UCRL-ID-122511

User's Guide to the META•Net Economic Modeling System Version 1.2

Alan Lamont

November 24, 1994

Lawrence Livermore National Laboratory

Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

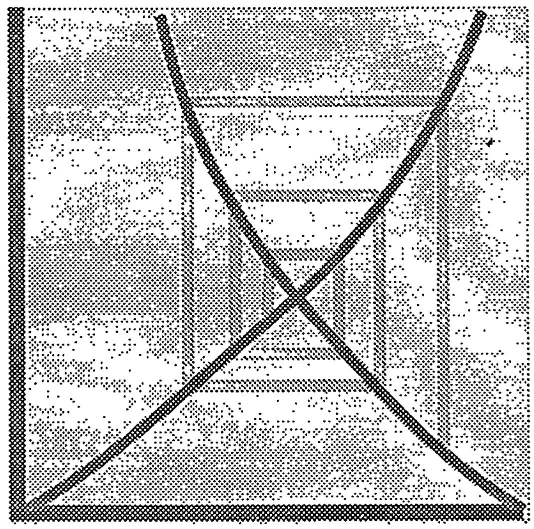
This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161

User's Guide to the META•Net Economic Modeling System

version 1.2



by:
Alan Lamont

Lawrence Livermore National Laboratory

November 25, 1994

User's Guide to the META•Net Economic Modeling System

Alan Lamont, LLNL

1. Introduction

In a market economy demands for commodities are met through various technologies and resources. Markets select the technologies and resources to meet these demands based on their costs. Over time, the competitiveness of different technologies can change due to the exhaustion of resources they depend on, the introduction of newer, more efficient technologies, or even shifts in user demands. As this happens, the structure of the economy changes. The Market Equilibrium and Technology Assessment Network Modelling System, META•Net, has been developed for building and solving multi-period equilibrium models to analyze the shifts in the energy system that may occur as new technologies are introduced and resources are exhausted.

META•Net allows a user to build and solve complex economic models. It models'a market economy as a network of nodes representing resources, conversion processes, markets, and end-use demands. Commodities flow through this network from resources, through conversion processes and market, to the end-users. META•Net then finds the multi-period equilibrium prices and quantities. The solution includes the prices and quantities demanded for each commodity along with the capacity additions (and retirements) for each conversion process, and the trajectories of resource extraction. Although the changes in the economy are largely driven by consumers' behavior and the costs of technologies and resources, they are are also affected by various government policies. These can include constraints on prices and quantities, and various taxes and constraints on environmental emissions. META•Net can incorporate many of these mechanisms and evaluate their potential impact on the development of the economic system.

This guide provides an overview of Version 1.2 of META•Net. It is not a complete documentation in that it does not contain all of the equations used by the system. It does discuss the modeling approach and provides instructions for building and running models with META•Net.

Overview of the META•Net Modelling System

The system consists of two parts, a library of standard modules that represent different types of processes (nodes) in a network, and an algorithm for finding the equilibrium solution to a model. To design a model, the user specifies nodes, their economic parameters and their connections to other nodes. The system will read the files containing the definition and assemble the network model. Using an iterative algorithm, it computes the multi-period solution to the network (equilibrium prices, quantities, capacity additions, etc.)

In META•Net each node is an object that sends messages about prices and demands directly to its neighbors in the network. Within each node there are several standard components that manage such tasks as communicating with other nodes, maintaining information about prices and quantities from previous iterations, and which can model blocks of capacity such as vintages. The economic parameters for the nodes include the demands functions for end-use nodes, costs and efficiencies for conversion nodes, and the costs for various levels of output in the resources nodes.

The signals that are passed between nodes have a standard structure that is used throughout the network. The fact that nodes have standard components and that standard signals are passed between nodes makes it relatively easy to develop new nodes: The procedures and data storage required for communicating with other nodes is provided by the system and the analyst does not need to be concerned with it.

One of the advantages of the network modeling approach is the fact that it can be easily extended to model new kinds of issues. Each of the nodes is independent of the others and different types of nodes can, in principle, represent a wide range of economic behaviors (e.g. a node can be written that models the decision to retro-fit facilities with emissions control equipment in response to new environmental regulations). However, incorporating new behaviors into the node models requires that new programs be written. This task can be very complex because the program must manage all the variables that define the state of a node and all the price and quantity variables that are passed between nodes. META•Net has been explicitly designed to make the job of modifying node models, or writing new ones as simple as possible. It is written in an object oriented language (object Pascal) which simplifies many of the data handling tasks that arise each time a node model is executed. The node models are also designed so that communication between nodes is all handled by standard managers. This reduces the programming effort and complexity of developing new types of nodes in the future.

Currently META•Net runs on a Macintosh computer with a 68020 processor, or higher, and a math co-processor. The amount of memory required depends on the size of the model to be run. A 20 node model require on the order of 550 K bytes and a 100 node model requires on the order of 1.8 M bytes.

Current status of the of META•Net software

At this time, META•Net is ibeing released as a Beta version. It has been been used on a few internal projects and demonstrations. Any errors discovered during these projects have been corrected. However these projects did not necessarily test the software over the full range of applications and model configurations that users may choose. Therefore, additional errors may be encountered by other users.

The software is available on a collaborative basis from Lawrence Livermore National Laboratory. Under this arrangement, users can use the software provided they agree to certain copyright restrictions. As collaborators, it is anticipated that users will report suspected errors and assist in correcting them.

It is expected that further enhancements of the software will be made in the future. It is hoped that collaborative users will make suggestions for enhancements that will help them in their own applications of META•Net.

The sections below first describe theoretical basis for the network approach in more detail. Then the structure of META•Net is described. The sections after that describe the node types that are included in the current version of META•Net and their input parameters. The final sections describe the process of building and running a model.

2. The Network Modelling Approach

The network approach represents the economy as a network of nodes. Each node represents various real entities in the system: end-users, conversion technologies, resources, etc. A node is actually a module that models the response of one of the entities to changes in input prices and quantities demanded. The network passes information about prices and quantity flows between the nodes. An iterative solution algorithm finds a solution in which all of the demands are in equilibrium with the prices.

This approach was developed as a way to organize large scale models of the energy system into modular structures that are easier to construct, modify, and analyze. It was originally used in the SRI-Gulf model of the energy system (Cazalet). The software architecture of that model, however, was difficult to modify. Subsequently, two systems were developed to make it easier to build and modify energy models. One of these is the Livermore Economic Modeling System (LEMS) developed at Lawrence Livermore National Laboratory (Rousseau), and the other is the Generalized Equilibrium Modeling System (GEMS) developed by Decision Focus Inc. (Decision Focus Inc.) Both of these systems are built around a library of standard modules that represent a variety of processes.

None of the existing systems takes advantage of recent advances in software architecture such as object oriented programming. Consequently, modifying modules or creating new behaviors requires explicitly programming a complex series of calls to internal data storage structures. The external databases that store information about the structure and parameters of a model are designed to be complete, but they are rigid and it is difficult to adapt them to make the model building and analysis process easy for the user. The META•Net system uses object oriented programming and is designed to be easy to modify to create new kinds of modules or modify the behavior of existing ones.

Theoretical Background

Hogan and Weyant discuss both the value this approach and its theoretical basis. They point out that because of the size and complexity of many energy models, they are often difficult to develop, understand, and use. To alleviate this it is important that models have the following characteristics:

- a consistent underlying theory,
- a model structure that corresponds to the natural organization of the data and system,
- a modular design,
- decentralized implementation, and
- efficient computation.

These characteristics are met through the network approach to modeling: The nodes are small modular models of particular processes in the real world. The nodes exchange well defined, standard information about quantities and prices. Therefore, the structure of the system corresponds to the natural organization of the data and system in that each module corresponds to an industry or even a process within an industry. The modules, and the well defined protocols for communication, lead to a modular structure and make it possible to decentralize the development of the various pieces of the model. The network approach does not, in and of itself, lend itself to efficient computation. However, when considering the total effort in building, maintaining, and modifying a large model, it is expected that the network approach would lead to significant overall savings.

Much of the Hogan and Weyant paper is devoted to developing the underlying theory of network modeling. Essentially, they show that the network approach can be derived from, and is equivalent to, a standard optimization formulation for economic models—the maximization of consumers' utility subject to feasibility constraints. They show that this problem can be reformulated to decompose it into a set of smaller optimization problems corresponding to consumers and processes (e.g. producers). In these smaller problems the consumers chose their consumption levels to maximize utility and the producers choose their input and output levels to maximize profits, given the prices of inputs and outputs. The input and outputs from each process must also satisfy the technological constraints on that process. The overall model must find a set of prices such that all quantities demanded and produced meet the feasibility constraints that quantities produced from one node must match the input quantities at the node using the output. The prices also must be consistent between between nodes in that the output price at one node must match the input prices at the node using the output.

Hogan and Weyant show that as long as these modules solve their local optimization problem correctly, the entire system will solve the original problem correctly. They also point out that solution of this optimization problem is equivalent to finding an equilibrium set of prices and quantities for the system.

It is true that many large scale energy models are, in one way or another, network models. They are comprised of a set of smaller models that pass information about prices and quantities back and forth to arrive at an equilibrium solution. Typically, these smaller modules represent entire sectors rather than lower level processes. Many of these have been developed in an *ad hoc* manner without ensuring that a consistent theoretical approach be used throughout. The value of a consistent underlying theory lies partly in ensuring that the overall system solves a well defined problem. It is also essential for defining a consistent set of interfaces between modules. The theory implicitly specifies what information is to be passed and what the exact definition of the information is. Because of the underlying theory, one can be assured that that set of information is sufficient. Thus, when writing new modules one can depend on specific information being available from other modules, and one knows exactly what information must be produced by a module if it is to be consistent with other modules.

In summary, the modular organization of the network model and the consistent use of an underlying theory makes it possible to develop systems of models that are truly modular and that have a consistent clear interpretation.

Network Solution Algorithm

The model solves for equilibrium prices and quantities through an iterative solution algorithm which is a version of the Gauss-Seidel algorithm. A single iteration calculates quantities and prices for each period. The model is solved for each period using a two pass algorithm. In the "down pass", quantities of energy demanded by end-user nodes are passed down through the network. On the "up pass", the prices required by each process are computed and passed up through the network. During each pass, many of the calculations require variable values that have not yet been calculated for that iteration. In those cases, values calculated during the previous iteration are used.

Figure 1 shows an example network that will illustrate the procedure. The calculation for a period starts at the end-use nodes. These use the prices from the previous iteration to calculate the quantities that they demand for that period and then passes the quantity down to the nodes that supply them. When the supplier nodes receive a quantity demanded on the down pass, they determine how much input they require from their supplier nodes it in order to supply the required quantities. This calculation is based on the model of its technology and it ensures that the technological constraints are always met. In the case of nodes that actually supply several other nodes (for example, the market nodes in Figure 1), the total quantity to be supplied is added up and passed down.

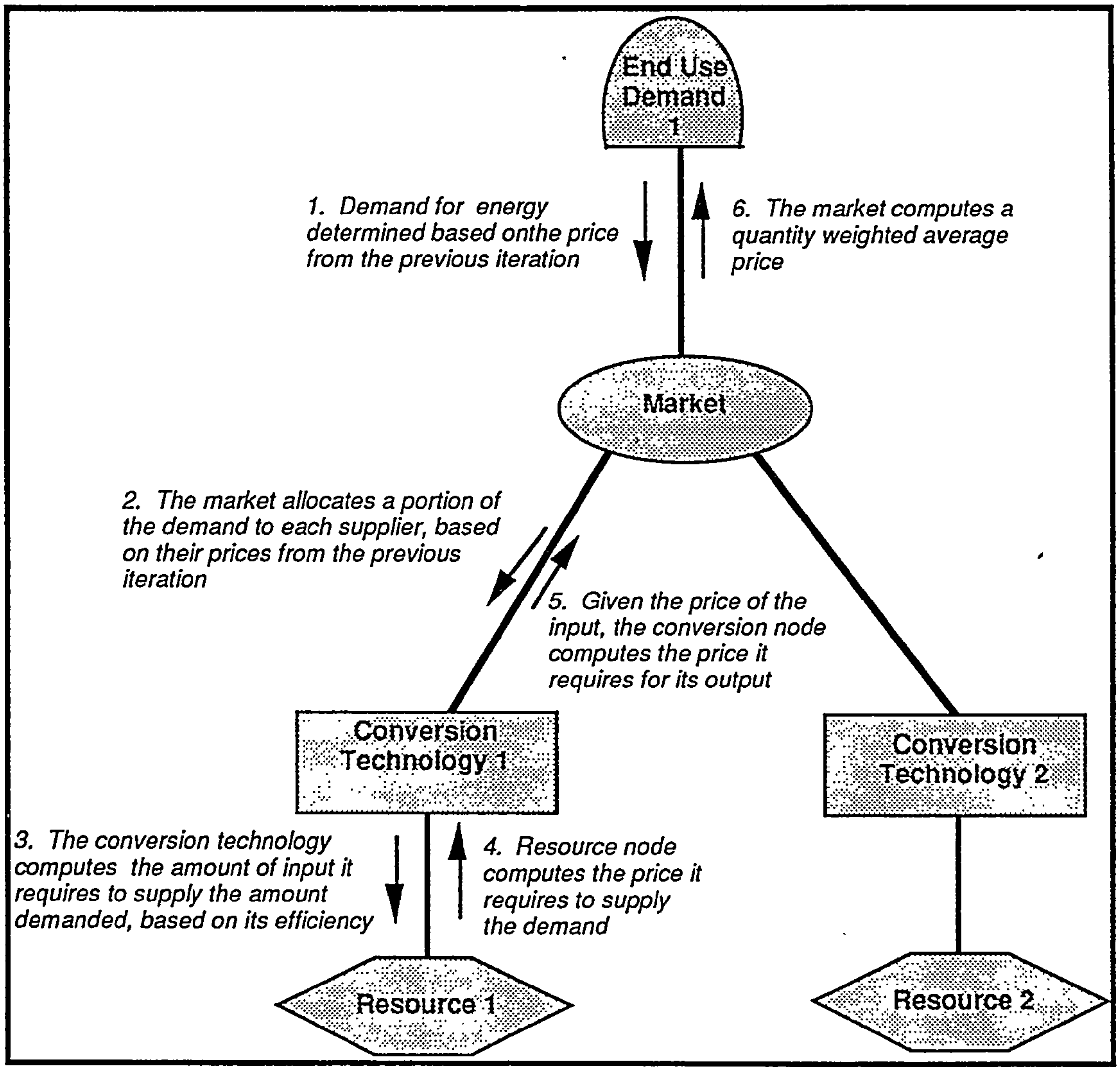


Figure 1: The solution is calculated using an iterative, two pass algorithm

When all the demands have been passed down to the resource nodes, the "up pass" is started. The resource nodes determine the price they require to meet the demands that were passed down to them. These prices are passed back up through the network. Each node receives the price of its inputs and calculates the corresponding price for its outputs. The node must take into account its fixed costs, variable costs, and efficiency.

The prices are passed up to the end-use nodes which then recalculate their demands. The demands are calculated from a model of consumer behavior (e.g. a demand function) which is assumed to model the consumers' utility maximization. These demands are then passed down to start the next iteration.

This process gradually converges to a market equilibrium. The convergence process can be illustrated graphically using the diagram in Figure 2 . This shows the supply and demand curves for a market. The convergence algorithm is analogous to computing the market equilibrium through what is known as a "cobweb algorithm" which uses the the following steps. 1) a initial quantity is chosen (it can be arbitrary), 2) use the supply curve to determine the price required to supply that quantity (this corresponds to the down pass in the algorithm), 3) use the demand curve to compute the quantity that would be demanded at that price (this is the up pass), 4) return to the supply curve to find the price that would be required to supply that quantity (another down pass) and 5) use the demand curve to determine how much would be demanded at that price (a final up pass). This brings us back to the starting point and a new series can be started. If the supply and demand curve has the correct slope, this process will gradually spiral into the equilibrium point.

Frequently the supply and demand curve do not have the correct slope to converge. Figure 3 illustrates that case. Here the cobweb algorithm will lead to divergence rather than convergence. "Relaxation" is used to make the model converge. Instead of adjusting the prices and quantities by the full amount indicated by the computation from the supply and demand curves, the prices and/or quantities are only changed by a fraction of the amount calculated. This will cause the algorithm to converge at the equilibrium point.

