OPERATIONS RESEARCH CENTER
Working Paper

Using the Monsanto Software

by
John Ruark

OR 325-98 March 1998

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Using the Monsanto software

by
John Ruark

OR 325-98 March 1998

Using the Monsanto Software

John Ruark

Massachusetts Institute of Technology
77 Massachusetts Ave., E40-139, Cambridge, MA 02139.

In 1995 Monsanto teamed with researchers at the MIT Integrated Supply Chain Management consortium to examine the changing nature of the supply chain of Monsanto's Crop Protection business. One of the primary outputs of this study was a software optimization package that was used for strategic what-if analyses by students at MIT and quantitative analysts at Monsanto. This paper is the documentation that was distributed with that software package. For an executive summary of that project, see Graves, S.C., C. Gutierrez, M. Pulwer, H. Sidhu, and G. Weihs, "Optimizing Monsanto's Supply Chain Under Uncertain Demand," *Annual Conference Proceedings—Council of Logistics Management,* Orlando, FL (1996), 501-516.

Table of contents:

STEPS IN EXECUTING A RUN

1. Create the input files
2. Place the input files and the application into the same folder
3. Run the application
4. Copy the output files to disk
5. Copy the output files to a PC
6. Run the Excel macro

1. Create the input files

There are two aspects of creating input files: Getting the correct data, and formatting the files properly. If the files are not formatted properly, the application will not run. If the data are not correct, the application may run but may not return meaningful solutions.

The section "The Input Files" describes the formats, filenames, and organizations of these files, as well as where to obtain the correct data.

2. Place the input files and the application into the same folder

We recommend keeping a separate folder for each run that uses a different data set or demand scenario. Since there is no version management or history log during the run, keeping output files with input files in one folder, with no other files, is the best way to keep track of which inputs go with which outputs.

This step is technically necessary because the application currently uses only the application's folder for file operations.

3. Run the application

Double-click on the application icon to start the execution cycle. The LINDO front-end window will appear, and the application will start reading data files and creating the LP. The application automatically runs the LINDO solver and outputs the results to the output files.

If the LINDO window presents a ":" prompt, then the application has dumped you into the LINDO command shell. At this point, the LP will be in LINDO's memory, as will be the (hopefully) optimal solution. All LINDO commands at this prompt are valid. To end the application, type "QUIT" at the LINDO prompt.

4. Copy the output files to disk

Upon completion of the application, copy the sixteen output files to a DOS diskette. The names and descriptions of these files can be found in the section "The Output Files." Most current Macintoshes have a PC Exchange extension which permits direct access reads of DOS diskettes in the Macintosh drive. If this is not possible, you may need to run Apple File Exchange to transfer the files.

It is possible to look at these files on the Macintosh, to examine preliminary results without converting the output into an Excel workbook. See the descriptions in "The Output Files" to assess which files would provide meaningful immediate data.

5. Copy the output files to a PC

You should be able to copy the files directly onto a hard-drive and have them processed by Excel without any further modification on your part. Excel will know that these files are coming from a Macintosh. We recommend, as above, that you keep the results from separate runs in separate directories on the PC.

6. Run the Excel macro

Prior to starting Excel, make sure the MONSMACR.XLS file is placed in the appropriate \EXCEL\XLSTART directory. This insures that when Excel starts, the necessary macros and the Monsanto menu are present.

After starting Excel, select "Monsanto|Create workbook from data set." The system will prompt you to find the first output file. This allows you to specify the directory from which files are read. The macro will open each of the files, format them, and add them to a new workbook. It then creates a number of pivot tables and graphs. Finally, it prompts you to save the new workbook to disk prior to beginning any extra analysis.

The specific lists, pivot tables, and graphs created are described in "The Monsanto Workbook in Microsoft Excel". The section "How the Marco Works" describes the function of the Excel macros.

THE INPUT FILES

Inputting data to the model involves two activities: insuring the actual numbers are correct and as desired, and insuring the data are properly formatted and in the correct files. This section summarizes the latter of these, the input file structure. There exists another document which details the actual data collection, definition, and description as it relates to Monsanto.

Most input files are to be stored as comma-separated value (".CSV") files. Each file has one header row followed by any number of data rows. It is imperative that there are *no* blank lines in the file, including before the header row and after the final row. The last row should end with a carriage return, but there should be no characters after that final return.

Some input files are dictionary (".DCT") files. These files define the possible values for the specified dictionary, one per row. The file name specifies the dictionary name; for example, FGILOCTN.DCT contains the dictionary FGILOCTN, which specifies final goods inventory locations. Each line in the file specifies a possible value for that dictionary; for example, in FGILOCTN.DCT, the line "Warehouse QRST" implies that "Warehouse QRST" is a possible final goods inventory location.

The following is a list of the input files:

DEMAND.CSV	DMNDCOST.CSV	FGICAP.CSV
FGIDISTC.CSV	FGIEND.CSV	FGIHOLDC.CSV
FGIINIT.CSV	FGIINV.CSV	FGILOCTN.DCT
HEADERS.CSV	LOSTSALE.CSV	PACKAGE.DCT
PACKCAP.CSV	PACKCHNG.CSV	PACKCOST.CSV
PACKRSRC.DCT	PACKUSE.CSV	PRODCAP.CSV
PRODCHNG.CSV	PRODCOST.CSV	PRODRSRC.DCT
PRODUCT.DCT	PRODUSE.CSV	RAWMATL.DCT
RMICAP.CSV	RMIDISTC.CSV	RMIEND.CSV
RMIHOLDC.CSV	RMIINIT.CSV	RMIINV.CSV
RMILOCTN.DCT	RMIPROD.CSV	RMIUSE.CSV
SCENARIO.DCT	SEGMENT.DCT	TIME.DCT
WEIGHTS.CSV	WIPCAP.CSV	WIPDISTC.CSV
WIPEND.CSV	WIPHOLDC.CSV	WIPINIT.CSV
WIPINV.CSV	WIPLOCTN.DCT	WIPUSE.CSV

These files are described below. The format for the description is as follows: First, the file name (minus the extension) is listed. This is followed by a set of headers, separated by a vertical bar. These headers are the actual headers to be used in the header row. Example:

RMIPROD : RAWMATL | TIME | UNITS

This represents the file RMIPROD.CSV, whose header row would be this:

```
RAWMATL,TIME,UNITS
```

The header list is followed by a brief description of the file.

Input file headers

DEMAND : SCENARIO | PRODUCT | PACKAGE | TIME | DEMAND

Demand for each product/package each month for each scenario

DMNDCOST : PRODUCT | PACKAGE | FGILOCTN | COST

Cost per unit of satisfying demand for each product/package at each storage location

FGICAP : FGILOCTN | TIME | CAPACITY

Inventory capacities at each FGI storage location each month

FGIDISTC : PRODUCT | PACKAGE | FGILOCTN | COST

Cost per unit of transferring each product/package to each FGI storage location

FGIEND : PRODUCT | PACKAGE | FGILOCTN | UNITS

Required ending inventories of each product/package at each FGI storage location

FGIHOLDC : PRODUCT | PACKAGE | FGILOCTN | COST

Holding cost per unit for each product/package at each FGI storage location

FGIINIT : PRODUCT | PACKAGE | FGILOCTN | UNITS

Initial inventories of each product/package at each FGI storage location

FGIINV : PRODUCT | PACKAGE | FGILOCTN | USAGE

For each storage location, amount of space required by one unit of each product/package

FGILOCTN dictionary

Defines the final goods (packaged product) storage locations

HEADERS dictionary

Defines the other headers (this file should contain the list of dictionary names, minus the ".DCT" extension)

LOSTSALE : PRODUCT | PACKAGE | TIME | COST

Lost sales cost per unit for each product/package

PACKAGE dictionary

Defines the packaging options

PACKCAP : PACKRSRC | SEGMENT | TIME | CAPACITY

Segments of piecewise production capacity at each packaging resource each month

PACKCHNG : PACKRSRC | PRODUCT | PACKAGE | COST | TIME

Cost and time to changeover to any product/package for each packaging resource

PACKCOST : PACKRSRC | SEGMENT | TIME | COST

Segments of piecewise production cost per unit for each product/package at each packaging resource each month

