10	6	**214**

ARP - Optimal	2	7	12	4	5	13	7	9	0	4	6	7	7	3	11	-2	3	3	-2	6	-13	-5	0	4	12	103

VITAE

The author began studying at the University of Louisville's Speed School of Engineering in 2002 with a major in Industrial Engineering. Ms. Forney worked for three semesters as a Co-op student at Algood Food Company, a local private-label peanut butter manufacturer. This position provided a wide range of experiences beyond traditional Industrial Engineering, including Electrical Engineering, programming, and network architecture.

After receiving her Bachelor's degree in August 2006, she immediately began graduate school. In addition to graduate coursework, she worked as a Teaching Assistant for the Introduction to Engineering freshman course. The author has always been strongly interested in simulation, so she was excited to choose this distribution system simulation as a project.

Upon completing her Master's of Engineering degree in Industrial Engineering in May of 2007, Ms. Forney will relocate to Tucson, AZ to begin her career as a Manufacturing Engineer for Raytheon Missile Systems.