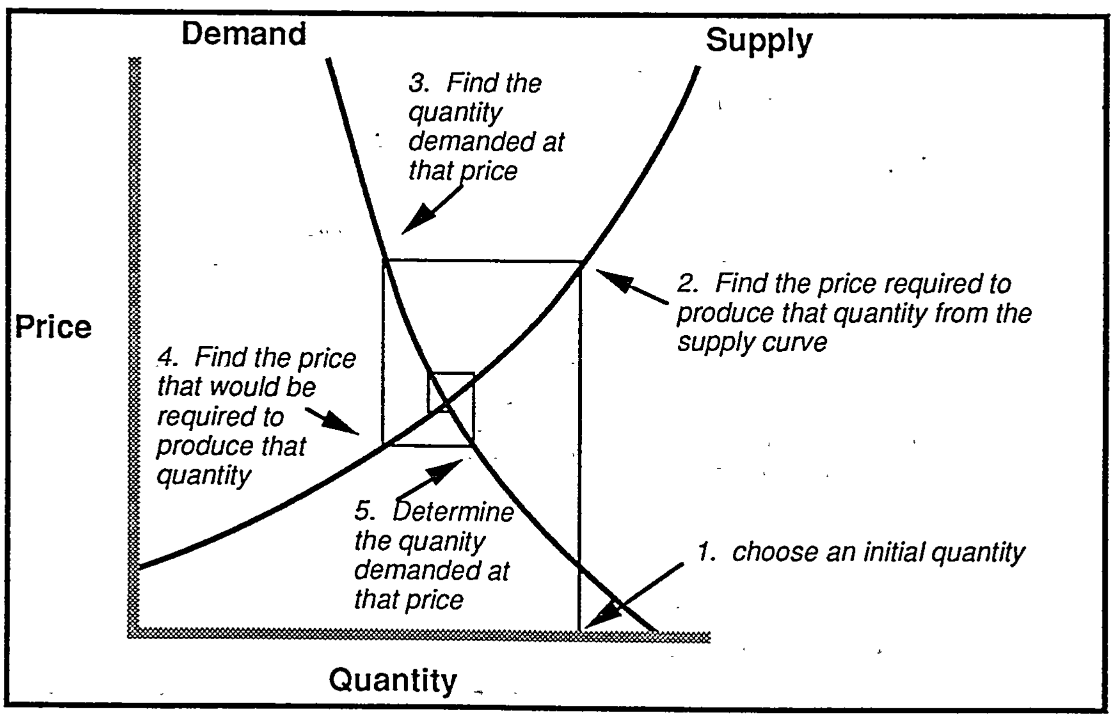
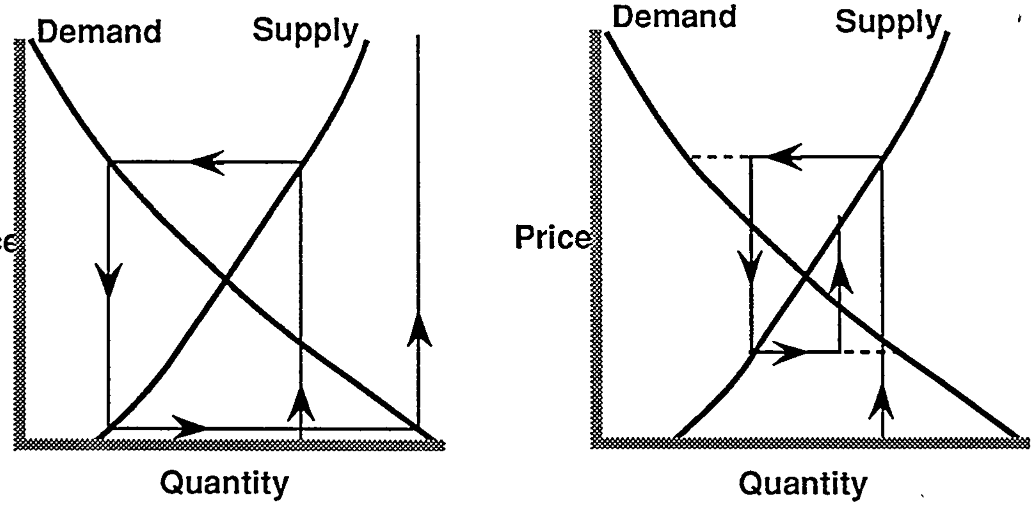


Figure 2 : The convergence process can be visualized using market demand a supply curves.



The slopes of the demand and supply curves lead to divergence

By reducing the change in quantities, the algorithm can be made to converge

Figure 3 : The slopes of the supply and demand curves can cause the algorithm to diverge. The relaxation procedure is used to make the algorithm converge.

Implementation of Constraints

In some cases constraints are applied to nodes. These will typically be constraints on capacity or on prices. Constraints are needed to simulate some type of regulations and to simulate other types of real situations such as the case where it is known that capacity for a particular process is physically limited. Constraints are implemented through the prices sent on the up pass.

Quantity and capacity constraints are implemented in a natural way: When the amount demanded exceeds the constraint, the price is raised until the quantity demanded is equal to the constraint. This essentially results in a supply curve that becomes very steep when the constraint is reached. As the price increases, less demand is allocated to it.

In the case of price constraint, a producer will increase production up to the point that the costs of production are equal to the price received. In the model we want to be sure that the amount demanded from a node is consistent with the constraint price—that is, the marginal cost of production at that demand is equal to the constraint price. Price constraints are implemented through a second set of prices. Each price signal actually consists of two prices: there is the "transaction price" which is the price paid by buyers. The second is the "constraint price" which is used to model constraints. The markets actually allocate based on the constraint price not on the transaction price. If there is no constraint, or it is not active, the constraint price is set equal to the transaction price. However, when a constraint becomes active, say when the demand to a price controlled node is so large that the price to be charged would exceed the price constraint, the constraint price is raised. This reduces demand to the node until the quantity produced by the node is consistent with the price constraint.

3. Components of the META•Net System

The META•Net system provides components for modeling the economic processes, managing data flows and solution algorithm, and for managing interactions with the user. This section introduces the major components.

Model Network

The network is a network of nodes where each node represents a process. The processes include resource extraction, conversion, end-uses, and markets. The network is built by the user by selecting node types from a library of nodes, specifying their connections to other nodes, and specifying their parameters. The nodes implement the solution algorithm by exchanging signals about quantities and prices.

There is one special node called the "top node" which initiates the message passing for each iteration of the model (shown as the parallelogram at the top of the network). As the algorithm proceeds, the Control Panel sends instructions to the top node to cycle the network for each iteration. The top node then sends instructions to each of the end-use nodes to begin a cycle of the network. From then on the nodes in the network will pass the signals back and forth needed to complete the iteration.

Components to Manage Communication Between Nodes

For all nodes, communication with other nodes is handled by the UpLink Manager and the DownLink Manager (An UpLink is an output link and a DownLink is an input link.) These managers perform the following functions:

- Maintain information about the node at the other end of the link, including such things as its identity, its type, or the type of pricing it receives.
- Maintains parameters about the operational relationship with the node at the other end of the link. For example, the manager keeps information about the input-output coefficient for conversion nodes.
- Maintain information about the most recent price and quantity signals sent and received for the past several iterations (the current version it keeps information for the past three iterations).
- A DownLink Manager a) sends quantity signals to the appropriate nodes after they have been prepared by the node, and b) records price signals sent to its node from other nodes.
- An UpLink Manager a) sends price signals to the appropriate nodes, and b) records quantity signals sent from other nodes.

The Link Managers have been designed to ensure that analysts who are designing new nodes have a wide range of information readily available. They also help programmers develop node models in a consistent way. Once a node has been developed, they are transparent to the user.

Other components manage the specific tasks of the nodes such as making allocations in the market nodes or keeping track of capacity additions or retirements in the conversion nodes. These components are described under the discussions of each node.

Price and quantity signals are passed between the nodes each time the network is cycled. Both kinds of signals carry header information that names the node that sent the signal. The rest of the data in the signal carries price or quantity information.

As will be described later, conversion nodes can be structured around several blocks of capacity where each block may model a set of facilities that are operated at a particular load factor, or a vintage of capacity. It is often necessary to send a separate quantity signal to each block of capacity, and each block may generate a different price. For this reason, the price and quantity signals can carry an array of price or quantity information, one element for each block of capacity.

In the quantity signal the information consists of the quantity demanded and the load factor at which it is to be delivered. There also provisions for including a load duration curve, however, this is not used by the nodes included in version 1.2.

The price signal includes the transaction prices and constraint prices demanded by each block of capacity. There are also provisions for including information about the capacity available in each block, although, again, this is not used in the nodes included in version 1.2.

User Interface

There are a number of additional components to the system that manage the user's interface and the execution of a run. These are shown in Figure 4 and include the *Control Panel* that interacts with the user and carries out the instructions to load and run the model, the *error manager* that detects and handles various errors during loading and running, the *convergence monitor* which gathers and displays information about convergence, and the *reader* which reads and writes files.

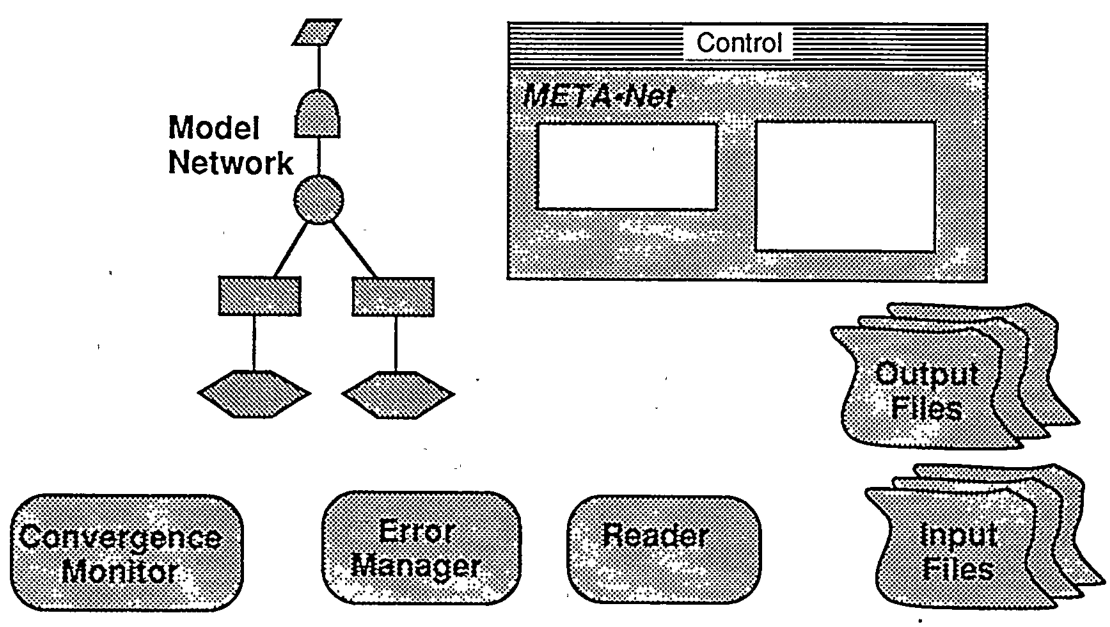


Figure 4 : The META•Net system consists of several interacting components

Control Panel

The Control Panel presents the window that displays information to the user and allows the user to give the system instructions. One window in the control panel displays messages from the system to the user, and the other window displays information about the state of convergence.

When the user instructs META•Net to load a model, the Control Panel instructs the Reader to read in the data about the nodes. It then instructs each node to initialize itself. During this initialization process, all of the links in the network are established so that the network is completely specified.

Error Manager

As the model is loading and running, each node checks for various types of errors such as variables out of range, or errors in specifying the links of the network. When a node detects an error, it sends a message to the Error Manager describing the error. The error manager writes an entry in the Error File (one of the output files) and then determines what to do with the error. In some cases the error is fatal and the run cannot continue. In that case it sends a message to the Control Panel to terminate the run. The Control Panel gives a message to user and then tries to save all files. In other cases it is possible to continue and the Error Manager instructs the Control Panel to give a message to the user and continue.

Convergence Monitor

During an iteration, the end-use and the resource nodes report information about the relative change in quantity from iteration to iteration to the Convergence Monitor. As each report comes in the Convergence Monitor keeps track of the nodes that have had the largest changes since the previous iteration. After an iteration is complete (i.e. the net has been cycled for each of the periods), the Convergence Monitor sends a message to the Control Panel identifying the node that had the largest change, and the period in which that change occurred. The Control Panel then displays the information to the user.

Reader

The Reader manages the interaction between the system and the input and output files. On input, it searches the files for records having specific headers. For output, the Reader constructs records as needed by the other components (including the required header information) and writes them to the files.

4. Structuring a Model in META•Net

This section describes the approach to building a model. For the most part, this is very straightforward: the user develops a network diagram, and specifies the nodes and their parameters. The major part of this section describes the modeling approaches and special node models used for modeling situations that are not straightforward. The first of these is the electric utility sector which requires special nodes to account for the fact that energy is demanded at several different load factors. Next we discuss the case of cyclic networks. This arises when some of the output of a node is used to create its own input. Cyclic networks are modelled by inserting a special type of link. Although it was stated earlier that a conversion node can only have a single output, this is not a completely rigid rule. It is possible to model some simple situations where a node has more than one output. The end of the section discusses the techniques used for implementing constraints, some considerations about the units used to describe quantities and prices in the model, and the time structure of a model

There are five basic node types available in version 1.2 of META•Net. These are sufficient for modelling a wide range of economic situations, particularly if they are combined to provide specialized behaviors.The five node types are: top, end-use, conversion, market, and resource nodes. The symbols used for these types in network diagrams are shown in Figure 5 .

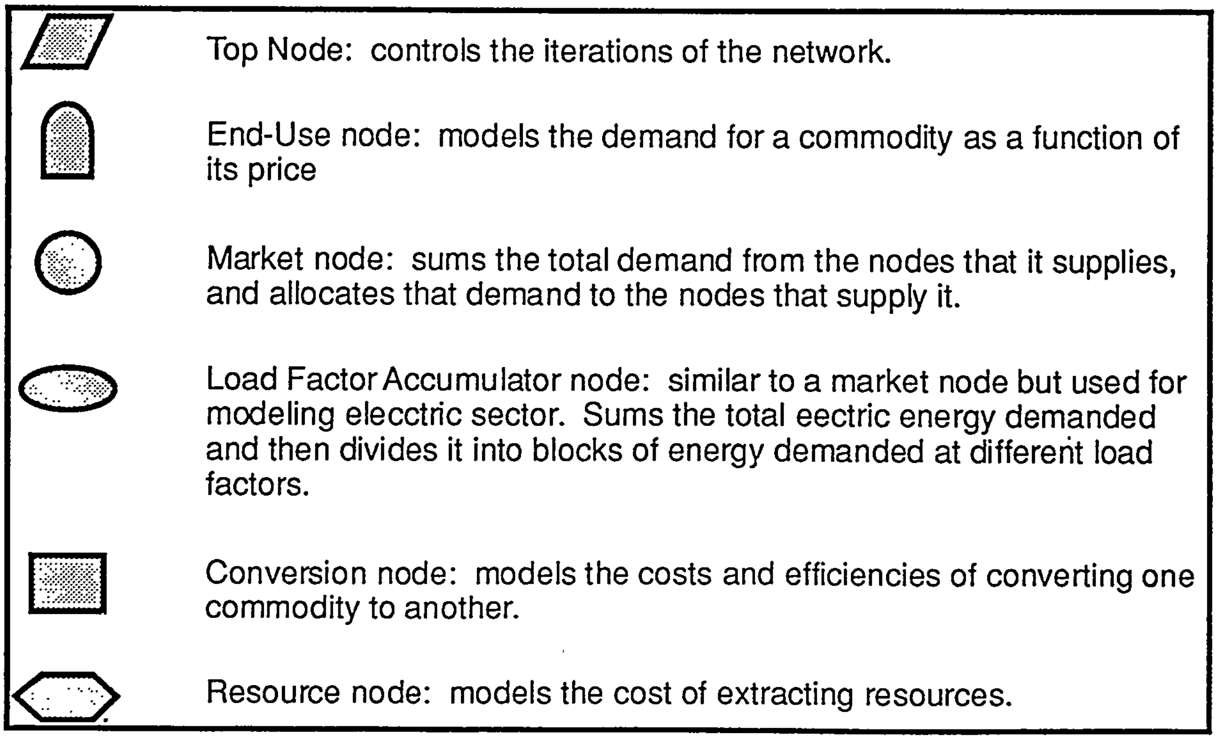


Figure 5: Basic node types are used in META•Net.