PACKRSRC dictionary

Defines the packaging resources

PACKUSE : PRODUCT | PACKAGE | PACKRSRC | USAGE

Packaging resource usage for each product/package

PRODCAP : PRODRSRC | SEGMENT | TIME | CAPACITY

Segments of piecewise production capacity at each production resource each month

PRODCHNG : PRODRSRC | PRODUCT | PACKAGE | COST | TIME

Cost and time to changeover to any product/package for each production resource

PRODCOST : PRODRSRC | SEGMENT | TIME | COST

Segments of piecewise production cost per unit for each product/package at each production resource each month

PRODRSRC dictionary

Defines the production resources (blending and flowables)

PRODUCT dictionary

Defines the products (formulations)

PRODUSE : PRODUCT | PRODRSRC | USAGE

Production resource usage for each product/package

RAWMATL dictionary

Defines the raw materials (technicals)

RMICAP : RMILOCTN | TIME | CAPACITY

Inventory capacities at each RMI storage location each month

RMIDISTC : RAWMATL | RMILOCTN | COST

Cost per unit of transferring each technical to each RMI storage location

RMIEND : RAWMATL | RMILOCTN | UNITS

Required ending inventories of each technical at each RMI storage location

RMIHOLDC : RAWMATL | RMILOCTN | COST

Holding cost per unit for each technical at each RMI storage location

RMIINIT : RAWMATL | RMILOCTN | UNITS

Initial inventories of each technical at each RMI storage location

RMIINV : RAWMATL | RMILOCTN | USAGE

For each storage location, amount of space required by one unit of each technical

RMILOCTN dictionary

Defines the raw material (technical) storage locations

RMIPROD : RAWMATL | TIME | UNITS
Production quantities for each technical for each month

RMIUSE : RAWMATL | PRODUCT | RMILOCTN | USAGE
Quantity of each technical required to produce one unit of each finished product

SCENARIO dictionary
Defines the scenario identifiers

SEGMENT dictionary
Defines the convex cost segment identifiers

TIME dictionary
Defines the time periods

WEIGHTS : SCENARIO | WEIGHT
Weight for each scenario

WIPCAP : WIPLOCTN | TIME | CAPACITY
Inventory capacities at each WIP storage location each month

WIPDISTC : PRODUCT | WIPLOCTN | COST
Cost per unit of transferring each product to each WIP storage location

WIPEND : PRODUCT | WIPLOCTN | UNITS
Required ending inventories of each product at each WIP storage location

WIPHOLDC: PRODUCT | WIPLOCTN | COST
Holding cost per unit for each product at each WIP storage location

WIPINIT : PRODUCT | WIPLOCTN | UNITS
Initial inventories of each product at each WIP storage location

WIPINV : PRODUCT | WIPLOCTN | USAGE
For each storage location, amount of space required by one unit of each product

WIPLOCTN dictionary
Defines the WIP storage locations

WIPUSE : PRODUCT | PACKAGE | USAGE
Quantity of each finished product required to produce one unit of each product/package

Default Values

For most files we assume that the value is zero if the row is omitted from the file. There are a few important exceptions:

RMIHOLDC, WIPHOLDC, FGIHOLDC

Omission ⇒ very high cost‡

RMIDISTC, WIPDISTC, FGIDISTC

Omission ⇒ very high cost‡

RMIINV, WIPINV, FGIINV

Omission ⇒ 1

WEIGHTS

Omission ⇒ (1/Number of scenarios)

Other Notes

- We expect that the format of the files RMIINIT.CSV, etc. and RMIEND.CSV, etc. will need to change so that the package column is interpreted literally (e.g., JUGS), and not as package at channel (e.g., JUGS FOR TRADE PARTNERS).
- The files PRODCHNG.CSV and PACKCHNG.CSV are not currently used.
- The files RMIHOLDC.CSV, WIPHOLDC.CSV, FGIHOLDC.CSV currently serve a special additional purpose: they are used to ascertain which locations are technical storage locations, which are finished product storage locations, and which are finished packaged product locations. For this reason, the "*" option must not be used in these files.

‡ These special default values are used to weakly enforce constraints on where certain raw materials or products can be stored. One must keep this in mind when using the "*" option.

THE OUTPUT FILES

This section describes the format, organization, and filenames of the output files generated by the application.

There are three types of output files:

1. *Aggregation reports.* These provide basic summary results of the run.
2. *Decision variable lists.* These detail specific decision variable parameters, such as value and reduced cost.
3. *Constraint lists.* These detail specific constraint parameters, such as slack and shadow prices.

The following is the list of files needed by the Monsanto macro for Microsoft Excel. For more details on these files, see the section "The Monsanto Workbook in Microsoft Excel."

costs.csv	FGIUsag.csv	PackUsag.csv
ProdUsag.csv	RMIUsag.csv	WIPUsag.csv
Dsjklt.csv	Iilt.csv	Isjklt.csv
Isjlt.csv	Pilt.csv	Pjlt.csv
Psjklt.csv	Qrmt.csv	Qrnt.csv
	Usjklt.csv	

Aggregation reports

Whereas the other two types of reports detail only the optimal solution of the LP, these reports manipulate the optimal solution in conjunction with input information such as costs and usages to generate summary reports of the run. There are currently three types of these reports:

1. Cost report
2. Production utilization report
3. Inventory utilization report

The Production utilization report covers all the production resources in one file and all the packaging resources in another. The Inventory utilization report covers RMI, WIP, and FGI locations, in three separate files. All of the reports have a header line, similar to the input files, which describes each column.

SCENARIO | TIME | COSTTYPE | COST

The Cost report summarizes total costs over each scenario, each time period, and each cost type. The scenarios and time periods are defined via the usual dictionary files. The cost types are currently hard-wired into the application (they have been determined by the description of the model formulation, page 6), and are summarized as follows:

Cost type	Included costs
Holding	all RMI, WIP, and FGI holding costs
Production	all production and packaging resources' production costs
Distribution	all storage locations distribution costs
LostSales	all lost sales costs

This file will report the cost for each scenario and time defined in their dictionaries, for each cost type. Furthermore, it reports several summations of costs. This is done by use of the word "Total" in place of a dictionary entry for each of the scenario, time, and cost type fields. So, for instance, following the breakdown of holding costs for a given scenario over all the time periods, the total holding cost is reported:

```
SCENARIO,TIME,COSTTYPE,COST
...
2,Jul-95,Holding,   23.03
2,Aug-95,Holding,   88.05
...
2,Jun-96,Holding,   23.00
2,Total,Holding,   126.34
```

Similarly, after all cost types are reported, their total for each time period is reported:

```
...
2,Jul-95,Total, 1050.00
2,Aug-95,Total,  533.23
...
```

After an entire scenarios results are reported, the total for that scenario is reported, followed by the total over all scenarios:

```
...
1,Total,Total, 12345.34
2,Total,Total, 54321.00
Total,Total,Total, 66666.99
```

This file is imported into Excel by the conversion macro. When the cost pivot table is created, the "Total" elements are hidden from the table; Excel computes the totals itself rather than using those provided. Thus, the totals in the file are primarily for cursory examination upon completion of an execution.

Production utilization report — *ProdUsag.csv*

PRODRSRC | TIME | USAGE | CAPACITY | UTILIZATION

The Production utilization report, and its identical twin, the Packaging utilization report, detail production resource consumption, in terms of the resource's time units, for each time period and for each resource. No totals are computed.

The production resource (prodrsrc) and time values are taken from their respective dictionaries. Production usage is determined from a specific decision variable (in the model, there is one decision variable for each usage number in the reports). Capacity numbers are taken from the associated capacity input file. Utilization is reported simply for completeness; it is, for each row in the report, the usage divided by the capacity, or zero if the capacity is zero.

Here is a sample part of the report:

```
PRODRSRC,TIME,USAGE,CAPACITY,UTILIZATION
...
ProdA,May-96,  20.0000,   40.0000,   0.5000
ProdA,Jun-96, 300.0000,  300.0000,   1.0000
ProdB,Jul-95,   0.0000,    0.0000,   0.0000
...
```

Note that the units for usage and capacity are the same as those used for all capacity and usage input numbers. Currently, these are hours for production resources.