Several of these five basic node types have more than one variation to model several different situations. The variations are discussed in the detailed descriptions of the nodes in Section 5.

To illustrate the process of designing a network and developing a model, a simple model called "Sample" is included in Appendix A. Sample demonstrates all of the features that are built into version 1.2 of META•Net. Appendix A includes the diagram of the network and copies of all of its data files.

By convention, networks are structured so that resources are at the bottom of the network and end use nodes are at the top. Thus, commodities flow from the bottom to the top. Drawing the network is generally straightforward, however, there are a few rules to keep in mind and techniques for modeling special situations. The section describing each of the node types describes the network connections that each type of node can have. META•Net will (usually) give an error if these rules are not followed.

Modeling the Electric Utility Sector

META•Net uses a simplified model of the electric utility sector similar to what is used in the LEMS and GEMS modeling systems Although a simple approach is used here, more accurate node models for modeling the electric utility capacity additions and dispatch have been developed (Lamont, 91). The current version of META•Net models the load duration curve as three blocks of energy representing peak, intermediate, and base load demand. The user specifies the fraction of electric energy that is in each category and their load factors. Each block of energy is allocated to the electric generation technologies according the price that each technology requires for providing energy at the load factor for that block. Because the price that each technology requires depends on the load factor, baseload technologies are allocated essentially all of the baseload demand, while the peaking technologies are allocated essentially all of the peak demand.

This approach to modeling the electric utility sector is implemented through three types of nodes: the Load Factor Accumulator Node, the Multiple Load Factor Market Node, and the TechBlock3 conversion Node. An example of the structure is shown in Figure 6 .

Each of the demand nodes sends down demands for total energy to the Load Factor Accumulator Node (note that the accumulator does not have to receive demands directly from end use node—any other nodes that demand electrical energy can send a signal to the accumulator node.) These signals specify total energy and do not differentiate between load factors. The accumulator then divides the load into three parts, one for each load factor category. Thus the accumulator models the load duration curve. The fractions for each load category, and the load factors for each category, are input by the user when the load factor accumulator node is specified. The accumulator then sends down the three-part quantity signal to the Electric Market Node specifying the energy in each load factor category and the load factor for each category.

The Electric Market Node allocates each of the blocks of energy to the nodes that supply it. Because each technology requires a different price for suppling energy at a different load factor, each technology will be allocated a different fraction of each block. The Electric Market Node is actually three markets in one node—one market for the peak, intermediate, and base load quantities. Within the Electric Market node are three "allocators" that do the actual allocations. When the Electric Market Node receives the quantity signal, it sends the peak load quantity to one allocator, the intermediate load quantity to the second, and the base load quantity to the third. Each allocator allocates its assigned energy to the electric generating technologies according to the price that they charge for generating energy at the load factor for the allocator. The allocators pass these allocations back to the Electric Market node which assembles them into quantity signals for each of the electric conversion technology nodes below. Thus, each electric generating technology will also receive a three part quantity signal, specifying the amount of energy it is to supply at peak, intermediate, and base load factors.

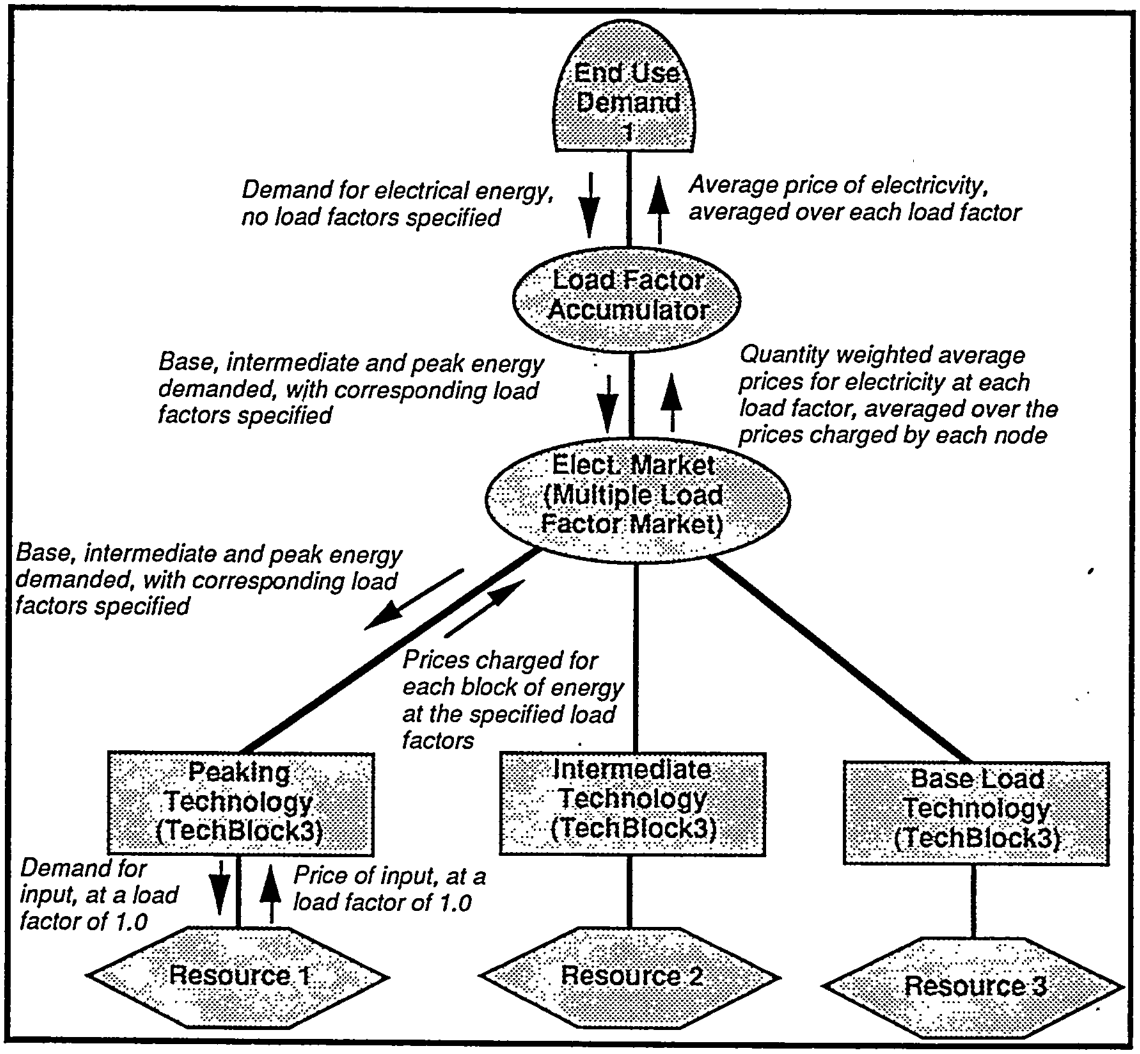


Figure 6 : The electrical sector is modeled using special nodes to accumulate electrical demand and divide it into peak, intermediate, and base load demand, and a market node and conversion nodes that can accommodate energy at multiple load factors

The electric generating technology nodes are represented by TechBlock3 nodes. Each TechBlock3 node represents three blocks of capacity. Each block operates at one of the load factors. When the three part quantity signal is sent down, the node passes each part to the corresponding technology block. The block then computes the inputs it requires to provide that quantity. The node assembles all of the inputs required by the three block and prepares quantity signals to send down the its input nodes. In the figure, these are simply shown as resource nodes. The signal that the TechBlock3 node sends to a supplier has just a single quantity that is to be supplied at a load factor of 1.0.

On the up pass, each TechBlock3 node receives price signals from each of its input nodes. This price is sent to each of the TechBlocks. The TechBlocks then calculate the price that they must charge in order to provide energy at the load factor that they serve. The TechBlock3 node assembles these prices into a three part price signal and sends it up tot he Electric Market node.

The Electric Market node receives price signals from each for the electric conversion nodes. Each price signal from each supplier carries the price that the node requires for supplying energy at each load factor. The Electric Market node then sends these prices to the corresponding allocator. Then, on the next down pass, each allocator has the prices that each of the electric generation nodes requires to supply energy at the allocator's load factor. It can then allocate the demand among the electric generation nodes based on these prices.

The Electric Market node computes the average price of electricity at each load factor, weighted by the quantity supplied by each technology and passes these prices up to the Load Factor Accumulator. The Load Factor Accumulator then computes the quantity weighted price of electricity, averaged over the three load factors.

Cyclic Networks

In some cases some of the output from a given node is required to produce the input for the same node. An example is shown in Sample. Here electric energy is needed to enrich nuclear fuel, which is, in turn, used to fuel a nuclear electric generator. Without special structuring, such a situation creates a "cyclic" network—during a single iteration an endless cycle of price and quantity signals would be sent around the loop from the nuclear generator, to the enrichment process, and back up the nuclear generator.

When it is necessary to model such a situation, a special type of link (known as a "loop" link) is used to connect the output of a market node to the input of a conversion node. When such a link is used, the market node handles the price and quantity signals so as not to allow a continuous cycle of signals to be sent.

The loop link is designated in the "Links" file described below.

Modeling Secondary Outputs

Although a conversion node is only allowed to have one output link, it is possible to model the case of multiple output products when there is a primary and a secondary product. The primary product is sent up on the output link. The network demands a specific quantity of the primary product depending on its price. The secondary product is produced in proportion to the primary product. If there is another market in the model for the secondary product, the two product case can be modelled.

An example of this s the cogeneration node in the Sample model. Here the primary product is heat and the secondary product is electricity. The industrial heat market demands heat based on its price. A quantity of electricity is generated as a secondary product in proportion to the amount of heat demanded. This secondary product is modelled using an input link with a negative input-output coefficient which is connected to the *output* of a market node (in this case a load factor accumulator which is a kind of market).

When the node receives a demand for heat on the down pass, it will send down a demand for fuel and a negative demand for electricity. This negative demand for electricity is subtracted from the total demand for electricity at the accumulator. When the conversion node receives a price for electricity on the up pass, the negative input-output coefficient will result in a negative cost for electricity and thus reduce the cost of heat production and the required price of heat.

There are some limits to the amount of a by-product that a node can produce and still lead to a valid model. A node should not produce so much of a by-product that it overwhelms the market node and results in a net negative demand. Also, a node should not earn so much from a by-product that the required price of the output is negative. In either case the result would be invalid, possibly leading to other internal errors as well.

Units for Measuring Quantities and Efficient Scaling of Units

The units used for measuring quantities can, in principle, be whatever is convenient for the user. The essential rule is that the units must be consistent throughout. That does not mean that the same units *must* be used at all points in the model. For example, electricity *could* be measured in megawatt hours, and coal *could* be measured in tons, as long as the input-out coefficient on the coal fired generator properly converted the demand for megawatt hours into the demand for tons of coal. The price curve for the coal resource must also be calibrated in terms of dollars per ton of coal.

To avoid confusion and blunders, it is preferable that the entire model use the same units.

Another issue in developing a model is referred to a "scaling" the units. There are some combinations of price and quantity units that do not work very well in some circumstances. For example, the model could use kilowatts for quantities and dollars/kilowatt for prices. This leads to very large quantities and very small prices which can lead to very slow convergence, especially when constraints are involved.

Units that seem to work well are millions of Btus and megawatt hours.

Defining Constraints

META•Net can impose quantity and price constraints on the conversion nodes. The general mechanism that imposes constraints through constraint prices is discussed above . This section describes the functions used for setting constraint prices and their parameters.

The constraints are actually imposed on the Tech Blocks not on the node as a whole, since the technology block represent the physical capacity of the node.

At the last step in processing the price signal in the Tech Block there is a check to determine whether or not there is a quantity or a price constraint on the block and, if so, whether or not it has been violated. If it has been violated, the procedure adjusts the constraint price to bring the block back into compliance with the constraint.

A Tech Block can have either a quantity constraint or a price constraint, but not both. If there is a quantity constraint, the block raises its transaction price and constraint price until the demand is reduced down to the constraint value. If there is a price constraint, the block raises only its constraint price. This drives the demand to the block down and this, generally, will reduce the price required by the block until it is equal to the price constraint.

There are cases where the price of the input to a node is not price sensitive (for example, if the node is supplied by a price track resource). The required price from the block will not be reduced no matter how small the demand to it becomes. Consequently the price constraint is not met. However, the demand to the node is close to zero, indicating that it has been driven out of business, as expected.

When the demand to the block exceeds the quantity constraint the constraint price is increased by a factor that depends on the ratio between the actual demand and the constraint quantity. In this way, the constraint price is gradually increased until the constraint is met.

The adjustment is done by multiplying the constraint price from the previous iteration by a "penalty factor" which is computed as:

$$\text{Penalty factor} = A * (R^B - 1) + 1$$

where:

A, B = input parameters

$$R = \min\left(\frac{\text{Q demanded}}{\text{Q constriant}}, \text{MaxConstraintRatio}\right)$$

When the quantity demanded is equal to the constraint, R equal 1, the penalty factor equals 1. If R is greater than 1, the penalty rises at an exponential rate up to a maximum determined by the MaxConstraintRatio which is also an input parameter to the Tech Block. The use of the MaxConstraintRatio ensures that the penalty factor will not cause an overflow if the quantity demanded is much larger than the constraint quantity.

At each iteration the new constraint price is computed as:

New constraint price =

(Last iteration constraint price * Penalty factor).

Thus when R is 1.0, no further adjustment occurs from iteration to iteration.

Note that if the constraint is not violated, the penalty function is not applied. If the demand is less than the constraint, the constraint price is adjusted downward toward the transaction price.

In the case of a price constraint, the process is similar except that the ratio R is the ratio of the required price of the node and the constraint price.

By adjusting the constraint parameters the user can make the constraint more or less abrupt. The steeper they are (e.g. large values for A and B) the more strongly the prices are steered towards the solution values. However this can lead to overshooting the solution values and slowing down convergence. The best values are found through experimentation. A starting value of 1.0 for A and 3.0 to 5.0 often work well for B. The MaxConstraintRatio usually works well at 2.0 or 3.0.

Time Structure of a Model Run

A model run analyzes a scenario covering a of a set of periods with a fixed number of years per period—for example, ten periods of five years each. An equilibrium is found for each period. It is assumed that prices and quantity flows are constant during the period.

In specifying the model, and in the internal calculations, all quantities are specified in quantity per year, not quantity per period. Therefore, once a model is set up, the time structure can be changed without having to change the model inputs.

5. Description of Node Types

This section describes the behavior of the different nodes, the types of links with other nodes that are either allowed or required, and the node parameters. Section 6 describes the input formats and legal values of the parameters. This section describes each of the node types in general terms. In a few cases some equations are provided. A full set of equations will be made available in a separate publication at a later date.

Top Node

There is only one Top Node in a network. It controls the iterations. Each time the network is to be cycled (e.g. an up pass and a down pass is executed) a message is sent to the Top Node and it sends a message to each of the End-Use nodes instructing them to determine their demands and send them down to the nodes that supply them.

The Top Node can have 50 input links. It must have one input link connecting it to each of the End-use node. The Top Node does not require any parameters.

End-use Node

These model end-user demand using constant elasticity demand functions. The demand function is specified by a price-quantity point on the demand function and an elasticity. The price-quantity point can change from period to period and thus the demand curve can shift over time. The elasticity is constant over all periods.

The parameters of the node are the price-quantity point for the first period and the elasticity. This defines the demand curve for the first period. The user also specifies an annual growth rate for the price-quantity points. At each period the price-quantity points are recomputed using these growth rates and the demand curve is updated. With this approach, the demand curve is shifted each period by shifting the price-quantity point that defines the demand curve. At each period the demand is computed for the middle year of the period, using the number of years since the first year of the first period. The quantity demanded each period is computed from the equaton:

$$q(p) = \frac{q_b}{p_b^{\xi}} \bullet p^{\xi}$$

where:

gb,pb = *the price quantity point for the period,*

ξ = *the elasticity.*

The End-Use Node has only one output link to the Top Node and only one input link from its supplier.

The parameter of the End-use node are:

1. Start Quantity: the quantity component of the price quantity point on the demand curve for the first period.
2. Demand Growth: the annual rate of growth of the quantity component of the price-quantity point for future periods, as a fraction.
3. Start Price: the price component of the price quantity point on the demand curve for the first period.

4. Price Growth: the annual rate of growth of the price component of the price-quantity point for future periods, as a fraction.

5. Elasticity: the elasticity of the demand function $[(\Delta q/q)/(\Delta p/p)]$. This is defined so that it is a negative number.

Market Node

The Market Node represents markets where different suppliers compete for market share and where prices are formed based on the prices from all of the suppliers to the market. Market nodes in general have a component called an "allocator" which does the actual market allocation and calculation of market prices. The Market node described here has just one allocator.

The Market node is generally used to allocate a demand (which may be the sum of demands from several nodes) among several different supplier nodes. In that situation it will have one or more output links and more than one input link. However, at other points in a network, it may be necessary to simply sum up a set of demands and send them on down. In that case a market node can be used with several output links and one input link.

On the down pass the Market node allocates demand to each of the suppliers to the market based on their constraint prices. The basic node uses a logistic function to calculate the shares. This form of market model is derived from the assumption that each purchaser in the market always buys from the supplier that has the lowest *effective* cost to that purchaser. However, the effective cost to a purchaser is not necessarily equal to the price charged by the supplier. In fact there will be a distribution of effective prices seen by purchasers due to variations in other costs such as transportation costs, site characteristics, or other local factors. Consequently, the supplier with the lowest price will not necessarily have the lowest effective price for all of the buyers and will not capture all of the market. other suppliers with somewhat higher prices may have lower effective prices for a fraction of the customers. Boyd *et al* show that these assumptions lead to a market share function with the following form:

$$\text{Share of } i^{th} \text{ supplier} = \frac{P_i^{-\gamma}}{\sum_{j=1}^{n} P_j^{-\gamma}}$$

where:

P_i = the price of the ith supplier,

γ = the market's "price sensitivity"

n = the number of suppliers to the market.

Figure 7 illustrates the behavior of this function. The market share of the ith supplier approaches 1.0 as its price becomes very small relative to the other suppliers, and approaches 0.0 as its price becomes very large. The value of γ controls the rate at which market share changes with price.

On the up pass the market node computes the market price as a quantity weighted average price of all the suppliers. Both the average transaction price and the average constraint price are calculated and sent up.

On the down pass, the market nodes can relax the quantity signals that are sent down. Relaxation is discussed in more detail in Section 7. When a quantity is "relaxed" the node computes a required chnage in the quantity between the current iteraton and the last iteration. Then it changes the quantity by a fraction of that amount. Tis smaller change is actually sent. Relaxation is used to ensure more stable convergence of the solution. In most cases it is helpful for the market nodes to relax the quantity signals they send down. However, in some situations, paricualrly when there are constraints on resource nodes, relaxation at the market nodes leads to very slow response to price changes at the resource nodes and can interfere with convergence. In the Control record of the SetUp file the user can specify whether or not the market nodes will relax the quantities.

The market node can have up to 50 output links, and up to 50 input links.

The Market node only requires one parameter, the price sensitivity, γ.

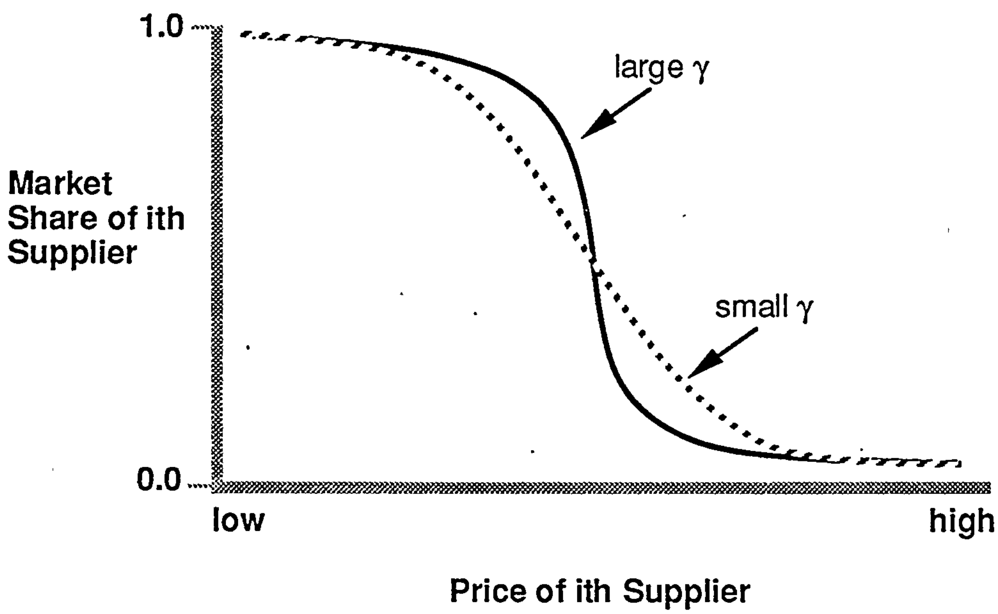


Figure 7: The market share of a supplier depends on its price and the market's price sensitivity, γ

TechBlockl Conversion Node

The TechBlock Conversion node models the conversion of one commodity into another. It is also used to model processes such as transportation processes that do not actually change the commodity, but move it from one location to another at a cost.

The TechBlock Conversion nodes are built around TechBlocks, each of which represents a single block of physical capacity. The conversion nodes included in this version of META•Net uses TechBlocks to represent capacity that is dedicated to serving a demand having a specific load factor (e.g. a load factor of 1.0 for conversion nodes that are not in the electric utility sector, and user specified load factors for conversion nodes in the electric sector). However, TechBlocks have also been used to represent capacity for different vintages in experimental nodes that have been developed (Lamont). The TechBlock1 node has just one TechBlock.

In this version, the inputs are required in fixed proportion to the output quantity. Conversion nodes can use more than one input to produce an output.

The paragraphs below discuss the procedures for handling the quantity and price signals.

When a quantity signal is received, the node passes the quantities requested for each TechBlock to the TechBlocks. Each TechBlock processes the quantities in the following steps:

Determine capacity additions: Capacity is added at the beginning of each period. Each capacity addition is assumed to last for a integral number of periods [although any value for life can be entered, the model computes the number of periods a capacity addition will last as ROUND(Life/(years per period))]. At the beginning of a period the amount of capacity available is equal to the sum of the capacity additions that have not been retired yet. These are the capacity additions added in the past Life/(years per period) - 1 periods. If this capacity is less than the quantity demanded from the TechBlock, then it adds enough capacity to meet the excess demand. The capacity added is the excess demand divided by (Availability Factor * Load factor). The model assumes that the need for capacity additions was anticipated in keeping with the perfect foresight assumptions.

In the first few periods, the existing capacity will have been added in the periods prior to the first period of the model run. It is assumed that capacity was added in equal parts in the previous Life/(years per period) periods. Thus in the first period of the model run, a fraction of the capacity will be ready to retire.

Compute quantity of each input required: The inputs required from each supplier are calculated as linear functions of the quantity demanded using Input-Output coefficients entered for the input links. In this version of META•Net all of the links supply inputs that are required in fixed proportion the output. (Experimental nodes have been built that have more complex calculations of the inputs required.) Note that there is no requirement that the input quantities be positive. By using negative Input-Output coefficients, one can model a negative demand, which corresponds to a by-product or a subsidy. Later examples will show ways to use this technique. Once the quantities needed are determined, the node assembles a quantity signal for each of its supplier nodes and sends them down.

On the Up-Pass, the node receives a price signal and computes the price required for its output. Several steps are needed to calculate prices. The price calculation takes into account future prices to be received, the utilization of existing capacity, and any constraints on the node. In the nodes in this version of META•Net, it is assumed that whenever demand is high enough to require the construction of new capacity, producers demand a price that allows them to cover the net present value of the cost of building and operating one unit of capacity. (A unit of capacity is the amount capacity needed to produce one unit of output per year.) In keeping with the assumption of perfect foresight, the basic approach is as follows: the future prices are known to the producer, thus the producer simply calculates the price to be charged this period in order to make the net revenues equal to the capital cost of a unit of capacity.

The prices are calculated in the following steps:

Determine the future revenue stream for a unit of capacity: For the early periods in the model, this is obtained by simply looking ahead the number of periods corresponding to the life of the new unit of capacity and retrieving the prices that will be charged. This information is available from the previous iteration of the model. In the later periods, the unit of capacity will last beyond the horizon year of the model and an estimate of the future prices must be made. This horizon price is assumed to be constant for all future periods. It is equal to the capital recovery cost of the unit of capacity, plus the costs of its inputs , where the cost of the inputs beyond the horizon are assumed to be equal to their cost in the last period.

Determine the future cost stream for operating a unit of capacity: This is obtained in the same way by simply retrieving the costs from the future periods. For periods beyond the model horizon, the last period prices are used. The AncillaryOperatingCosts (the other costs per unit output that are not included in the costs of the inputs) are added to the costs of inputs.

Determine the costs for the current period: These are passed up with the price signal. AncillaryOperatingCosts are added to them.

Calculate the price required to make the net discounted revenue equal to the capital investment: If the capacity added were operated 24 hours a day, 365 days a year, the discounted net revenues would be equal to the discounted prices, minus the future discounted unit operating costs for the future periods, plus the price minus the operating cost for the current period. However, this calculation based on prices does not take into account that the facility is not always operating. The fraction of the time it actually operates is equal to the availability factor times the load factor. Therefore, the net revenues must be multiplied by the load factor and the availabilityfactor to get the actual income that can be expected from the capacity addition. The price for the current period must be set so that this total revenue is equal to the capital investment needed add a unit of capacity. This capital cost is equal to the specific capital cost. Without considering taxes, the equation is:

$$(Lf * Af)*[(Pc - Cc)* DCF + Rf] = SCC$$

where:

Pc	=	current period price,
DCF	=	discounted cash flow factor, discounts a stream of revenues or costs over one period back to the beginning of the period,
Rf	=	future net revenues (i.e. after the current period), discounted back to the beginning of the current period,
Cc	=	costs per unit output during the current period (input costs plus ancillary operating costs)
SCC	=	specific capital costs (defined below),
Lf	=	the load factor at which the facilities are operated (e.g. the fraction of the time that the facility is called upon to produce output),
Af	=	the availabilityfactor of the facilities (e.g the fraction of the time that the facility is available for operation).

This equation is then solved for the current period price, Pc. Note that all the future prices, operating costs, and current operating costs are known.

The version of the model also takes income taxes into account in a simple way. Simple straight line depreciation is assumed over the life of the investment. This is accounted for in the stream of net revenues, the estimate of revenues beyond the model horizon, and the current period price calculation. Future version of the model could take into account more accurate depreciation calculations.

Adjust price if demand is less than capacity: The price calculation above assumes that a new increment of capacity will be needed. If the demand is less than capacity (i.e. the installed capacity times the availability factor) then the price would be less. To model this, the user specifies a MinCapacityFactor for the node. If the demand is less than (capacity * Load

Factor * Availability Factor * MinCapacityFactor), the price is just equal to the operating cost (costs of inputs plus AncillaryOperatingCost). If demand is greater than installed capacity times the AvailabilityFactor, then the price is the price calculated above. If demand is in between, the price is linearly interpolated between the full cost and the operating cost. For conversion nodes that are not in the electric utiolity sector the Load Factor is 1.0. The conversion nodes in the electric utility sector are modelled using the TechBlock3 conversion node described below. The TechBlock3 conversion node receives a load factor from the Accumulator Node.

Adjust price for constraints: In the last step, price and quantity constraints are checked. There can only be one type of constraint on a TechBlock in a run. If there is a quantity constraint, and the quantity demanded exceeds this constraint, then the transaction and the price are increased. The increase is calculated from a penalty function described in Section 4. At each iteration the price is adjusted until the quantity demanded is equal to, or less than, the quantity constraint. Of course, if the quantity falls below the constraint value, the prices calculated above are used.

If there is a price constraint and the price calculated above exceeds the constraint, then the constraint price alone is raised to reduce the demand to the node. If the input prices to the node are price sensitive, reducing the demand can reduce the costs to the point where the price that the node must charge is less than the constraint. However, it can happen that even when the demand is driven to zero, the price will still be greater than the constraint. In this case the final price will be greater than the constraint, but the total quantity demanded will be zero, implying that the node is not operating.

The maximum number of inputs is 50 in the current version. However, it can only produce one output commodity and can only have one output link. If the model requires that the output go to more than one customer node, the single output is connected through a market node which has one input link and multiple output links.

The TechBlock1 Conversion node's parameters are:

1. First Year of Availability: The first calendar year that the technology is available. It will not produce output until the first period that includes that year.
2. Specific Capital Cost: the cost required to build enough capacity to produce one unit of output per year at a capacity factor and load factor of 1.0 (i.e., if quantities are measure in mmBtus, then it would be the costs needed to build enough capacity to deliver 1 mmBtus per year.)
3. Ancillary Operating Cost: The additional operating cost required in addition to the costs of inputs (supplied by other nodes) per unit of output.
4. Life: The number of years that a typical facility will last before it must be retired.
5. AvailabilityFactor: The fraction of time that a typical facility is available for production. To achieve an actual annual output of X, X/(availability factor * load factor) units of capacity must actually be constructed.
6. Minimum Capacity Factor: The minimum capacity factor at which the facility will include any capital costs.

In addition to these parameters, input-output coefficients and the installed capacity at the start of the first period must also be provided (these are described in the section describing the input files).

Resource Node

The Resource node computes the price required for a resource as a function of the quantity demanded. On the down pass it receives a quantity signal, determines the price it requires, and then sends the price signal up.

This type of node can model any depletable resource such as a mined resource or a resource extracted from a well. In either case the general approach is the same. The model assumes that the resource will be developed in blocks, that is a certain amount will be developed, or committed, each period (the size of the commitment depends on the demand that period). A committed block corresponds to a physical block of resource that is depleted over time. For each block that is committed there is a minimum price that must be recovered by the resource owner in order to develop it and bring it to market. It is assumed that blocks are always committed in order of increasing price.

The rate at which a block can produce is a function of its size. At each period, the rate at which the existing blocks (which have not been exhausted) can produce is calculated. If the demand to the resource exceeds that rate, then a new block that is large enough to meet the excess demand is committed. The price from the resource node is determined by the price required to develop this new block (this price may be adjusted to take into account the resource owner's rent as is described later).

The relationship between the size of block and the rate of production is determined differently for mines and wells. For a mine, it is assumed that once a block is committed, it will produce at a constant rate until it is exhausted. It is further assumed that all mines represented by the node have the same life. Thus, for a mine the production from a block is equal to the size of the block divided by its life. Conversely, the size of a block needed to meet a given demand is equal to the annual demand times the life.

In the case of a well, it is assumed that the production from a block will be equal to a fraction of the total reserves in the block that remain. This is described by the "production to reserves ratio". Thus, for a well the size of a block needed to meet a demand is equal to the demand divided by the production to reserves ratio. Conversely, the annual production that can be obtained from a block is equal to the size of the block times the production to reserves ratio.

To compute the price the node first calculates the total production that can be obtained from the existing committed blocks that have not been exhausted yet. If additional commitments are required it calculates the size required. The price to be charged is then computed from the total amount of additional commitments that have been made since the start of the first period, including the commitment to be made in the current period.

The required price is modelled as an exponential function of the commitments. This called the "price-committments curve". The user specifies it by specifying three points on the curve. The first point is the price required if no additional commitments are required beyond what is committed at the start of the first period of the model. This is simply specified as Price0 since it implies that additional commitments are zero. The second and third points are specified as a price and a level of commitments. These are (Price1, Commitments1) and (Price2, Commitments2). When the Resource node is initialized, it fits an exponential curve through these three points of the form

$$\text{price} = A + B * \text{commitments}^{C}.$$

The price vs. commitments curve specifies the minimum price that a resource owner must recover in order to commit an additional block of resource. However, the owner also has the option to not develop the resource and wait until the price rises. This would be advantageous if the discounted price from one of the future periods is greater than the current price. Since META•Net assumes perfect foresight, the resource node can look ahead to all future prices (computed in the previous iteration) and discount them back to the current period. If the maximum of these discounted prices is greater than the required price computed from the price vs. additional commitments curve, the node will demand the greater price. This additional price is known as "Hotelling rent". Note that the discount rate used by the node is an input. If a very high discount rate is used, the node will compute price from the price-committments curve without any rent.