Packaging utilization report — *PackUsag.csv*

PACKRSRC | TIME | USAGE | CAPACITY | UTILIZATION

The Packaging utilization report details the same results as the Production utilization report, as described above, but for packaging resources, instead. Here is part of a sample Packaging utilization report:

```
PACKRSRC,TIME,USAGE,CAPACITY,UTILIZATION
...
Packaging,May-96,  30.0000,   90.0000,   0.3333
Packaging,Jun-96,  10.0000,  100.0000,   0.1000
...
```

As with the other utilization reports, the units for usage and capacity are in terms of the units used in the input files; for the current data sets, these units are hours.

RMI utilization report — *RMIUsag.csv*

RMILOCTN | TIME | USAGE | CAPACITY | UTILIZATION | SHADOW

This report and its brethren, the WIP utilization and FGI utilization reports, constitute the Inventory utilization reports. They are similar to the Production utilization reports.

The inventory location (rmiloctn) and time values are taken from their respective dictionaries. Inventory usage is determined from a specific constraint's slack (in the model, there is one capacity constraint for each usage number in the reports). Capacity numbers are taken from the associated capacity input file. Utilization is reported simply for completeness; it is, for each row in the report, the usage divided by the capacity, or zero if the capacity is zero.

These files also report the dual prices (shadow) for the storage locations for each month. Clearly, these can be non-zero only when utilization is 100% (complementary slackness condition).

Here is a sample:

```
RMILOCTN,TIME,SCENARIO,USAGE,CAPACITY,UTILIZATION,SHADOW
...
Tech Store 1,May-96,   0.0000, 100.0000,  0.0000,   0.0000
Tech Store 1,Jun-96,  10.0000,  10.0000,  1.0000,  12.3456
Tech Store 2,Jul-95,  50.0000, 100.0000,  0.5000,   0.0000
...
```

The units for usage and capacity are the same as in the input files. Here, these units are kGal.

WIP utilization report — *WIPUsag.csv*

WIPLOCTN | TIME | USAGE | CAPACITY | UTILIZATION | SHADOW

The WIP utilization report details the same results as the RMI utilization report, as described above, but for WIP locations, instead. Here is part of a sample WIP utilization report:

```
WIPLOCTN,TIME,SCENARIO,USAGE,CAPACITY,UTILIZATION,SHADOW
...
WIP1,May-96,   30.0000,   90.0000,   0.3333,   0.0000
WIP1,Jun-96,  200.0000,  200.0000,   1.0000,  12.0000
...
```

As with the other utilization reports, the units for usage and capacity are in terms of the units used in the input files; for the current data sets, these units are kGal.

FGI utilization report — *FGIUsag.csv*

FGILOCTN | TIME | USAGE | CAPACITY | UTILIZATION | SHADOW

The FGI utilization report details the same results as the RMI utilization report, as described above, but for FGI locations, instead. Here is part of a sample FGI utilization report:

```
FGILOCTN,TIME,SCENARIO,USAGE,CAPACITY,UTILIZATION,SHADOW
...
ABC Bulk,Jun-96,       100.0000,  200.0000,   0.5000,   0.0000
Public Terms.,Jul-95, 120.0000,  120.0000,   1.0000,   0.5000
...
```

As with the other utilization reports, the units for usage and capacity are in terms of the units used in the input files; for the current data sets, these units are kGal.

Decision variable lists

These 10 files detail the 10 sets of decision variables specified by the model. For each decision variable, the application outputs several values:

- *Value.* This is the optimal value of the variable in the optimal solution.
- *Cost.* This is the total cost associated with the variable in the objective function. It equals (Value * UnitCost)
- *Reduced Cost.* This is the optimal reduced cost of the variable. Following standard LP theory, the reduced cost will be zero if Value is non-zero, while the reduced cost may be zero or positive if Value is at its lower bound, and reduced cost may be zero or negative if Value is at its upper bound, if one exists.
- *Unit Cost.* This is the coefficient of the variable in the objective function. It is the cost per unit of the variable. These values are taken directly from associated cost input files.

In all cases, each file name is the associate decision variable set it details. Each file has a header row which describes each column. In general, a header looks like this:

-VARIABLE SPECIFIC FIELDS- | VALUE | COST | REDCOST | UNITCOST

where the "variable specific fields" depend on the particular set of decision variables. For instance, for decision variable subscripted "ilt", the variable specified fields are RAWMATL | RMILOCTN | TIME. These four data fields are listed in the headers below as ***DATA FIELDS***.

This table summarizes the decision variable lists:

List Description	Output filename
Demand by SKU/FGI	Dsjklt.csv
Ending Technical Inventory	Iilt.csv
Ending Final Inventory by SKU/FGI	Isjklt.csv
Ending Product Inventory	Isjlt.csv
Technical Production Destination	Pilt.csv
Production of Product	Pjlt.csv
Production by SKU/FGI	Psjklt.csv
Production Resource Consumption	Qrmt.csv
Packaging Resource Consumption	Qrnt.csv
Lost Sales by SKU/FGI	Usjklt.csv

Demand by SKU/FGI List **Dsjklt.csv**

SCENARIO | PRODUCT | PACKAGE | FGILOCTN | TIME | ***DATA FIELDS***

This file details the demand by scenario, product, package, FGI location, and time. This file is comparable to the input file DEMAND.CSV. For each scenario, product, package, and time, and demand is specified, and the LP simply assigns that demand over the possible FGI locations for that scenario, product, package, time combination.

The units for these values are volumes, determined by the units in the input file DEMAND.CSV.

Ending Technical Inventory List **Iilt.csv**

RAWMATL | RMILOCTN | TIME | ***DATA FIELDS***

This file details the *ending* inventory of the raw material by raw material, RMI location, and time.

Because the method of reporting ending inventories excludes the starting inventory, the application adds a starting inventory level for each scenario, product, WIP location set. The starting inventory is stored in the VALUE column; all other data fields are set to zero. The starting inventory is delineated with the word "Initial" in the TIME column.

A sample is shown here:

```
RAWMATL,RMILOCTN,TIME,VALUE,COST,REDCOST,UNITCOST
Tech 1,Tech Store 1,Initial, 0.0000,  0.0000, 0.0000, ...
Tech 1,Tech Store 1,Jul-95,  4.4330, 84.0000, 0.0000, ...
Tech 1,Tech Store 1,Aug-95,  0.0000,  0.0000, 4.4300, ...
...
```

The units of these values are in terms of volume. These values would have to be multiplied by a storage usage factor to determine inventory capacity consumption.

Ending Final Inventory by SKU/FGI List **Isjklt.csv**

SCENARIO | PRODUCT | PACKAGE | FGILOCTN | TIME | ***DATA FIELDS***

This file details the *ending* inventory by scenario, product, package, FGI location, and time.

Because the method of reporting ending inventories excludes the starting inventory, the application adds a starting inventory level for each scenario, product, package, FGI location set. The starting inventory is stored in the VALUE column; all other data fields are set to zero. The starting inventory is delineated with the word "Initial" in the TIME column.

A sample is shown here:

```
SCENARIO,PRODUCT,PACKAGE,FGILOCTN,TIME,VALUE,COST,REDCOST...
2,Rainbow,Cans for TP,Trade Part.,Initial, 0.5000, 0.0000, ...
2,Rainbow,Cans for TP,Trade Part.,Jul-95,  1.5000,18.5400, ...
```

```
2,Rainbow,Cans for TP,Trade Part.,Aug-95, 44.0000,22.0000, ...
...
```

The units of these values are in terms of volume. These values would have to be multiplied by a storage usage factor to determine inventory capacity consumption.

Ending Product Inventory List — *Isjlt.csv*

SCENARIO | TIME | PRODUCT | WIPLOCTN | ***DATA FIELDS***

This file details the *ending* inventory of a product prior to packaging, by scenario, time, product, and WIP location. The holding cost at the WIP locations are set artificially high to discourage storing products there. As with the lost sales variables, this is designed mainly to avoid an infeasible solution.

Because the method of reporting ending inventories excludes the starting inventory, the application adds a starting inventory level for each scenario, product, WIP location set. The starting inventory is stored in the VALUE column; all other data fields are set to zero. The starting inventory is delineated with the word "Initial" in the TIME column.

A sample is shown here:

```
SCENARIO,TIME,PRODUCT,WIPLOCTN,VALUE,COST,REDCOST...
2,Initial,Rainbow,WIP1, 0.5000, 0.0000, 0.0000, ...
2,Initial,Cricket,WIP1, 8.0000,12.0000, 0.0000, ...
2,Jul-95,Rainbow,WIP1,  2.0000, 8.0000, 0.0000, ...
2,Aug-95,Cricket,WIP1,  0.0000, 0.0000, 3.5454, ...
...
```

The units of these values are in terms of volume. These values would have to be multiplied by a storage usage factor to determine inventory capacity consumption.

Technical Production Destination List — *Pilt.csv*

RAWMATL | RMILOCTN | TIME | ***DATA FIELDS***

This file details the production of the raw material destined for each RMI location, by raw material, RMI location, and time.

The units of these values are in terms of volume. For a given time and raw material, the total production (summed over the RMI locations) is specified in the input file RMIPROD.CSV. Thus, this list can be used mainly to verify that the correct amount of RMI was produced, and to check that the raw materials are being stored in the correct RMI locations.

Production of Product List — *Pjlt.csv*

PRODUCT | WIPLOCTN | TIME | ***DATA FIELDS***

This file details the production of a product prior to packaging, by time, product, and WIP location. Because the holding costs at the WIP location are set so high, these production values usually represent total flow through a WIP location during a period rather than strictly the flow into a WIP location.

The units for production are volume. These values would have to be multiplied by a production usage factor to convert to resource consumption.

Production by SKU/FGI List — *Psjklt.csv*

SCENARIO | PRODUCT | PACKAGE | FGILOCTN | TIME | ***DATA FIELDS***

This file details the production by scenario, product, package, FGI location, and time.

The units for production are volume. These values would have to be multiplied by a production usage factor to convert to resource consumption.

Production Resource Consumption List — *Qrmt.csv*

PRODRSRC | TIME | SEGMENT | ***DATA FIELDS***

This file details the consumption of production resource capacity, by time, production resource, and segment. It is a more detailed report than the Production utilization report, described above, in that it provides a breakdown by segment of the variable data fields.

The units of the values are in terms of production resource capacity units, as defined in the input file PRODCAP.CSV. The data in PRODCAP.CSV form simple upper bounds for the decision variables in this list.

Packaging Resource Consumption List — *Qrnt.csv*

PACKRSRC | TIME | SEGMENT | ***DATA FIELDS***

This file details the consumption of packaging resource capacity, by time, packaging resource, and segment. It is a more detailed report than the Packaging utilization report, described above, in that it provides a breakdown by segment of the variable data fields.

The units of the values are in terms of packaging resource capacity units, as defined in the input file PACKCAP.CSV. The data in PACKCAP.CSV form simple upper bounds for the decision variables in this list.

Lost Sales by SKU/FGI List *Usjklt.csv*

SCENARIO | PRODUCT | PACKAGE | FGILOCTN | TIME | ***DATA FIELDS***

This file details the lost sales by scenario, product, package, FGI location, and time. Note that currently, unit cost for all lost sales is set at $100,000, to discourage as much as possible lost sales. The implications of such a high cost are that lost sales are incurred only when the problem would otherwise be infeasible.

The units for these values are volumes, determined by the units in the input file DEMAND.CSV.

Constraint lists

These ten files detail ten of the sets of constraints specified by the model. For each constraint, the application outputs several values:

- *Slack.* If the constraint is an inequality, this number indicates the slack or excess for the constraint.
- *Shadow price.* This is the dual price for the constraint. Clearly, if the slack is non-zero, the shadow price will be zero since the constraint is not binding (complementary slackness).
- *RHS.* This is the original right hand side of the constraint.
- *Allowable increase.* This number indicates by how much the RHS could be increased while retaining the optimality of the optimal basis.
- *Allowable decrease.* This number indicates by how much the RHS could be decreased while retaining the optimality of the optimal basis.

Since these files are not imported into Excel in this iteration of the application, the file names are longer Macintosh names. Each file name contains a short description of the constraint set as well as its equation number from the formulation.

Each file has a header row which describes each column. In general, a header looks like this:

-CONSTRAINT FIELDS- | SLACK | SHADOW | RHS | ALLOWUP | ALLOWDOWN

where the "constraint fields" depend on the particular constraint set. For instance, for constraint set 1, the constraint fields are RAWMATL I TIME. The five data fields are listed in the headers below as ***DATA FIELDS***.

This table summarizes the constraint lists:

List Description	Output filename
Technical production assignment	ProdTech Constraint 1
Product production balance	ProdFgdBal Constraint 3
SKU production balance	ProdPackBal Constraint 5
Technical inventory balance	InvTechBal Constraint 6
Technical inventory capacity	InvTechStor Constraint 7
WIP inventory balance	InvFgdBal Constraint 8
WIP inventory capacity	InvFgdsStor Constraint 9
SKU/FGI inventory balance	InvPackBal Constraint 10
SKU/FGI inventory capacity	InvPackStor Constraint 11
Demand is met	DemandMet Constraint 12

Note that some of the constraints (namely, sets 2, 4, and 13a-c) are not listed in the summary here. This is because these constraints have been implemented with simple bounds, and thus they don't readily return shadow prices. As a result, there is also no range information for those constraints.

Technical production assignment — *ProdTech Constraint 1*

RAWMATL | TIME | ***DATA FIELDS***

This constraint is an inventory balance constraint for the RMI locations. Slacks should be zero.

Product production balance — *ProdFgdBal Constraint 3*

PRODRSRC | TIME | ***DATA FIELDS***

This constraint is a production balance constraint. It equates the time required on a production resource with the actual time used on that resource based upon what is produced. Slacks should be zero.

SKU production balance — *ProdTech Constraint 5*

SCENARIO | PACKRSRC | TIME | ***DATA FIELDS***

This constraint is a packaging balance constraint. It equates the time required on a packaging resource with the actual time used on that resource based upon what is packaged. Slacks should be zero.

Technical inventory balance — *InvTechBal Constraint 6*

RAWMATL | RMILOCTN | TIME | ***DATA FIELDS***

This is a standard inventory balance equation for each raw material at each RMI location. Slacks should be zero.

Technical inventory capacity — *InvTechStor Constraint 7*

TIME | RMILOCTN | ***DATA FIELDS***

This is a standard inventory capacity equation for each RMI location. The amount of raw materials stored at the location cannot exceed its capacity. Slack may be positive.

WIP inventory balance — *InvFgdBal Constraint 8*

SCENARIO | PRODUCT | WIPLOCTN | TIME | ***DATA FIELDS***

This is a standard inventory balance equation for each product at each WIP location, by scenario and time. Slacks should be zero.

WIP inventory capacity — *InvFgdsStor Constraint 9*

SCENARIO | TIME | WIPLOCTN | ***DATA FIELDS***

This is a standard inventory capacity equation for each WIP location. Slacks may be positive.

SKU/FGI inventory balance — *InvPackBal Constraint 10*

SCENARIO | PRODUCT | PACKAGE | FGILOCTN | TIME | ***DATA FIELDS***

This is a standard inventory balance constraint for each SKU at each FGI location, by scenario and time. Slacks should be zero.

SKU/FGI inventory capacity — *InvPackStor Constraint 11*

SCENARIO | TIME | FGILOCTN | ***DATA FIELDS***

This is a standard inventory capacity constraint for each FGI location, by scenario and time. Slacks may be positive.

Demand is met — *DemandMet Constraint 12*

SCENARIO | TIME | PRODUCT | PACKAGE | ***DATA FIELDS***

This constraint insures that the allocation of demand to FGI locations matches the demand levels specified in DEMAND.CSV. Slacks should be zero.

THE MONSANTO WORKBOOK IN MICROSOFT EXCEL

After the conversion macro has finished, Excel will be in a new workbook which contains all of the output data from the run, except for the constraint information.