The Resource node has one output link and no input links. Its parameters are:

1. Price 0: the price required at the current level of commitments.
2. Price 1: The price required for the first point on the price-commitments curve.
3. Commitments 1: the amount of commitments for the first point on the price-commitments curve.
4. Price 2: The price required for the second point on the price-commitments curve.
5. Commitments 2: the amount of commitments for the second point on the price-commitments curve.
6. Resource Type: a flag that indicates whether the resource is a mine or a well.
7. Initial Production: the annual quantity produced from the current commitments.
8. Production to Reserves Ratio: the production to reserves ratio. The value used is only significant if the resource is a well.
9. Resource Life: the life of a typical mine. The value entered is significant only if the resource is a mine.
10. Discount Rate: the annual discount rate used by the resource owner to evaluate discounted future revenues, entered as a fraction.

Resource Price Node

The Resource Price node computes its price as a function over time and is independent of the demand made on it. It is used to model the prices of resources that are relatively insensitive to the demands made on them. An example would be the world oil market in a model of the US energy system. To a first approximation, the world oil price is independent of US demands.

The Resource node has one output link and no input links. Its parameters are the price in the first period and the annual percentage increase.

Load Factor Accumulator Node

The Load Factor Accumulator node provides a simple model of the load duration curve for the electrical sector (the method used for modelling the electrical sector is described in Section 4). On the down pass it receives a quantity of energy demanded from end-users or conversion processes that use electricity. These quantity signals do not contain load factor information, just total quantities of energy. The Load Factor Accumulator then divides this energy into three blocks: peak, intermediate and base load energy. The fraction of the energy in each block is specified by the user along with the load factor for each block.

On the up pass, the Load Factor Accumulator computes the average price of electricity over the three load factors and passes that up to the nodes that it supplies.

The Load Factor Accumulator can supply up to 50 conversion or end use nodes. It receives its input from one Multiple Load Factor Market Node.

The Load Factor Accumulator's parameters are:

1. Fraction 1: the fraction of energy to be assigned to the first load factor category.
2. Load Factor 1: the load factor for the first load factor category.
3. Fraction 2: the fraction of energy to be assigned to the second load factor category.
4. Load Factor 2: the load factor for the second load factor category.
5. Fraction 3: the fraction of energy to be assigned to the third load factor category.
6. Load Factor 3: the load factor for the third load factor category.

The Load Factor Accumulator makes no assumptions about which category is peak, intermediate or base load. That is simply determined by the load factors assigned to the categories.

Multiple Load Factor Market Node

The Multiple Load Factor Market node is used to model the electric utility sector (described in Section 4). It is similar to the ordinary Market node except that it is always used as the supplier to a Load Factor Accumulator Node. Consequently, the quantity signal it receives specifies three different quantities to allocate, and the load factors at which they are to be generated. It has three allocators to allocate the quantities. Each allocator uses the appropriate constraint price data from the price signals sent by the market suppliers and does its allocation as is described for the ordinary market node.

The input nodes for the Multiple Load Factor Market are always TechBlock3 Conversion nodes (described below). These nodes compute prices at three different load factors. When the TechBlock3 Conversion node sends up a price signal, it contains a price for each load factor. Each allocator is sent one price from each of the input nodes (e.g. one allocator gets all the prices for the peak load factor, one gets all the prices at the intermediate load factor, and one gets the prices for the base load factor). On the down pass, each allocator uses its respective set of prices to make its allocation between suppliers.

The number of permissible links and the parameters are the same as those of the ordinary market node.

TechBlock3 Conversion Node

The TechBlock3 Conversion Node is used for modeling the electric sector. Section 4 described the method for modeling the electric sector in detail. However, for the purpose of describing the TechBlock3 Conversion Node, the procedures can be summarized as follows: In the electric sector, energy is demanded at different load factors. META•Net approximates the electric demand by breaking the electric energy demanded into three portions, each one demanded at a different load factor. The energy is divided into the three portions by the Load Factor Accumulator Node using user specified fractions and load factors. On the down pass, the quantity signal sent the TechBlock3 node contains three separate quantities demanded, each quantity at a different load factor (the load factor is included in the quantity signal).

The TechBlock3 Node contains three separate TechBlocks, each one is dedicated to serving a portion of demand at a different load factor. Price and quantity signals are processed in the same way as the TechBlock1 node. However, on the down pass, the TechBlock3 node sends one of the quantities demanded to each of the TechBlocks along with the load factor. Each tech block then uses the procedures described above to determine the quantities of its input that is required.

On the up pass, the node receives a price signal from each of its input nodes. This price information is passed to each of the TechBlocks which calculate their required prices just as is described for the TechBlock Conversion node. In calculating their prices, each TechBlock uses the load factors that were sent down with the quantity signal. Due to the different load factors, each TechBlock will compute a different price. These prices are sent up to the next node (which is a Multiple Load Factor Market).

The parameters and links for the TechBlock3 node are the same as for the TechBlock1 node.

6. Input Files Required to Define a Model

This section describes the files required to run META•Net. These files contain the general parameters for controlling the run, the structure of the nodes and links, and the parameters for each of the nodes in the model. This section only describes the structure of the files and the legal values of the parameters. The detailed descriptions of the parameters required for each of the nodes was discussed above under the descriptions of each node type.

META•Net reads its data from a standard set of comma delimited ASCII files. Data in the files is stored in records with a standard header format. Results from the run are written to output files having the same header format. The formats are designed so that a relational data base manager can be used to assemble input files and to display output results in any way that the user requires. This approach makes it easier to separate the data management from the structure of META•Net and gives the maximum flexibility to design the input and output processing in a way that is best suited to the user's needs. Currently Microsoft Excel is used as a simple data manager.

META•Net always requires at least four files. If constraints are included in the model, one additional file is needed for the constraint information. These files are:

1. SetUp File which contains data about the models periods and time horizon, default economic variables, and diagnostics control.
2. Node File which contains a list of the nodes and some identifying information about them,
3. Links File which contains a list of the links and identifying information about them,
4. Components File which contains the parameters of all the nodes and their components, and
5. Constraints File which contains data abut price and quantity constraints and the parameters of the functions that implement the constraints.

Each node has one record in the Nodes File, one record in the Links File for each input link, and one or more records in the Components File. Conversion nodes may also have a record in the Constraints File.

All of these files must be in the same folder together and they must be named as follows: *ModelName*.SetUp, *ModelName*.Nodes, *ModelName*.Links, *ModelName*.Components, *ModelName*.Constraints. *"ModelName"* is the name of the model chosen by the user.

Detailed description of the parameters and records that must be supplied for each type of node are provided in the tables below. First the contents of the Node File and the Links File are described. The records for these two files contain the same information for all nodes. Then the records and parameters to be entered in the Components File are described for each node type. The files for the Sample model provide an example of the way that the data is entered.

File and Record Structure

The files are text files and can be created with a word processor, by a data base, or from a spreadsheet. Each file contains a series of records that all have a standard header format. The fields of the records are separated by commas and each line ends with an end-of-line marker (carriage return).

To make building and maintaining input files easier, the user can insert comments into the file. Any record that has a "c" in the first field is a comment. The user can enter anything in the rest of the record. If it is important to have the comments align when the file is viewed in a spreadsheet, commas must be entered to separate the fields. Records that contain data have a "d" in the first field.

The first five fields of every record constitute the record header which identifies the node and component that the record refers to and the data set that the record contains. These are referred to below as the "header fields". In some cases a record does not refer to any particular component, or even to a particular node. In that case "na" is entered into the corresponding field. The fields of the record header are:

1. "Comment/Data" marker: this is a "c" or a "d" as is described above,
2. Node name: this is the name of a node for which the record contains data.
3. Component type: the type of component for which the record contains data.
4. Component Identifier: in some cases the components are numbered, in other cases they have names.
5. Data Set: this identifies the type of data contained in the record.

The remaining fields of the record contain the parameters for the particular node or component. These are referred as the "parameter fields".

Procedure for Loading Data From the Files

In order to better understand the structure of the data required to run META•Net and the types of errors that can occur, this section describes the way that META•Net loads its data. First, the user selects the SetUp file to load a model (loading a model is described below). From this META•Net can determine the names of all the other files required, since the are in the format of "*ModelName.DataFile*". It then proceeds to open all of the other files needed. If it cannot find a constraints file, it assumes that there is none. Thus the constraints file is not necessary. As it is opening files, it gives the user messages about which files were found.

Next, META•Net reads the data in the SetUp file and reads the list of nodes in the Nodes file. It gives the user a message when this is done. Then META•Net reads the links in the model. For each node, it searches the Links file looking for all records that contain link data for that node. When it finds a record it reads the data about the link. When it can find no more records for a node, it goes on to look for the link data for the next node. Finally, META•Net initializes each of the nodes by setting up any internal data structures and reading in data from the Components File and the Constraints File (if there is one). Each node needs at least one data record for the parameters for the node as a whole and some nodes need additional records for each of its components. To find the records for a node, it forms a header for each record that it needs and searches the Components File until it finds a record with a matching record or determines that none is there.

If META•Net cannot find a record that it is looking for, it gives and error message that it cannot find the record. It will continue to search for other records in order to find as many errors as possible. However, the model will not run unless it can find all the records needed.

Since META•Net keeps searching until it finds a record or determines that it is not there, the order of the records is not critical. But, when data is being loaded, the procedure always starts searching for the next record at the point in the file where it found the last record (if it does not find a record it loops back to the beginning of the file). Therefore, if the records are in the right order, the files load much more quickly. This is especially true in the current version of META•Net since all searches are done on the disk rather than in memory. The sections below describe the best order for the records.

Data Entered in the SetUp File

Five records are required in the SetUp file. These provide information for the model as a whole and some default values that can be used by nodes. The records are:

1. General data: provides a name for the run,
2. Period data: specifies the structure of the periods in the run,
3. Control data: controls relaxation,
4. Economic data: specifies default values for economic variables, and
5. Diagnostic parameters: control the diagnostics in the model.

The fields of these records are described below.

General data record:

	Head. Fields					Param. Field
Field #	*1*	*2*	*3*	*4*	*5*	*6*
Field Name	*Cmnt/ Data*	*Node*	*Comp. Type*	*Comp. Id*	*Data type*	*Run Name*
Legal Values	"d"	"na"	"na"	"na"	"General"	Name of user's choosing
Example	d	na	na	na	General	Test run #4

The "Run Name" is a name to identify the run, chosen by the user. This will be output in the NodeResults file. It does not have to be the same name as was used for the "ModelName" in the file names.

Periods data record:

	Head. Fields					Param. Fields		
Field #	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Field Name	*Cmnt/ Data*	*Node*	*Comp. Type*	*Comp. Id*	*Data type*	*Number of Periods*	*Years per Period*	*Starting Year*
Legal Values	"d"	"na"	"na"	"na"	"Periods"	integer , between 1 and 10	integer, equal or greater than 1	a calender year
Example	d	na	na	na	Periods	10	5	1992

Note that the maximum number of periods allowed in version 1.2 is ten.

Control data record:

	Head. Fields					Param. Field	
Field #	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Field Name	*Cmnt/ Data*	*Node*	*Comp. Type*	*Comp. Id*	*Data type*	*Relaxation Coefficient*	*Relax Market Nodes Flag*
Legal Values	"d"	"na"	"na"	"na"	"Control"	Real Value between 0.0 and 1.0	"true" or "false"
Example	d	na	na	na	Control	0.5	true

The Relaxation Coefficient and strategies for setting it are discussed below. The "Relax Market Nodes Flag" tells the market nodes whether or not to relax the quantity signals when they are sent down to the supplier nodes.

Economic data record:

	Head. Fields					Param. Fields	
Field #	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Field Name	*Cmnt/ Data*	*Node*	*Comp. Type*	*Comp. Id*	*Data type*	*Interest Rate*	*Income Tax Rate*
Legal Values	"d"	"na"	"na"	"na"	"Economic"	real value, =>0.0	real value, => 0.0
Example	d	na	na	na	Economic	0.10	0.45

The interest rate and income tax rate are used by most conversion and resource nodes for calculating rate of return. Both rates are expressed as decimal numbers, not as percentages.

Diagnostic data record:

	Head. Fields					Param. Fields	
Field #	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Field Name	*Cmnt/Data*	*Node*	*Comp. Type*	*Comp. Id*	*Data type*	*Trace On*	*Min. Converge. Amount*
Legal Values	"d"	"na"	"na"	"na"	"Diagnostic"	"TRUE" or "FALSE"	real value
Example	d	na	na	na	Diagnostic	FALSE	0.45

If Trace On is TRUE, the model will print of a record of every price and quantity signal sent during the first cycle of the network. This can help identify errors in the network. The

Minimum Convergence Amount is the smallest change in amount that the Convergence Monitor will record. If a convergence signal is sent with a change in quantity smaller than this, the Convergence Monitor will ignore it. This helps filter out nuisance signals from nodes that are receiving very small quantities which happen to be large percentage changes. If all of the signals sent to the Convergence Monitor in an iteration have changes smaller than the Minimum Convergence Amount, it will print out "null" for the node name.

Data Entered in the Nodes File

Each record contains a header that identifies the record as containing parameters for a node and information that identifies the node. Each record has the following structure:

	Header Fields				
Field #	*1*	*2*	*3*	*4*	*5*
Field Name	*Cmnt/ Data*	*Node*	*Comp. Type*	*Comp. Id*	*Data type*
Legal Values	"d"	User defined name, up to 30 chars.	"na"	"na"	"NodeID"
Example	d	Fuel Mkt	na	na	NodeID

	Param. Fields			
Field #	*6*	*7*	*8*	*9*
Field Name	*Node Type*	*Region*	*Sector*	*Note*
Legal Values	One of designators listed in Table 1	User defined name, up to 30 chars.	User defined name, up to 30 chars.	User defined, up to 30 chars.
Example	Market	West	Petrol	Study1

The NodeType identifies the type of node to use from the library of nodes. Table 1 shows the node types that can be used. The values for Region, Sector, and Note are not used by META•Net itself. They are defined so that a user can then use a relational data base to extract or insert data from all nodes in a given Region or Sector, or having a given Note.

Node Type	Designator
Top	Top
End-use	EndUse
Load Factor Accumulator	SimpLFAccum
Market	Market
Multiple Load Factor Market	MultLFMarket
TechBlock1 Conversion	TB1Conv
TechBlock3 Conversion	TBLFConv
Resource	Resource
Resource Price	ResourceP

Table 1: Designators for each node type.

Data Entered in the Links File

Each link in the model has a record of the following format in the Links File:

	Header Fields				
Field #	*1*	*2*	*3*	*4*	*5*
Field Name	*Cmnt/ Data*	*Node*	*Comp.Type*	*Comp. Id*	*Data type*
Legal Values	"d"	Name of one of the nodes in the model	"InLink"	integer for the sequential number of the link	"LinkData"
Example	d	Fuel Mrkt	InLink	1	LinkData

	Param. Fields			
Field #	*6*	*7*	*8*	*9*
Field Name	*DownNode*	*Net Type*	*Customer Type*	*Input Type*
Legal Values	Name of one of the nodes in the model	"normal" or "loop" (see discussion)	"all" (see discussion)	"operating" (see discussion)
Example	Fuel Mkt	normal	all	operating

In the Links File, the input links for a node are numbered sequentially and this number is entered into the Component Id field to identify each link (links must also be identified in the Components File, but in that file they are identified by the name of the node at the input end).

As is discussed in the section on structuring models, some networks are cyclic and require "loops" within the network. These are identified to the model by labeling one link in the loop as a "loop link" in the Net Type data field. All other links are labelled "normal" in the Net Type field.

The Customer Type field refers to the type of pricing that the customer of a node receives. This feature is not used in any of the nodes included in this version of META•Net. It has been used for experimental nodes that allow for price discrimination between customers. For the nodes in this version, "all" is entered in the Customer Type field indicating that there is no price discrimination.