Contained on each sheet in this workbook is either a pivot table which summarizes a decision variable group, or the actual list of decision variables. The decision variables and aggregated reports, their source output files, resulting worksheets, and resulting display formats are summarized in the following table:

Output Report	Output File	Worksheet Name	Worksheet Display
Cost report	costs.csv	Costs Pivot Costs Data†	pivot table autofiltered list
Production utilization report	ProdUsag.csv	Prod Usage Pivot Prod Usage Data	pivot table and graph autofiltered list
Packaging utilization report	PackUsag.csv		
RMI utilization report	RMIUsag.csv	RMI Usage Pivot RMI Usage Data†	pivot table and graph autofiltered list
WIP utilization report	WIPUsag.csv	Inv Usage Data	autofiltered list
FGI utilization report	FGIUsag.csv		
Demand by SKU/FGI	Dsjklt.csv	SKU Pivot SKU Data†	pivot table autofiltered list
Ending Final Inventory by SKU/FGI	Isjklt.csv		
Production by SKU/FGI	Psjklt.csv		
Lost Sales by SKU/FGI	Usjklt.csv		
Ending Tech. Inventory	Iilt.csv	RM Pivot RM Data†	pivot table autofiltered list
Tech. Production Destination	Pilt.csv		
Ending Product Inventory	Isjlt.csv	WIP Data	autofiltered list
Production of Product	Pjlt.csv	Product Data	autofiltered list
Production Resource Consumption	Qrmt.csv	ProdRsrc Data	autofiltered list
Packaging Resource Consumption	Qrnt.csv	PackRsrc Data	autofiltered list

† Sheet is hidden; use "Format | Sheets | Unhide..." to access.

For more information on manipulating pivot tables and autofiltered lists, see the Excel on-line help or documentation. This section assumes a rudimentary knowledge of pivot fields and pivot items.

The autofiltered lists are only nicely formatted and filtered versions of the output files, imported into Excel. The content of those sheets is described under the corresponding output report in the section "The Output Files." The pivot tables and the graphs are described below.

For a description of what the macro does to create this workbook, see the section "How the Macro Works."

Costs report — *Costs Pivot*

This pivot table summarizes the various costs of the system, as described in the objective function of the model and reported in the output file COSTS.CSV. By default, the table shows the cost by time and cost type, over all scenarios, with totals. The table below shows the default pivot fields:

Field Orientation	Field Contents
Page fields	Scenario
Row fields	Time
Column fields	Cost type
Data fields	Value (cost)

The base sheet for this pivot table is "Costs Data," which by default is hidden. The pivot table is saved without its underlying data, so it may be necessary to refresh to pivot table before manipulating it.

Production Usage Report and Graph — *Prod Usage Pivot*

This pivot table summarizes consumption of production and packaging resources, as reported in the output files PRODUSAG.CSV and PACKUSAG.CSV. The default arrangement of the pivot table is to examine the usage levels and utilizations for each time period for a single resource. By default, the graph shows the usage and excess capacity, as well as utilization, for the currently selected production resource. The table below shows the default pivot fields:

Field Orientation	Field Contents
Page fields	Resource
Row fields	Time
Column fields	none
Data fields	Usage

	Capacity Utilization

The base sheet for this pivot table is "Prod Usage Data," which by default is hidden. In general, the graphs of pivot tables are fairly competent at adapting to pivots in the pivot table; however, this may not be true in this case due to the complexity of the graph (it is a combination bar and line graph with overlapping series). The pivot table is saved without its underlying data, so it may be necessary to refresh to pivot table before manipulating it.

RMI Usage Report and Graph — *RMI Usage Pivot*

This pivot table summarizes consumption of RMI location storage space, as reported in the output files RMIUSAG.CSV. The default arrangement of the pivot table is to examine the usage levels and utilizations for each time period for a single RMI location. By default, the graph shows the usage and excess capacity, as well as utilization, for the currently selected RMI location. The table below shows the default pivot fields:

Field Orientation	Field Contents
Page fields	RMI Location
Row fields	Time
Column fields	none
Data fields	Usage Capacity Utilization

The base sheet for this pivot table is "RMI Usage Data," which by default is hidden. In general, the graphs of pivot tables are fairly competent at adapting to pivots in the pivot table; however, this may not be true in this case due to the complexity of the graph (it is a combination bar and line graph with overlapping series). The pivot table is saved without its underlying data, so it may be necessary to refresh to pivot table before manipulating it.

> *Note that a similar pivot table for the other inventory locations can also be created (based upon the sheet "Inv Usage Data"). These data sets also include a scenario, since inventory levels at WIP and FGI locations are scenario dependent. It would be simple to add this other pivot table; contact the authors for an extension if it is desired during Phase 1 analysis.*

SKU/FGI Report — *SKU Pivot*

This pivot table summarizes the four decision variable sets which deal with individual SKUs at FGI locations. These four sets are orders of magnitude larger than other decision variables, so the amount of data summarized here is immense.

This report details the following decision variables:

Decision variable	Description
Dsjklt	Demand
Isjklt	Ending inventory
Psjklt	Production
Usjklt	Lost Sales

These sets of decision variables are referred to as the Types of variables, and they have their own field in the pivot table called "TYPE." This is a result of the format of the database worksheet on which the pivot table is based; for each combination of scenario, product, package, time, and FGI location, there are four values describing the production, demand, lost sales, and ending inventory for that combination. These result in four rows in the worksheet, and they are differentiated with a "TYPE" column added during the run of the macro which creates the workbook. Unhide the worksheet "SKU Data" to see this in action.

The dimensions of these variables, as indicated by the subscripts, are:

Subscript	Dimension
s	Scenario
j	Product
k	Package
l	FGI location
t	Time

It is possible to organize the pivot table to examine any grouping or ordering of these five dimensions. Some examples are described below.

By default, the pivot table will present a matrix which summarizes the total volume of demand, inventory, production, and lost sales for each product, totaled over all scenarios, packages, and FGI locations. In this form, this matrix essentially quantifies the balance constraints for each product in each time period. The table below shows the default pivot fields:

Field Orientation	Field Contents
Page fields	Package FGI Location Scenario
Row fields	Product
Column fields	Time Type
Data fields	Value Cost

The base sheet for this pivot table is "SKU Data," which by default is hidden. The pivot table is saved without its underlying data, so it may be necessary to refresh to pivot table before manipulating it.

Suggestions for the SKU/FGI pivot table

Here are some suggestions for manipulating the placement of pivot fields to examine different perspectives on the data:

- To examine performance for a particular product, make the Product field a page field, and move the FGI Location and Package fields to row fields. Then select a particular product to view. This matrix will show the actual production, demand, inventory, and lost sales values and costs for that product at each FGI location and package over time.
- To examine a set of variables, such as all the production values together, or all lost sales together, switch the order of the column fields Time and Type, so that Type comes before Time.
- To compare two products at a particular FGI location or for a given package, from the default pivot table arrangement, select the FGI location or package you desire and compare.
- To compare different scenarios for a given product, make the Product field a page field and the Scenario field a row field. This is only useful if the run had more than one scenario.

These is currently no way to return the table directly to its default format. You might want to copy the worksheet prior to changing the fields or formatting.

Using SKU Pivot to examine direct vs. indirect shipments

We have included another macro which creates a worksheet which modifies the SKU Pivot to detail the volumes and costs of distribution, based upon package type (Bulk or Packaged) and channel type (Direct or Indirect).

To run this macro, select "Monsanto|Create distribution network pivot." This macro will create the a worksheet called "DistNetwork", setup the pivot table, and give it a default format and layout. Here is the default field setup:

Field Orientation	Field Contents
Page fields	Type Scenario Product
Row fields	Distributor Dist. Method
Column fields	Container
Data fields	Value (kGal) Value (%) Cost ($) Cost (%)

This table indicates that there are some fields which have not been defined before, and are not part of the original model. These fields (Distribution, Dist. Method, and Container) are actually Groupings of pivot items from the original data's own fields (search for "grouping," topic "Grouping selected items in a pivot table field" in Microsoft Excel Help for more information).

Dist. Method The Dist. Method field captures both the package type and the channel of distribution. This field is based upon the original field FGI Location, and maps as follows:

Dist. Method Group	Contains
Bulk Direct	WX Bulk YZ Bulk QRST Bulk
Bulk Indirect	Public Terminals
Package Direct	Trade Partners Warehouse 1
Package Indirect	Public Terminals

These assignments are hard-coded into the macro which creates this pivot table. It would be simple to change the assignments if necessary.

Container The Container field aggregates the Dist. Method field into simply "Package" and "Bulk." Package elements include Package Direct and Package Indirect, while Bulk includes Bulk Direct and Bulk Indirect, as assigned above.