The Input Type field indicates the type of input tat this link carries to the node having this link as an input. For all the nodes in this version of META•Net all of the links carry inputs that go into the operating cost equation of the node. Thus "operating" is entered in this filed for all nodes (even nodes that are not conversion nodes). Some experimental nodes have been built with links that carry inputs such as emissions permits (Lamont 92). The costs of these inputs and the calculation of demands for them are handled differently in the node that are simple operating inputs.

The model loading process will be faster if the sets of links for the nodes are entered in the file in the same order as the nodes are listed in the Nodes File, and if the links are entered in numerical order.

Data Entered in the Components and Constraints Files

The Components File contains the parameters for each of the nodes and for their components. Each node requires one record for data that pertains to the node as a whole. The components of the node may also require one or more records for their data. Finally, the conversion nodes may have records that specify constraints. Records specifying the constraints are entered in the Constraints File.

Again, the loading process will be faster if the sets of records for the nodes are entered into the Components file and the Constraints File in the same order as the nodes are entered in the Nodes File.

The following sections describe the records required for each type of node.

Top Node

The Top Node does not require a record in the Components File.

End-use Node

The End-Use Node requires one record in the Components File.

	Header Fields				
Field #	*1*	*2*	*3*	*4*	*5*
Field Name	*Cmnt/ Data*	*Node*	*Comp.Type*	*Comp. Id*	*Data type*
Legal Values	"d"	User defined name, up to 30 chars.	"na"	"na"	"NodeData"
Example	d	Trans.Dem	na	na	NodeData

	Param. Fields				
Field #	*6*	*7*	*8*	*9*	*10*
Field Name	*Start Quantity*	*Demand Growth*	*Start Price*	*Price Growth*	*Elasticity*
Legal Values	Real, greater than or equal to 0.0	Real	Real, greater than or equal to 0.0	Real	Real, less than or equal to 0.0
Example	172.3	0.03	5.7	0.00	-0.5

Market Node

A Market Node requires one record In the Components File.

	Header Fields					**Param. Field**
Field #	*1*	*2*	*3*	*4*	*5*	*6*
Field Name	*Cmnt/ Data*	*Node*	*Comp. Type*	*Comp. Id*	*Data type*	*Price Sensitivity*
Legal Values	"d"	Must match name in Nodes File	"na"	"na"	"NodeData"	Real, =>0.0
Example	d	Fuel Mkt	na	na	NodeData	5.0

Multiple Load Factor Market Node

A Multiple Load Factor Market Node also requires one record in the Components File.

	Header Fields					Param. Field
Field #	*1*	*2*	*3*	*4*	*5*	*6*
Field Name	*Cmnt/ Data*	*Node*	*Comp. Type*	*Comp. Id*	*Data type*	*Price Sensitivity*
Legal Values	"d"	Must match name in Nodes File	"na"	"na"	"NodeData"	Real,=>0.0
Example	d	Elect.Mkt	na	na	NodeData	5.0

TechBlock1 Conversion Node

The TechBlock Conversion Node requires one record of parameters for the node as a whole, one record for its TechBlock, one record for each input link, and one record if the TechBlock has a price or quantity constraint.

The parameter record for the node is:

	Header Fields				
Field #	*1*	*2*	*3*	*4*	*5*
Field Name	*Cmnt/ Data*	*Node*	*Comp.Type*	*Comp. Id*	*Data type*
Legal Values	"d"	Must match name in Nodes File	"na"	"na"	"NodeData"
Example	d	Petrol.Refin	na	na	NodeData

	Param. Fields					
Field #	*6*	*7*	*8*	*9*	*10*	*11*
Field Name	*First Year of Availability*	*Specific Capital Cost*	*Ancillary Operating Cost*	*Life*	*Availiibility Factor*	*Minimum Capacity Factor*
Legal Values	calendar year	Real, greater than or equal to 0.0	Real	Real, greater than or equal to 0.0	Real, between 0.0 and 1.0	Real,between 0.0 and 1.0, less than Capacity Factor
Example	1960	32.5	0.35	20.0	0.85	0.40

The TechBlock requires a record with the following format:

	Header Fields					Param. Field
Field #	*1*	*2*	*3*	*4*	*5*	*6*
Field Name	*Cmnt/ Data*	*Node*	*Comp.Type*	*Comp. Id*	*Data type*	*Starting Capacity*
Legal Values	"d"	Must match name in Nodes File	"TechBlock"	Integer=1	"TBlockData"	Real,>= 0.0
Example	d	Coal.Boiler	TBlock	1	TBlockData	38.0

Each input link for the node requires a record with the following format:

	Header Fields					Param. Field
Field #	*1*	*2*	*3*	*4*	*5*	*6*
Field Name	*Cmnt/ Data*	*Node*	*Comp. Type*	*Comp. Id*	*Data type*	*Input-Output Coefficient*
Legal Values	"d"	Must match name in Nodes File	"InLink"	Name of node at other end of the link	"InputLinkData"	Real
Example	d	Coal.Boiler	InLink	Coal.Mkt	InputLinkData	1.3

Since META•Net searches the input files for the records it needs, the loading process can be speeded up if the Conversion Nodes records are in the following order: node parameters, Tech Block data, and link data, with the link data records in the same order as they were listed in the Links File.

If the TechBlock has a constraint, the following record is entered in the Constraints File:

	Header Fields				
Field #	*1*	*2*	*3*	*4*	*5*
Field Name	*Cmnt/ Data*	*Node*	*Comp.Type*	Integer, ranging from 1 to 3	*Data type*
Legal Values	"d"	Must match name in Nodes File	"TechBlock"	1	"PriceConstraint" or "QuantityConstraint"
Example	d	Coal.Boiler	TechBlock	1	QuantityConstraint

	Param. Fields					
Field #	*6*	*7*	*8*	*9*		*Last Period*
Field Name	*Constraint Sensitivity A*	*Constraint Sensitivity B*	*Maximum Constraint Ratio*	*Constraint Value for period 1*	• • •	*Constraint Value for last period*
Legal Values	Real, > 0.0	Real, >0.0	Real, > 0.0	Real, >0.0		Real, >0.0
Example	1.0	3.0	2.0	50.0		100.0

TechBlock3 Conversion Node

The TechBlock3 Conversion Node requires one record of parameters for the node as a whole, one record for each TechBlock, one record for each input link, and one record for each TechBlock that has a price or quantity constraint.

The parameter record for the node is:

	Header Fields				
Field #	*1*	*2*	*3*	*4*	*5*
Field Name	*Cmnt/ Data*	*Node*	*Comp.Type*	*Comp. Id*	*Data type*
Legal Values	"d"	Must match name in Nodes File	"na"	"na"	"NodeData"
Example	d	Coal.Boiler	na	na	NodeData

	Param. Fields					
Field #	*6*	*7*	*8*	*9*	*10*	*11*
Field Name	*First Year Available*	*Specific Capital Cost*	*Ancillary Operating Cost*	*Life*	*Availability Factor*	*Minimum Capacity Factor*
Legal Values	Integer year	Real, => 0.0	Real	Real, => 0.0	Real, => 0.0, =<1.0	Real,=> 0.0, =<1.0, < Capacity Factor
Example	1960	42.5	0.35	40.0	0.67	0.40

Each TechBlock requires a record with the following format:

	Header Fields					Param. Field
Field #	*1*	*2*	*3*	*4*	*5*	*6*
Field Name	*Cmnt/ Data*	*Node*	*Comp.Type*	*Comp. Id*	*Data type*	*Starting Capacity*
Legal Values	"d"	Must match name in Nodes File	"TechBlock"	Integer, ranging from 1 to 3	"TBlockData"	Real,>= 0.0
Example	d	Coal.Boiler	TBlock	1	TBlockData	38.0

Each input link for the node requires a record with the following format (note that only one record is entered for each link to a node. All the Tech Blocks use the same set of links):

	Header Fields					Param. Field
Field #	*1*	*2*	*3*	*4*	*5*	*6*
Field Name	*Cmnt/ Data*	*Node*	*Comp. Type*	*Comp. Id*	*Data type*	*Input-Output Coefficient*
Legal Values	"d"	Must match name in Nodes File	"InLink"	Name of node at other end of the link	"InputLinkData"	Real
Example	d	Coal.Boiler	InLink	Coal.Mkt	InputLinkData	1.3

For each TechBlock that has a constraint, the following record is entered into the Constraints File (it is not necessary that all Tech Blocks in a node have a constraint):

	Header Fields				
Field #	*1*	*2*	*3*	*4*	*5*
Field Name	*Cmnt/ Data*	*Node*	*Comp.Type*	*Comp. Id*	*Data type*
Legal Values	"d"	Must match name in Nodes File	"TechBlock"	Integer, ranging from 1 to 3 *	"PriceConstraint" or "QuantityConstraint"
Example	d	Coal.Boiler	TechBlock	1	QuantityConstraint

	Param. Fields					
Field #	*6*	*7*	*8*	*9*		*Last Period*
Field Name	*Constraint Sensitivity A*	*Constraint Sensitivity B*	*Maximum Constraint Ratio*	*Constraint Value for period 1*	• • •	*Constraint Value for last period*
Legal Values	Real, > 0.0	Real, >0.0	Real, > 0.0	Real, >0.0		Real, >0.0
Example	1.0	3.0	2.0	50.0	• • •	100.0

* the value of the Comp. Id. must correspond to the number of the Tech Block

A complete example of the records entered into the Components File for the TechBlock3 node is:

	Header Fields				
Field #	*1*	*2*	*3*	*4*	*5*
Record 1	d	Coal.Boiler	na	na	NodeData
Record 2	d	Coal.Boiler	TechBlock	1	TBlockData
Record 3	d	Coal.Boiler	TechBlock	2	TBlockData
Record 4	d	Coal.Boiler	TechBlock	3	TBlockData
Record 5	d	Coal.Boiler	InLink	Coal.Mkt	InputLinkData

	Param. Fields					
Field #	*6*	*7*	*8*	*9*	*10*	*11*
Record 1	1960	42.5	0.35	40.0	0.67	0.40
Record 2	38.0					
Record 3	38.0					
Record 4	38.0					
Record 5	1.3					

Resource Node

The Resource Node requires one record in the Components File:

	Header Fields				
Field #	*1*	*2*	*3*	*4*	*5*
Field Name	*Cmnt/ Data*	*Node*	*Comp.Type*	*Comp. Id*	*Data type*
Legal Values	"d"	Must match name in Nodes File	"na"	"na"	"NodeData"
Example	d	Coal.Resource	na	na	NodeData

	Param. Fields				
Field #	*6*	*7*	*8*	*9*	*10*
Field Name	*Price 0*	*Price 1*	*Commitments 1*	*Price 2*	*Commitments 2*
Legal Values	Real, >= 0.0	Real, >Price0	Real, >= 0.0	Real, > Price1	Real, > Commitments1
Example	42.5	55.0	40.0	127.0	70.0

	Param. Fields contd.				
Field #	*11*	*12*	*13*	*14*	*15*
Field Name	*Resource Type*	*Initial Production*	*Production to Reserves Ratio*	*Resource Life*	*Discount Rate*
Legal Values	"mine" or "well"	Real, >0.0	Real, > 0.0, <1.0	Real, > 0.0	Real,> 0.0
Example	mine	300.0	0.07	20.0	0.12

Recall that if the resource is a *mine*, only the Resource Life parameter is used. If it is a *well*, only the Production to Reserves parameter is used. However, values for both parameters must be entered.

Resource Price Node

The Resource Price node requires one record in the components file:

	Header Fields				
Field #	*1*	*2*	*3*	*4*	*5*
Field Name	*Cmnt/ Data*	*Node*	*Comp.Type*	*Comp. Id*	*Data type*
Legal Values	"d"	Must match name in Nodes File	"na"	"na"	"NodeData"
Example	d	Coal.Resource	na	na	NodeData

	Param. Fields	
Field #	*6*	*7*
Field Name	*Starting Price*	*Price Growth*
Legal Values	Real, > 0.0	Real, => 0.0
Example	12.8	0.07

Accumulator

The Accumulator requires one record in the Components File:

	Header Fields				
Field #	*1*	*2*	*3*	*4*	*5*
Field Name	*Cmnt/ Data*	*Node*	*Comp.Type*	*Comp. Id*	*Data type*
Legal Values	"d"	Must match name in Nodes File	"na"	"na"	"NodeData"
Example	d	Coal.Resource	na	na	NodeData

	Param. Fields					
Field #	*6*	*7*	*8*	*9*	*10*	*11*
Field Name	*Fraction 1*	*Load Factor 1*	*Fraction 2*	*Load Factor 2*	*Fraction 3*	*Load Factor 3*
Legal Values	Real,=>0.0	Real >0.0, <=1.0	Real,=>0.0	Real >0.0, <=1.0	Real,=>0.0	Real >0.0, <=1.0
Example	0.05	0.1	0.35	0.6	0.6	1.0

Note that the three fractions should sum to 1.0. Also, it does not matter which fraction is chosen for which load factor.

7. Initializing and Running a Model

Once the files are defined, the model is ready to run. Very briefly, the following steps are require to run a model: The user prepares the input files, starts META•Net and instructs it to load the model. The model begins making iterations and reporting out the convergence status. Along the way the user can pause META•Net to change the relaxation factor and restart. Once the user is satisfied that convergence is satisfactory, he pauses META•Net and instructs it to quit. Before quitting, all output files are written.

The outputs from the nodes are all written to one file. Each node has one or more records in the file that record information about its variables (e.g. prices, quantities, installed capacities, etc.) Because all of the records have the same header format, the user can load the file into a database or a spreadsheet for analysis.

These steps are described in more detail below.

Starting META•Net, Loading a Model, and Stopping

Before making a run, the SetUp, Nodes, Links, Components, and Constraints files are placed together in one folder (there can be other things in the folder, too). Then META•Net is started by double clicking on its icon (it does not have to be in the same folder with the input files).

When META•Net starts, it will show a window like the one in Figure 8. This is the Control Panel. The scrolling window on the left displays status messages from the software to the user. The window on the right displays convergence information. Two buttons are at the bottom. One instructs META•Net to start running, and the other tells META•Net to pause so that the user can change the relaxation or quit. The small window to the right of the buttons displays the relaxations coefficient. The user can edit this when the model is paused. Toward the top right, there is a small window that shows the number of iterations and another window the shows the name of the model.

The only menu available is the "File" menu. It has two items: "Load Model" and "Quit".

To start a run, the user chooses "Load Model" from the File menu. META•Net displays a standard file selection dialogue box. The user selects the SetUp file for the model to be run and opens it. Note that the files for the model do not have to be in the same folder with META•Net. It is only important that they be together in the same folder.

META•Net will start loading the model's files. It first displays messages about which files are being opened. Then there is a sequence of messages about the loading steps. These are:

- "Reading Nodes": reading all the data i the Nodes file.
- "Reading Links": reading data in the Links file. Each node reads the information about its own input links, including the name of the link at the other end.
- "Initializing nodes": setting up each node's internal structures such as the link managers and any other internal objects it needs.
- "Initializing links": each node sends a message to the the node at the other end of each of its input links instructing that node establish a connection.

- "Checking nodes": each node has a procedure that checks the status of its links and internal parameters. This does not check for everything that can be incorrect, but it will detect a number of logical errors in the structure of the network and the values of the node's parameters.