Distributor The distributor field captures whether a package good is going through the Trade Partners or through Independent Distributors. This field is based upon the original field Package. The item "TP Dist." contains the elements "Jugs for TP Dist." and "Shuttle for TP Dist." The item "Ind. Dist." contains the elements "Jugs for Ind. Dist." and "Shuttle for Ind. Dist."

The Data Fields Although there are four data fields shown, the two value fields (kGal and %) are the same data, just formatted differently, as are the two cost fields ($ and %).

This pivot sheet gives an idea of what can be done with these pivot tables. This pivot table in particular contains a wealth of information; for the one scenario, seven product case we have been using, the original data sheet has over 3000 data elements.

This pivot table summarizes the production and inventory of technicals at each RMI storage location for each time period. This report details the following decision variables:

Decision variable	Description
Iilt	Inventory
Pilt	Production

These sets of decision variables are referred to as the Types of variables, and they have their own field in the pivot table called "TYPE." This is a result of the format of the database worksheet on which the pivot table is based; for each combination of RMI location, raw material, and time period, there are two values describing the production and ending inventory for that combination. These result in two rows in the worksheet, and they are differentiated with a "TYPE" column added during the run of the macro which creates the workbook. Unhide the worksheet "RM Data" to see this in action.

The dimensions of these variables, as indicated by the subscripts, are:

Subscript	Dimension
i	Raw material
l	RMI location
t	Time

It is possible to organize the pivot table to examine any grouping or ordering of these three dimensions.

The default arrangement of the table is to present the production and inventory levels for each technical in each time period, summed over all RMI locations. The table below shows the default pivot fields:

Field Orientation	Field Contents
Page fields	RMI Location
Row fields	Raw material Production/Inventory
Column fields	Time Type
Data fields	Value Cost

The base sheet for this pivot table is "RM Data," which by default is hidden. The pivot table is saved without its underlying data, so it may be necessary to refresh to pivot table before manipulating it.

Subscripts

Subscript convention is retained in all variables, constants, and indices. The subscripts are defined as follows:

Subscript	Dimension	Range	Example
i	Technical (RM)	i = 1, 2	Technical 1
j	Formulation	j = 1…7	Rainbow
k	Package	k = 1…5	Cans at Ind. Dist.
jk	SKU for channel		Rainbow cans at Ind. Dist.
l	Storage	l = 1…L	Bulk at plant
m	Prod Resource	m = 1, 2	Production 1
n	Pack Resource	n = 1	Packaging
r	Segment	r = 1	
s	Scenario	s = 1…3	
t	Time	t = 1…12	Jan-96

The following notation is used for all variables: Lower case variables denote decision variables, while upper case variables denote problem parameters. The class of exceptions is the set of decision variables for inventories, which use a capital I.

Decision Variables

Variable	Meaning
p_{ilt}	units of technical i produced in time t sent to storage location *l*
q^{r}_{mt}	hours of production resource m's r[th] segment consumed in time t
p_{jlt}	units of finished product j produced in time t sent to storage location *l*
q^{r}_{nt}	hours of packaging resource n's r[th] segment consumed in time t
p^{s}_{jklt}	units of finished packaged product jk produced in time t sent to storage location *l* in scenario s
I_{ilt}	units of technical i at storage location *l* at the end of time t
I^{s}_{jlt}	units of finished product j at storage location *l* at the end of time t in scenario s
I^{s}_{jklt}	units of finished packaged product jk at storage location *l* at the end of time t in scenario s

d^s_{jklt}	units of demand for finished packaged product jk allotted to storage location *l* in time t in scenario s
u^s_{jklt}	units of lost sales for finished packaged product jk at storage location *l* in time t in scenario s

Constraints

There are 13 sets of constraints, as defined below.

Production Constraints on Technical

(1) $$\sum_l p_{ilt} = P_{it} \qquad i = 1,2 \quad t = 1 \ldots 12$$

p_{ilt} is the decision variable that represents how many units of technical i produced in time t go to technical storage location *l*. P_{it}, the number of units of technical i produced in time t, is found in RMIPROD.CSV

Production Constraints on Finished Goods

(2) $$q^r_{mt} \le Q^r_{mt} \qquad m = 1,2 \quad r = 1 \ldots R \quad t = 1 \ldots 12$$

q^r_{mt} is the decision variable that represents how many hours of production resource m's r^{th} segment are used in time t. The r^{th} production segment can not be greater than Q^r_{mt}, the size of its piecewise linear range. Q^r_{mt} is found in PRODCAP.CSV. These constraints are implemented with simple upper bounding.

(3) $$\sum_j \sum_l V_{jm} \, p_{jlt} = \sum_r q^r_{mt} \qquad m = 1,2 \quad t = 1 \ldots 12$$

p_{jlt} is the decision variable that represents how many units of finished product j are produced in time t and sent to storage location *l*. The hours required of production resource m in time t is the sum over all finished products j of $\sum_l p_{jlt}$ multiplied by V_{jm}, a scalar representing the number of hours of production resource m consumed in producing one unit of finished product j. This sum, the hours required, must equal the sum of the q^r_{mt}, which is the hours of production resource m consumed in time t. V_{jm} is found in PRODUSE.CSV.

Production Constraints on Packaged Goods

(4) $$q^r_{nt} \le Q^r_{nt} \qquad n = 1,2 \;\; r = 1 \ldots R \;\; t = 1 \ldots 12$$

q^r_{nt} is the decision variable that represents how many hours of packaging resource n's r[th] segment are used in time t. The r[th] packaging segment can not be greater than Q^r_{nt}, the size of its piecewise linear range. Q^r_{nt} is found in PACKCAP.CSV. These constraints are implemented with simple upper bounding.

(5) $$\sum_j \sum_k \sum_l V_{jkn}\, p^s_{jklt} = \sum_r q^r_{nt} \qquad n = 1,2 \;\; t = 1 \ldots 12 \;\; s = 1 \ldots 3$$

p^s_{jklt} is the decision variable that represents how many units of finished packaged product jk are produced in time t and sent to storage location *l* in scenario s. The hours required of packaging resource n in time t is the sum over all finished packaged products jk of $\Sigma\, p^s_{jklt}$ multiplied by V_{jkn}, a scalar representing the number of hours of packaging resource n consumed in producing one unit of finished packaged product jk. This sum, the hours required, must equal the sum of the q^r_{nt}, which is the hours of packaging resource n consumed in time t. V_{jkn} is found in PACKUSE.CSV.

Inventory Constraints on Technical

(6) $$I_{ilt} = I_{il,t\text{-}1} + p_{ilt} - \sum_j \sum_\lambda A_{ijl}\, p_{j\lambda t} \qquad i = 1,2 \;\; l = 1 \ldots L \;\; t = 1 \ldots 12$$

Inventory for each technical i at each storage location *l* at the end of each time t equals the units in inventory at the end of time t-1, plus production in time t, minus the units consumed in producing finished product in time t. The summation $\sum_\lambda p_{j\lambda}$ is summed over all WIP locations. A_{ijl} is a scalar that represents the units of technical i coming from storage location *l* required to produce one unit of finished product j. Its value is found in RMIUSE.CSV.

When t=0, I_{ilt} is the initial inventory of technical at storage location *l*, an input to the model found in RMIINIT.CSV.

(7) $$\sum_i S_{il} I_{ilt} \le W_{lt} \qquad l = 1 \ldots L \;\; t = 1 \ldots 12$$

The total storage space used by technical at storage location *l* in time t can not exceed the total storage space available at storage location *l* in time t. S_{il} is a scalar that represents how much storage space at storage location *l* is consumed by one unit of technical i. The value for S_{il} is found in RMIINV.CSV. W_{lt} is found in RMICAP.CSV.

Inventory Constraints on Finished Products

(8) $$I^s_{jlt} = I^s_{jl,t-1} + p_{jlt} - \sum_k \sum_l A_{jk} p^s_{jklt} \qquad j = 1 \ldots 7 \;\; l = 1 \ldots L \;\; t = 1 \ldots 12 \;\; s = 1 \ldots 3$$

Inventory for each finished product j at each storage location *l* at the end of each time t equals the units in inventory at the end of time t-1, plus production in time t, minus the units consumed in producing finished packaged product in time t. A_{jk} is a scalar that represents the units of finished product j required to produce one unit of finished packaged product jk. Its value is found in WIPUSE.CSV.

When t = 0, I^s_{jlt} is the initial inventory of finished product j at storage location *l* in scenario s, an input to the model found in WIPINIT.CSV.

(9) $$\sum_j S_{jl} I^s_{jlt} \le W_{lt} \qquad l = 1 \ldots L \;\; t = 1 \ldots 12 \;\; s = 1 \ldots 3$$

The total storage space used by finished products at storage location *l* in time t cannot exceed the total storage space available at storage location *l* in time t in any scenario. S_{jl} is a scalar that represents how much storage space one unit of finished product j consumes at storage location *l*. S_{jl} is found in WIPINV.CSV. W_{lt} is found in WIPCAP.CSV.

Inventory Constraints on Finished Packaged Products

(10) $$I^s_{jklt} = I^s_{jkl,t-1} + p^s_{jklt} - d^s_{jklt} + u^s_{jklt}$$

$$j = 1 \ldots 7 \;\; k = 1 \ldots 5 \;\; l = 1 \ldots L \;\; t = 1 \ldots 12 \;\; s = 1 \ldots 3$$

Inventory for each finished packaged product jk at each storage location *l* at the end of each time t for each scenario equals the units in inventory at the end of time t-1, plus production in time t, minus the units consumed in satisfying demands in time t in scenario s, plus any lost sales in time t in scenario s.

When t = 0, I^s_{jklt} is the initial inventory of finished packaged product jk at storage location *l*, an input to the model found in FGIINIT.CSV. This initial inventory is the same for all of the scenarios, since it measures the number of units at storage location *l* before any demands have materialized.

(11) $$\sum_j \sum_k S_{jkl} I^s_{jklt} \le W_{lt} \qquad l = 1 \ldots L \;\; t = 1 \ldots 12 \;\; s = 1 \ldots 3$$

The total storage space used by finished packaged products at storage location *l* in time t can not exceed the total storage space available at storage location *l* in time t. S_{jkl} is a scalar that represents how much storage space is consumed by one unit of finished packaged product jk. S_{jkl} is found in FGIINV.CSV. W_{lt} is found in FGICAP.CSV.

(12) $\sum_{l} d^{s}_{jklt} = D^{s}_{jkt}$ $j=1\ldots7\ \ k=1\ldots5\ \ t=1\ldots12\ \ s=1\ldots3$

All the units of demand for finished packaged product jk in scenario s, D^{s}_{jkt}, must be assigned to a storage location. D^{s}_{jkt} is found in DEMAND.CSV

End of Horizon Constraints

(13a) $I_{ilt} \geq E_{il}$ $i=1,2\ \ l=1\ldots L\ \ t=12$

(13b) $I^{s}_{jlt} \geq E_{jl}$ $j=1\ldots7\ \ l=1\ldots L\ \ t=12\ \ s=1\ldots3$

(13c) $I^{s}_{jklt} \geq E_{jkl}$ $j=1\ldots7\ \ k=1\ldots5\ \ l=1\ldots L\ \ t=12\ \ s=1\ldots3$

The units of inventory at the end of the last (12th) time period must be at least E, whose values are found in RMIEND.CSV, WIPEND.CSV, and FGIEND.CSV. These constraints are implemented with simple lower bounding.

Non-Negativity Constraints

$p_{ilt} \geq 0$ $i=1,2\ \ l=1\ldots L\ \ t=1\ldots12$

$q^{r}_{mt} \geq 0$ $m=1,2\ \ r=1\ldots R\ \ t=1\ldots12$

$p_{jlt} \geq 0$ $j=1\ldots7\ \ l=1\ldots L\ \ t=1\ldots12$

$q^{r}_{nt} \geq 0$ $n=1,2\ \ r=1\ldots R\ \ t=1\ldots12$

$p^{s}_{jklt} \geq 0$ $j=1\ldots7\ \ k=1\ldots5\ \ l=1\ldots L\ \ t=0\ldots12\ \ s=1\ldots3$

$I_{ilt} \geq 0$ $i=1,2\ \ l=1\ldots L\ \ t=0\ldots12$

$I^{s}_{jlt} \geq 0$ $j=1\ldots7\ \ l=1\ldots L\ \ t=0\ldots12$

$I^{s}_{jklt} \geq 0$ $j=1\ldots7\ \ k=1\ldots5\ \ l=1\ldots L\ \ t=0\ldots12\ \ s=1\ldots3$

$d^{s}_{jklt} \geq 0$ $j=1\ldots7\ \ k=1\ldots5\ \ l=1\ldots L\ \ t=1\ldots12\ \ s=1\ldots3$

$u^{s}_{jklt} \geq 0$ $j=1\ldots7\ \ k=1\ldots5\ \ l=1\ldots L\ \ t=1\ldots12\ \ s=1\ldots3$

Objective Function

The objective function consist of four costs: holding, production, distribution, and lost sales. The objective is to minimize the sum of these four costs.

Holding Cost

$$\text{Holding Cost} = \sum_t \sum_i \sum_l H_{il} I_{ilt} + \sum_s \sum_t \sum_j \sum_l \alpha_s H_{jl} I^s_{jlt} + \sum_s \sum_t \sum_j \sum_k \sum_l \alpha_s H_{jkl} I^s_{jklt}$$

The holding cost is found by multiplying the amount of product held at each stage in each time period by its associated per unit holding cost. These per unit costs are found in RMIHOLDC.CSV, WIPHOLDC.CSV, and FGIHOLDC.CSV. α_s is a scalar that places a weight on scenario s.

Production Cost

$$\text{Production Cost} = \sum_t \sum_m \sum_r C^r_{mt} q^r_{mt} + \sum_t \sum_n \sum_r C^r_{nt} q^r_{nt}$$

The production cost is found by multiplying the amount of each product produced in each piecewise segment in each time period by its associated per hour production cost. These per hour costs are found in PRODCOST.CSV and PACKCOST.CSV.

Distribution Cost

Distribution Cost =

$$\sum_t \sum_i \sum_l X_{il} p_{ilt} + \sum_t \sum_j \sum_l X_{jl} p_{jlt} + \sum_t \sum_j \sum_k \sum_l \sum_s \left[\alpha_s X_{jkl} p^s_{jklt} + \alpha_s Y_{jkl} d^s_{jklt}\right]$$

The distribution cost is found by multiplying the amount of each product distributed through each storage location in each time period by that storage location's per unit cost. X is the cost of shipping one unit of a product to the storage location; these per unit costs are found in RMIDISTC.CSV, WIPDISTC.CSV, and FGIDISTC.CSV. Y_{jkl} should be interpreted as the per unit cost of satisfying demand for finished packaged product jk at storage location *l*; and is found in DMNDCOST.CSV.

Lost Sales Cost

$$\text{Lost Sales Cost} = \sum_{s}\sum_{t}\sum_{j}\sum_{k}\sum_{l}\alpha_s U_{jkt} u^{s}_{jklt}$$

The lost sales cost is found by multiplying the amount of lost sales for each finished packaged product at each storage location in each time period by U_{jkt}, its associated per unit penalty cost. U_{jkt} is found in LOSTSALE.CSV.

Summary of inputs to the model

This table summarizes all of the input parameters to the model.