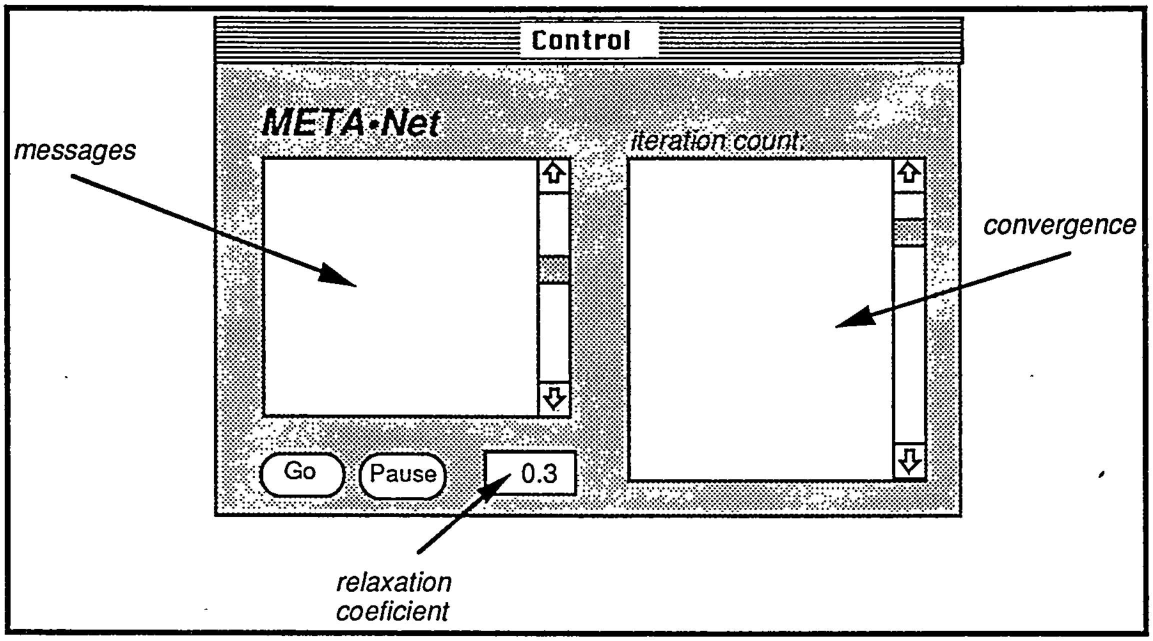


Figure 8: When META•Net starts, it displays the Control Panel.

As each node is processed during these steps a period(".") is written to the screen to indicate the progress.

For a model of any size the loading process can take several minutes. This is basically due to the fact that ß1.0 of META•Net searches the disc file for each record that it needs rather than loading everything into memory. Future versions of the software can probably speed up the operation considerably. When the loading process is over, META•Net sounds a short chord to call the user back from his coffee break to get started with the run.

Occasionally the user makes an error in specifying the model: a link may be left out, or a name may be misspelled. The loading process can detect a number of these. When it detects an error in loading, it will write a message in the message window instructing the user to check the error file. Note that it usually only writes a message for the first error to avoid filling up the window with messages. Since errors indicate that the model is incorrectly specified, the model will not run if an error is detected in loading. The error message will state this. At that point, the user can only choose Quit from the File menu and open the Error File to find out what was wrong. The error handling is discussed in more detail below.

When the model is loaded, the relaxation value window will show the value loaded from the SetUp file, the Pause button will be dimmed, and the Go button will be activated. To start the model running, the user clicks on the Go button. Iterations will begin. After each iteration, a message is written out in the Convergence Window. The section below discusses the format of this information, the way that it is interpreted, and the process of changing the relaxation coefficient to control convergence.

At any time, the iterations can be paused by click on the Pause button. This is done to either change the relaxation coefficient, to review the convergence history that is printed out, or to quit the run. The model will not stop immediately when Pause is clicked. It will finish the current iteration before stopping. When it stops, the Pause button is dimmed and the Go button is activated.

To end the run, the user pauses the model, then selects Quit from the File menu. META•Net writes out all of the output files and then quits.

Controlling Relaxation and Convergence

Relaxation is a critical part of the convergence procedure. During a run the convergence status of the model is displayed. The user can adjust the convergence behavior by adjusting the "relaxation coefficient" during the run. This section describes the relaxation scheme, strategies for managing the relaxation coefficient during a run to speed up the convergence. The section also describes the convergence behavior that models can display and ways to determine when the model has converged.

The Relaxation Coefficient

Each node at each iteration computes a new price required or a new quantity demanded. However, if it simply sent this quantity or price, the model would frequently not converge to a solution: there are many situations when this simple procedure would actually diverge and be unstable. The relaxation scheme adjusts prices and quantities before sending them on. At each iteration a node computes the new price or quantity desired. Then it sends a value that is equal to the value from the previous iteration, plus a fraction of the difference between the previous value and the new computed value. This fraction is the "relaxation coefficient". In META•Net it must be between 0.0 and 1.0. If it is 0.0, the model makes no adjustments from iteration to iteration; If it is 1.0, it adjusts the full amount at each iteration.

Relaxation is not actually used by all nodes. Only the end-use and resource nodes use it all of the time. Conversion nodes do not relax their signals unless they have a constraint imposed on them. In that case, they relax the price signals.

Changing the relaxation during a run is straightforward. First, click the Pause button and wait for the system to stop (the Pause button will dim and the Go button will become active when it stops). Then edit the value of the relaxation coefficient in the relaxation window. The value should always be a decimal fraction between 0.0 and 1.0. Be careful in editing because this version of META•Net does not check the number for validity. Finally, click the Go button and the model will start running again

Monitoring Convergence Behavior

As is described earlier, during an iteration, the end-use and the resource nodes report the relative change in their quantities since the last iteration to the Convergence Monitor. As each report comes in the Convergence Monitor keeps track of the nodes that have had the largest changes. After an iteration is complete (i.e. the net has been cycled for each of the periods), the Convergence Monitor sends a message to the Control Panel. This is displayed in one line in the convergence window listing the name of the node, the period, and the fractional change in quantity.

The convergence behavior of models built in META•Net tend to follow a standard pattern. Initially there is a fairly chaotic set of perhaps 5 to 30 iterations. The changes from iteration to iteration will be positive, then negative. They can be very large. There can also be long sequences (tens of iterations) when the same node at the same period has the same fractional change. This appears to be caused by the situation where a node needs to reduce the quantity demanded to near zero, but the relaxation only allows the quantity to be reduced by a set fraction each period.

After the initial set of iterations, the convergence will tend to settle into a pattern where the quantities increase for several iterations (the fractional change is positive) and then decrease for several iterations. The size of the changes will also oscillate. The changes may be very small for a few iterations, then the magnitude of the change will begin to grow up to a maximum and then gradually reduce back down to a small value. If the model is converging, one will notice that the maximum and the minimum are gradually getting smaller (in magnitude) from oscillation to oscillation. Eventually the magnitude of the maximum becomes small enough that the model is converged for all practical purposes.

Because of this oscillatory behavior, one cannot simply wait until the change between two iterations is small and then quit. Very often some very small change may be observed, even for four or five iterations, and then the change will grow again.

The initial chaotic behavior is due to the way that the initial values of prices and quantities are set. The initial quantities and prices on each link are set to arbitrary values—META•Net creates a set of initial price and quantity signals and assigns them to the link managers. These values are undoubtedly not very close to the final solution. However, in the initial iterations, the relaxation process uses these as the "previous iteration" values and makes adjustments away from them. A large initial relaxation coefficient causes some seemingly chaotic behavior for a number of iterations, but allows the solution to adjust away from these arbitrary initial values more quickly. (Future versions of META•Net are expected to include a "warm start" capability so that a new model run can start from a previous solution. This will reduce the number of iterations needed to get close to the correct solution.)

Managing the Relaxation Coefficient

The initial relaxation coefficient is set in the SetUp file. At any time the user can change this value (even before the run starts, if desired). Faster convergence can be obtained by properly setting the initial relaxation coefficient and by making a few adjustments during he run.

In principle, using a large relaxation coefficient (greater than 0.3) should lead to the fastest convergence. However, this can cause a model to overshoot the solution from iteration to iteration, which leads to slower convergence; or to be outright unstable, which will not lead to convergence at all. A small relaxation coefficient (say, smaller then 0.1) will avoid overshooting and instabilities, but only allows small steps each iteration. The best compromise is different for each model must be found through experimentation. It particularly depends on the constraints that are imposed.

Generally speaking, it seems best to start with a large relaxation coefficient (say, 0.5) and run for a few iterations, then reduce it as much as needed to obtain fairly smooth convergence to the final solution.

The large relaxation coefficient is needed at the beginning to allow the model to adjust away from the initial, arbitrary quantities and prices and find values that are at least the right order of magnitude. Once the initial behavior has smoothed out (after perhaps 20 iterations), the relaxation coefficient can be reduced. In some cases a single reduction is the best strategy. In other cases it appears to be best to reduce the relaxation coefficient several times. This can be particularly true when there are constraints. It will sometimes happen that the model will reach some sort of a limiting cycle: the changes will oscillate between positive and negative, with the same values in each oscillation. Further reducing the relaxation coefficient will usually cause the oscillations to diminish and make the model converge.

It should be noted that the largest change that the model can make from iteration to iteration is controlled by the relaxation coefficient. Consequently, when the relaxation coefficient is reduced, there will immediately be a reduction in the largest change shown in the Convergence Window. But this reduction is entirely due to the reduction in the relaxation coefficient, not the convergence of the model. It is important to observe the patterns of changes in judging whether or not the model has converged, not just the absolute amount of changes.

Output Files

At the end of the run META•Net writes out three files that contain node results, errors messages, and trace information (if the trace was turned on in the SetUp file). These files will be created in the same folder that contains the META•Net program.

The data are written in records that have headers with the same format as the input records. The header fields are: comment/data marker, node name, component type, component identifier, and data set. The header is followed by a series of data fields containing information about the node or error. The files are written in comma delimited text files. Thus they can be read by a spreadsheet or data base to sort and analyze the output.

The node results are the primary output of a run. At the end of the run META•Net instructs each node to write out its internal data to the File named "NodeResults". Each node type writes a different set of information. Most nodes write out the price and quantity information for each of its output links. For conversion nodes, price and quantity data is written for each Tech Block, rather than for the node as a whole. Conversion nodes also write the capacity and capacity additions for each period. Resource nodes write out the commitments for each period. End use nodes write out the price and quantity information for their input links.

Errors are written to the ErrorReport file as they are detected. The section below on error handling describes the information in the error file.

If tracing has been turned on in the SetUp file, the TraceOutput file will contain a record of every signal sent between nodes. The record describes the type of siganl (quantity or price), the sending node, and the receiving node. This can be usefull for checking the continuity of large networks. However, the error checking step in initializing nodes generally catches most errors. If tracing is used, it should only be used for one iteration since it tends to create *very* large output files.

Error Handling

During the loading and running a model, errors are handled by the Error Manager. This section describes the general types of errors and the way they are displayed to the user.

When an error is detected a message is sent to the Error Manager. The message contains a description of the error and its severity. Severity levels are:

1. "warning" which is just advice to the user.
2. "bad net" which indicates that there is a problem with the structuring of the network. META•Net will continue loading the network, but the model will not run.

3. "fatal" which indicates problems like divide by zero and the run cannot continue. In this case META•Net attempts to write the files out and quit normally.

4. "low memory" indicates that there is not enough memory allocated to META•Net to proceed with the run.

The Error Manager writes a message to the user in the message area of the Control Panel stating that an error was detected and that the user should check the ErrorReport file. It also states whether or not META•Net will keep running until it is through loading (if it was loading when the error was detected), or will quit immediately. Generally only one message is written even if a number of errors are detected.

After the Error Manager receives an error message, it writes a record in the ErrorReport file. This record has the same header format of the other records used in META•Net. The node name and component identifier of the record indicate the node and component where the error was detected. Note that it is not always possible to determine where the error occurred so these might indicate that they are unknown.

The data fields of the record contain the following information:

1. Period: the period of the model run when the error occurred.
2. Iteration: the iteration being processed when the error occurred.
3. Severity: the severity, as described above.
4. Message: a short description of the error.
5. Current node: the node that was executing when the error occurred (sometimes the header field cannot display the name of the node but this field can because it uses a separate mechanism for keeping track of the execution of nodes)
6. Current procedure: the procedure that was executing when the error occurred. This is not infallible, but it can help with isolating a problem.

The message should be a fairly good description of the problem and indicate what can be done to fix it.

The general types of errors that can be detected are described below:

Invalid data entries: this occurs when the user enters, say, an alpha value in a field that requires a number, or a number is out of range.

Invalid node parameters: in some cases parameters must fall into a certain range or one parameter must be larger than another. The error message will explain what is incorrect.

Incorrect links between nodes: different nodes allow different number of links. META•Net checks the number of links for each node and signals an error if it is not correct.

Unable to find a needed record: As the model is loading, META•Net looks for all the records needed to specify each node in the Nodes file. If a record is missing, or the entries in the header fields are incorrectly spelled, META•Net will not be able to find the record and will signal an error.

Computational errors: these are errors such as divide by zero or underflows. Generally these are caused by errors in entering data.

Low memory: each node that is created requires a certain amount of memory. If insufficient memory has been allocated to META•Net, this error will occur. Usually it occurs during the loading process. If it does occur, use the "Get Info" window for META•Net to increase the memory allocation to META•Net.

During the model loading META•Net attempts to continue loading even when errors in the network or the nodes have been detected. This helps the user find as many errors as possible in a single run. However, as with any compiler, a single error can cause the error manager to believe that subsequent correct input is erroneous. Thus when there are several errors, some of the later error messages will not appear to make sense. They may disappear when the earlier errors are corrected.

META•Net checks for errors in many stages in the model. However, it does not check all situations where an error can occur. Eventually, through enough testing and further refinement, error trapping can be added at all points where it is needed. For the present, the user should be aware that errors can occur that will crash the operating system.

Memory Requirements

The amount of memory required by META•Net depends on the number of nodes in the model. As a rough estimate, about 200 K is required for META•Net itself, and about 16k per node is required. On the Macintosh™ the memory allocated to META•Net is adjusted by changing the "current size" in the "Get Info" window.

If there is insufficient memory for the model to run, a message will be written to the user and the run will terminate. Normally this should only happen while the model is loading. Once it is loaded it should not need to allocate any additional memory for the model. The appearance of a low memory error after the model has been running can indicate that the network is not structured correctly. As an example, this can happen when a node expects to have an input link, but none has been specified. Normally the error checking mechanism should catch this type of problem, but it may not in every circumstance.

Managing Data With A Spreadsheet

The data for models of modest size can be handled efficiently using a spreadsheet program such as Microsoft Excel™. This approach makes it possible to use files that are formatted for easy reading, and editing the data is simple. The data files for the Sample model are built with Excel.

Build the required files by entering data into a set of spreadsheets. When the files have been built, save them in "Comma Separated Value (CSV)" format and give them the correct names for the model's files. When Excel saves a file in CSV format, it saves it as a text file with each of the data fields separated by a comma.

To open and edit one of the model's files, instruct Excel to read text files as comma separated values (this is done by selecting "Options" in the "Open File" dialogue box). Then when Excel opens the file, the fields will appear in separate cells and they can be directly edited.

To make the process of editing easier, two sets of files can be kept. One set is saved in the spreadsheet's "normal" format, using names that are different from the names of the model's input files (recall that the model's files all have the name format of *"ModelName.FileType"*). These files can include formats for the column widths text styles to make them easier to read. When they have been edited, they are saved under the names of the model's input files in CSV format.

Output can also be handled by a spreadsheet. The model's output is saved in a comma separated text file named "NodeResults". This can be opened by a spreadsheet (after setting it to read comma separated values text files). The cells can be reformatted and the records can be sorted however the user requires. Then it can be saved in the spreadsheet's normal format using a name that describes the run it came from. In Excel a macro can be prepared that does all the standard formatting of column widths, text styles and number formats when the NodeResults file is opened.

References

Boyd, Dean W., Robert L. Phillips, Stephan G. Regulinski, "A model of technology selection by cost minimizing producers", in *Management Science*, Vlo. 28, No. 4, April 1982

Cazalet, E. G., *Generalized Equilibrium Modeling: The Methodology of the SRI-Gulf Energy Model,* Final report prepared by Decision Focus Inc., Palo Alto, CA for the U. S. Federal Energy Administration, May 1977

Decision Focus Inc., *The DFI Generalized Equilibrium Modeling System (GEMS),* Decision Focus Inc., Mountain View CA, 1979

Hogan, William W. and John P. Weyant, (1982), "Combined Energy Models", in *Advances in the Economics af Energy and Resources*, Vol. 4, pages 117-150, JAI Press Inc.

Lamont, Alan D. , *A Demonstration of the Network Approach to Energy Economics Modeling Using the ProtoNet Modeling System,* Lawrence Livermore National Laboratory, Livermore CA, UCRL-ID-110643, 1992

Rousseau, W. F., S.S. Sussman, R. N. Castleton, and J. T. Rambo, *Economic Models and Algorithms Used in the Livermore Economic Modeling System,* Lawrence Livermore National Laboratory, Livermore CA, UCRL-52527, 1978

Appendix A: Sample Model

This sample model is used to illustrate the model building process. It also can be used as a template for developing a new model by redefining nodes and parameters.