Variable	Definition	Containing File
P_{it}	amount of technical i produced in time t	RMIPROD.CSV
A_{ijl}	amount of technical i from storage location *l* required to produce one unit of finished product j	RMIUSE.CSV
A_{jk}	amount of finished product j required to produce one unit of finished packaged product jk	WIPUSE.CSV
S_{il}	amount of space consumed by one unit of technical i at storage location l	RMIINV.CSV
S_{jl}	amount of space consumed by one unit of finished product j at storage location l	WIPINV.CSV
S_{jkl}	amount of space consumed by one unit of finished packaged product jk at storage location l	FGIINV.CSV
W_{lt}	capacity of storage location l in time t	RMICAP.CSV WIPCAP.CSV FGICAP.CSV
D^{s}_{jkt}	amount of demand for finished packaged product jk in time t under scenario s	DEMAND.CSV
Q^{r}_{mt}	size of the r^{th} segment's range (in hours) for production resource m in time t	PRODCAP.CSV
Q^{r}_{nt}	size of the r^{th} segment's range (in hours) for packaging resource n in time t	PACKCAP.CSV
V_{jm}	hours of production resource m's capacity required to produce one unit of product j	PRODUSE.CSV
V_{jkn}	hours of packaging resource n's capacity required to produce one unit of finished packaged product jk	PACKUSE.CSV

H_{il}, H_{jl}, H_{jkl},	holding costs per unit at storage location l	RMIHOLDC.CSV WIPHOLDC.CSV FGIHOLDC.CSV
C^{r}_{mt}	production costs per hour of production resource m in piecewise segment r in time t	PRODCOST.CSV
C^{r}_{nt}	production costs per hour of packaging resource n in piecewise segment r in time t	PACKCOST.CSV
X_{il}, X_{jl}, X_{jkl}	per unit distribution costs to storage location l	RMIDISTC.CSV WIPDISTC.CSV FGIDISTC.CSV
Y_{jkl}	per unit cost of satisfying demand for finished packaged product jk at storage location l	DMNDCOST.CSV
U_{jkt}	per unit lost sale penalty costs for finished packaged product jk in time t	LOSTSALE.CSV
I_{ilt}	at t = 0, initial inventory of technical i at storage location l	RMIINIT.CSV
I^{s}_{jlt}	at t = 0, initial inventory of finished product j at storage location l in scenario s	WIPINIT.CSV
I^{s}_{jklt}	at t = 0, initial inventory of finished packaged product jk at storage location l in scenario s	FGIINIT.CSV
E_{il}, E_{jl}, E_{jkl}	minimum ending inventory at storage location l	RMIEND.CSV WIPEND.CSV FGIEND.CSV

Dimensionality – One Scenario

Constraints		
(Cons)	rows	Example
(1)	I T	24
(2)	M R T	0 (SUB)
(3)	M T	24
(4)	N R T	0 (SUB)
(5)	N T S	12
(6)	I L_I T	48
(7)	L_I T	24
(8)	J L_J T S	96
(9)	L_J T S	12
(10)	J K L_K T S	840
(11)	L_K T S	84
(12)	J K T S	480
(13a)	I L_I	0 (SLB)
(13b)	J L_J S	0 (SLB)
(13c)	J K L_K S	0 (SLB)
Total		1,644

Variables		
Var.	cols	Example
p_{ilt}	I L_I T	48
q^r_{mt}	M R T	24
p_{jlt}	J L_J T	96
q^r_{nt}	N R T	12
p^s_{jklt}	J K L_K T S	840
I_{ilt}	I L_I T	48
I^s_{jlt}	J L_J T S	96
I^s_{jklt}	J K L_K T S	840
d^s_{jklt}	J K L_K T S	840
u^s_{jklt}	J K L_K T S	840
Total		3,684

Data for example:

I = 2	L_I = 2	M = 2	R = 1
J = 8	L_J = 1	N = 1	T = 12
K = 5	L_K = 7	S = 1	J K L_K = 70

Note: Industrial LINDO permits 8,000 rows and 16,000 columns.

Dimensionality – Three Scenarios

Constraints		
(Cons)	rows	Example
(1)	I T	24
(2)	M R T	0 (SUB)
(3)	M T	24
(4)	N R T	0 (SUB)
(5)	N T S	36
(6)	I L_I T	48
(7)	L_I T	24
(8)	J L_J T S	288
(9)	L_J T S	36
(10)	J K L_K T S	2,520
(11)	L_K T S	252
(12)	J K T S	1,440
(13a)	I L_I	0 (SLB)
(13b)	J L_J S	0 (SLB)
(13c)	J K L_K S	0 (SLB)
Total		4,692

Variables		
Var.	cols	Example
p_{ilt}	I L_I T	48
q^r_{mt}	M R T	24
p_{jlt}	J L_J T	96
q^r_{nt}	N R T	12
p^s_{jklt}	J K L_K T S	2,520
I_{ilt}	I L_I T	48
I^s_{jlt}	J L_J T S	288
I^s_{jklt}	J K L_K T S	2,520
d^s_{jklt}	J K L_K T S	2,520
u^s_{jklt}	J K L_K T S	2,520
Total		10,596

Data for example:

I = 2	$L_I = 2$	M = 2	R = 1
J = 8	$L_J = 1$	N = 1	T = 12
K = 5	$L_K = 7$	S = 3	J K L_K = 70

Note: Industrial LINDO permits 8,000 rows and 16,000 columns.

HOW THE MACRO WORKS

This section describes how the conversion macro functions, and how it modifies the output files for better analysis. As always, the best description of how a macro operates is the source code, which can be viewed by selecting "Window| Unhide..." and select MONSMACR.XLS. The primary entry point for the macro (called when you select the conversion routine from the Monsanto menu) is the subroutine "OpenAndConvertMonsantoCSVs."

In essence, the main steps of the macro are:

1. Import the output files into one workbook as numerous worksheets
2. Format (prettify) the worksheets, and setup the autofilter for them
3. Setup all the pivot tables
4. Add the graphs to the usage sheets

1. Import the output files

The first step in importing the files is determining where they are located. Thus the macro will ask the user to find the first file. It then assumes that all remaining input files are located in the same directory. If not, the macro should fail with a file-not-found error. The function that handles importing is "ImportDataToWorkbook."

For each input file, the macro specifies three parameters. First is the file name itself. Second is the "type" of the data. This describes the data contained in the file. For instance, the output file COSTS.CSV has a type "Costs." By default, all output files are assigned a type equal to the file name without the ".CSV." This is because the type is used only to identify the data. Third is the name of the sheet to which the data is imported. By default, the sheet name is the type of the data with "Data" appended; the file COSTS.CSV is imported into a worksheet name "Costs Data." However, when several files are imported into one worksheet (as PILT.CSV and IILT.CSV are imported into "RM Data"), a worksheet name must be specified to the function.

During the import, the macro adds a column to the end of the data sets. Thus, the autofiltered list worksheets which replicate the output files are not actual copies of the output files, because of the added column. This extra column has a header of "TYPE," and the type for a given row corresponds to the "type" of the data assigned to the file from which that row generated. Thus, in the worksheet "CostsData" there is a last column with header "TYPE" in which all values in the list are "Costs."

For a list of all worksheets and the files whose data they contain, consult the table on page 22.

2. Format the worksheets

The following actions are performed on each worksheet once they are imported:

- Font size is set to 8 pt. (the default font type and style are retained; normally this is Arial regular).
- Autofilter is turned on. Search Microsoft Excel Help for "Autofilter" for more information on using the autofilter features.
- Columns are sized to fit all the data in the worksheet.

The subroutine which formats the worksheets is "FormatDataSheet."

3. Setup the pivot tables

Because of the nature of the pivot tables, there are separate macros for each pivot table. This was to insure sufficient customization for each table to display a default report format. The general procedure for creating a pivot table is:

1. Find the source data sheet and range
2. Invoke the pivot table wizard and setup the pivot field orientations
3. Format the fields and set default values
4. Zoom the window to an appropriate setting for viewing
5. Size the columns to fit data
6. Hide the original data sheet

The original data sheet is hidden because except for reduced cost and unit cost values, the pivot table for a given data set can display all values within that data set. If you need to see reduced cost or unit cost values for a data set, there are two ways to accomplish it:

- Unhide the original worksheet using "Format | Sheet | Unhide..."
- Add the reduced cost or unit cost field to the pivot table by invoking the pivot table wizard. To do this, make sure the active cell is within the pivot table, and select "Data | Pivot Table..."

4. Add the usage graphs

Two graphs are added by the macro. One each is placed onto the pivot table sheets for production usage and RMI usage. These graphs originally contain three series, defined by the default three columns in the pivot tables for those sheets. Excel provides intelligent graphing from pivot tables, so that as pivot tables are changed with pivots, the graph adapts to the new table structure. Note that Excel is not so successful with these graphs because they are complex combination graphs.

The procedure for setting up a usage chart is as follows:

1. Locate the pivot table being graphed
2. Invoke the chart wizard to create the groundwork for the chart
3. Set the chart title to reflect the resource being graphed
4. Format the chart to have the usage and capacity overlap, and other such fine tunings.