Figure A-1 shows the network. That is followed by the input files. These have been printed out from Excel and formatted for easy reading. The results of a model run are presented at the end of the Appendix. Again, these have been formatted in Excel.

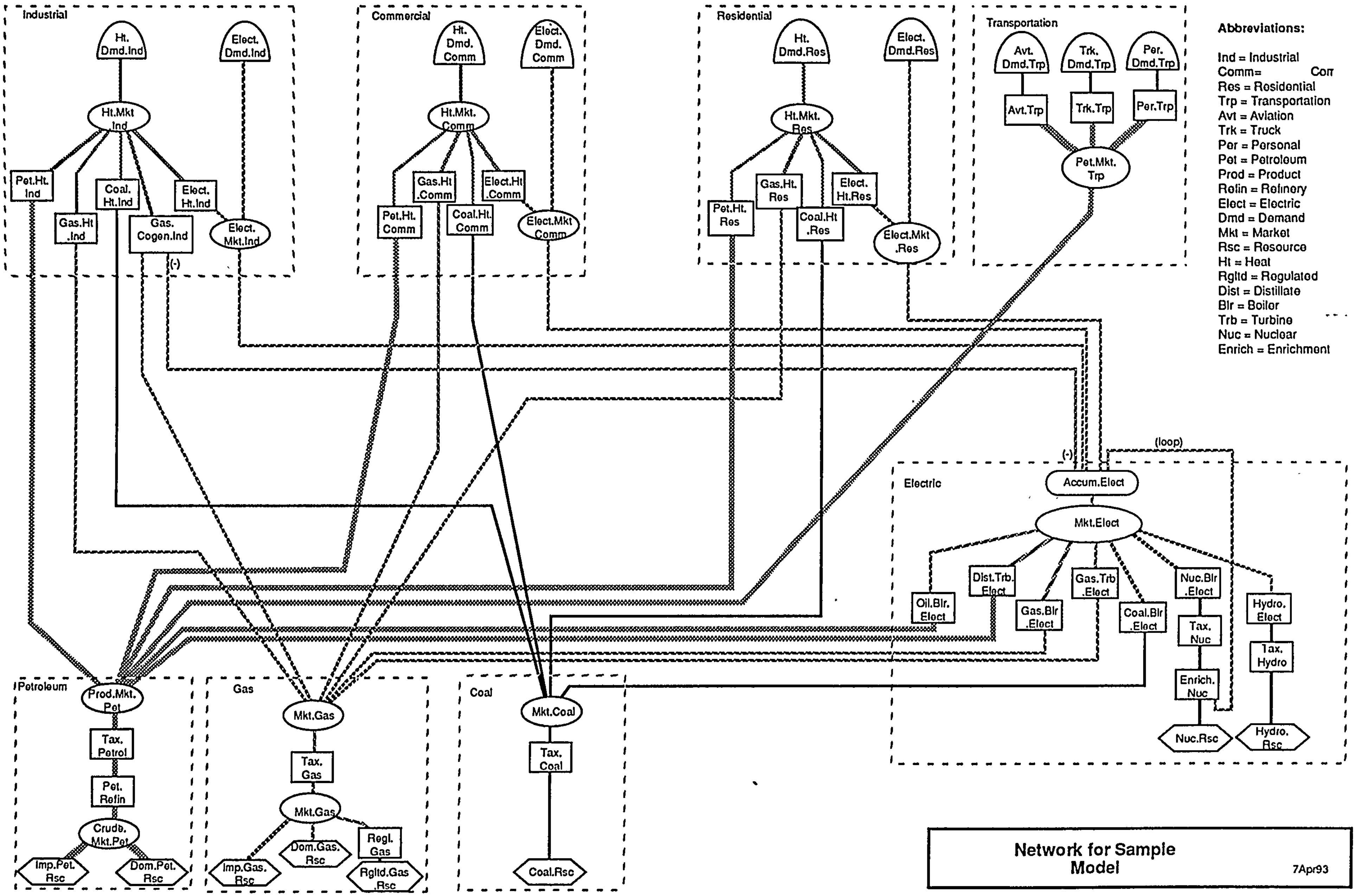
To illustrate the use of the relaxation parameter and the convergence behavior of the model, a run was made with Sample Model 1.2 using the following pattern of relaxation parameters changes. The relaxations and the observed model behavior were as follows:

From Iter.#	To Iter.#	Relax Param.	Observed Convergence Behavior
1	30	0.5	The Domestic petroleum Resource node has the largest changes. Changes are in the order of 20 to 30 (2,000 to 3,000 % change per iteration) at first. The changes decline to around 2 (200%) and oscillate between large and small values.
31	150	0.2	At about 36 iterations, the Regulated Gas Resource begins to have the largest changes. The changes decline steadily from about 0.3 (30% per iteration) to 0.098 at around 58 iterations. Then the changes slowly begin to increase to 0.114 at 100 iterations. At 118 iterations, the changes begin to decrease rapidly to 0.001 (0.1 % per iteration) at 144 iterations. Then the changes begin to oscillate between positive and negative values. Heat Demand Residential occasionally has the largest changes.
151	160	0.1	Heat Demand Residential has the largest changes. The changes decline steadily to -4.2e-4 (-0.04 % per iteration).

Appendix A: Sample Model

This sample model is used to illustrate the model building process. It also can be used as a template for developing a new model by redefining nodes and parameters.

The figure below shows the network. That is followed by the input files. These have been printed out from Excel and formatted for easy reading. The results of a model run are presented at the end of the Appendix. Again, these have been formatted in Excel.



Network for Sample Model

7Apr93

Setup File (Excel format)

c	**Setup file for Sample Model**						
c							
c	**Node**	**Comp**	**CompId**	**DataSet**	**RunTitle**		
d	na	na	na	General	Sample 1.2		
c					**NumPeriods**	**YrsPerPeriod**	**StartYear**
d	na	na	na	Periods	10	5	1990
c					**Relax**	**Relax Mkt Nodes?**	
d	na	na	na	Control	0.5	FALSE	
c					**InterestRate**	**TaxRate**	
d	na	na	na	Economic	0.1	0	
c					**TraceOn**	**MinConvrge**	
d	na	na	na	Diagnostic	FALSE	0.000001	

Nodes File (Excel format)

c	**Nodes File for Sample Model**							
c	**Name**	**Comp.**	**Comp. Id.**	**DataSet**	**Type**	**Region**	**Sector**	**Notes**
d	Top	na	na	NodeParams	Top			
c								
c	**End-use**							
d	Ht.Dmd.Ind	na	na	NodeParams	EndUse	all	Industrial	
d	Elect.Dmd.Ind	na	na	NodeParams	EndUse	all	Industrial	
d	Ht.Dmd.Comm	na	na	NodeParams	EndUse	all	Commercial	
d	Elect.Dmd.Comm	na	na	NodeParams	EndUse	all	Commercial	
d	Ht.Dmd.Res	na	na	NodeParams	EndUse	all	Residential	
d	Elect.Dmd.Res	na	na	NodeParams	EndUse	all	Residential	
d	Avt.Dmd.Trp	na	na	NodeParams	EndUse	all	Transportation	
d	Trk.Dmd.Trp	na	na	NodeParams	EndUse	all	Transportation	
d	Per.Dmd.Trp	na	na	NodeParams	EndUse	all	Transportation	
c								
c	**Markets**							
d	Ht.Mkt.Ind	na	na	NodeParams	Market	all	Industrial	
d	Elect.Mkt.Ind	na	na	NodeParams	Market	all	Industrial	
d	Ht.Mkt.Comm	na	na	NodeParams	Market	all	Commercial	
d	Elect.Mkt.Comm	na	na	NodeParams	Market	all	Commercial	
d	Ht.Mkt.Res	na	na	NodeParams	Market	all	Residential	
d	Elect.Mkt.Res	na	na	NodeParams	Market	all	Residential	
d	Pet.Mkt.Trp	na	na	NodeParams	Market	all	Transportation	
d	Prod.Mkt.Pet	na	na	NodeParams	Market	all	Peroleum	
d	Crude.Mkt.Pet	na	na	NodeParams	Market	all	Peroleum	
d	Mkt1.Gas	na	na	NodeParams	Market	all	Gas	
d	Mkt2.Gas	na	na	NodeParams	Market	all	Gas	
d	Mkt.Coal	na	na	NodeParams	Market	all	Coal	
d	Accum.Elect	na	na	NodeParams	SimpLFAccum	all	Electric	
d	Mkt.Elect	na	na	NodeParams	MultLFMarket	all	Electric	
c								
c	**Conversions**							
d	Pet.Ht.Ind	na	na	NodeParams	TB1Conv	all	Industrial	

d	Gas.Ht.Ind	na	na	NodeParams	TB1Conv	all	Industrial	
d	Coal.Ht.Ind	na	na	NodeParams	TB1Conv	all	Industrial	
d	Gas.Cogen.Ind	na	na	NodeParams	TB1Conv	all	Industrial	
d	Elect.Ht.Ind	na	na	NodeParams	TB1Conv	all	Industrial	
d	Pet.Ht.Comm	na	na	NodeParams	TB1Conv	all	Commercial	
d	Gas.Ht.Comm	na	na	NodeParams	TB1Conv	all	Commercial	
d	Coal.Ht.Comm	na	na	NodeParams	TB1Conv	all	Commercial	
d	Elect.Ht.Comm	na	na	NodeParams	TB1Conv	all	Commercial	
d	Pet.Ht.Res	na	na	NodeParams	TB1Conv	all	Residential	
d	Gas.Ht.Res	na	na	NodeParams	TB1Conv	all	Residential	
d	Coal.Ht.Res	na	na	NodeParams	TB1Conv	all	Residential	
d	Elect.Ht.Res	na	na	NodeParams	TB1Conv	all	Residential	
d	Avt.Trp	na	na	NodeParams	TB1Conv	all	Transportation	
d	Trk.Trp	na	na	NodeParams	TB1Conv	all	Transportation	
d	Per.Trp	na	na	NodeParams	TB1Conv	all	Transportation	
d	Refin.Pet	na	na	NodeParams	TB1Conv	all	Peroleum	
d	Tax.Pet	na	na	NodeParams	TB1Conv	all	Peroleum	
d	Tax.Gas	na	na	NodeParams	TB1Conv	all	Gas	
d	Tax.Coal	na	na	NodeParams	TB1Conv	all	Coal	
d	Tax.Nuc	na	na	NodeParams	TB1Conv	all	Electric	
d	Tax.Hydro	na	na	NodeParams	TB1Conv	all	Electric	
d	Reg.Gas	na	na	NodeParams	TB1Conv	all	Gas	
d	Enrich.Nuc	na	na	NodeParams	TB1Conv	all	Electric	
d	Oil.Blr.Elect	na	na	NodeParams	TBLFConv	all	Electric	
d	Dist.Trb.Elect	na	na	NodeParams	TBLFConv	all	Electric	
d	Gas.Blr.Elect	na	na	NodeParams	TBLFConv	all	Electric	
d	Gas.Trb.Elect	na	na	NodeParams	TBLFConv	all	Electric	
d	Coal.Blr.Elect	na	na	NodeParams	TBLFConv	all	Electric	
d	Nuc.Blr.Elect	na	na	NodeParams	TBLFConv	all	Electric	
d	Hydro.Elect	na	na	NodeParams	TBLFConv	all	Electric	
c								
c	**Resources**							
d	Imp.Pet.Rsc	na	na	NodeParams	ResourceP	all	Peroleum	

d	Dom.Pet.Rsc	na	na	NodeParams	Resource	all	Peroleum	
d	Imp.Gas.Rsc	na	na	NodeParams	ResourceP	all	Gas	
d	Dom.Gas.Rsc	na	na	NodeParams	Resource	all	Gas	
d	Rgltd.Gas.Rsc	na	na	NodeParams	Resource	all	Gas	
d	Coal.Rsc	na	na	NodeParams	Resource	all	Coal	
d	Nuc.Rsc	na	na	NodeParams	Resource	all	Electric	
d	Hydro.Rsc	na	na	NodeParams	ResourceP	all	Electric	

Links File (Excel format)

Links File for Sample Model

c								
c	**Name**	**Comp.**	**Comp**	**DataSet**	**DownNode**	**NetType**	**CustomerType**	**InputType**
d	Top	InLink	1	LinkData	Ht.Dmd.Ind	normal	all	operating
d	Top	InLink	2	LinkData	Elect.Dmd.Ind	normal	all	operating
d	Top	InLink	3	LinkData	Ht.Dmd.Comm	normal	all	operating
d	Top	InLink	4	LinkData	Elect.Dmd.Comm	normal	all	operating
d	Top	InLink	5	LinkData	Ht.Dmd.Res	normal	all	operating
d	Top	InLink	6	LinkData	Elect.Dmd.Res	normal	all	operating
d	Top	InLink	7	LinkData	Avt.Dmd.Trp	normal	all	operating
d	Top	InLink	8	LinkData	Trk.Dmd.Trp	normal	all	operating
d	Top	InLink	9	LinkData	Per.Dmd.Trp	normal	all	operating
c								
c	**End-use**							
d	Ht.Dmd.Ind	InLink	1	LinkData	Ht.Mkt.Ind	normal	all	operating
d	Elect.Dmd.Ind	InLink	1	LinkData	Elect.Mkt.Ind	normal	all	operating
d	Ht.Dmd.Comm	InLink	1	LinkData	Ht.Mkt.Comm	normal	all	operating
d	Elect.Dmd.Comm	InLink	1	LinkData	Elect.Mkt.Comm	normal	all	operating
d	Ht.Dmd.Res	InLink	1	LinkData	Ht.Mkt.Res	normal	all	operating
d	Elect.Dmd.Res	InLink	1	LinkData	Elect.Mkt.Res	normal	all	operating
d	Avt.Dmd.Trp	InLink	1	LinkData	Avt.Trp	normal	all	operating
d	Trk.Dmd.Trp	InLink	1	LinkData	Trk.Trp	normal	all	operating
d	Per.Dmd.Trp	InLink	1	LinkData	Per.Trp	normal	all	operating
c								
c	**Markets**							
d	Ht.Mkt.Ind	InLink	1	LinkData	Pet.Ht.Ind	normal	all	operating
d	Ht.Mkt.Ind	InLink	2	LinkData	Gas.Ht.Ind	normal	all	operating
d	Ht.Mkt.Ind	InLink	3	LinkData	Coal.Ht.Ind	normal	all	operating
d	Ht.Mkt.Ind	InLink	4	LinkData	Gas.Cogen.Ind	normal	all	operating
d	Ht.Mkt.Ind	InLink	5	LinkData	Elect.Ht.Ind	normal	all	operating
d	Elect.Mkt.Ind	InLink	1	LinkData	Accum.Elect	normal	all	operating
d	Ht.Mkt.Comm	InLink	1	LinkData	Pet.Ht.Comm	normal	all	operating
d	Ht.Mkt.Comm	InLink	2	LinkData	Gas.Ht.Comm	normal	all	operating

d	Ht.Mkt.Comm	InLink	3	LinkData	Coal.Ht.Comm	normal	all	operating
d	Ht.Mkt.Comm	InLink	4	LinkData	Elect.Ht.Comm	normal	all	operating
d	Elect.Mkt.Comm	InLink	1	LinkData	Accum.Elect	normal	all	operating
d	Ht.Mkt.Res	InLink	1	LinkData	Pet.Ht.Res	normal	all	operating
d	Ht.Mkt.Res	InLink	2	LinkData	Gas.Ht.Res	normal	all	operating
d	Ht.Mkt.Res	InLink	3	LinkData	Coal.Ht.Res	normal	all	operating
d	Ht.Mkt.Res	InLink	4	LinkData	Elect.Ht.Res	normal	all	operating
d	Elect.Mkt.Res	InLink	1	LinkData	Accum.Elect	normal	all	operating
d	Pet.Mkt.Trp	InLink	1	LinkData	Prod.Mkt.Pet	normal	all	operating
d	Prod.Mkt.Pet	InLink	1	LinkData	Tax.Pet	normal	all	operating
d	Crude.Mkt.Pet	InLink	1	LinkData	Imp.Pet.Rsc	normal	all	operating
d	Crude.Mkt.Pet	InLink	2	LinkData	Dom.Pet.Rsc	normal	all	operating
d	Mkt1.Gas	InLink	1	LinkData	Imp.Gas.Rsc	normal	all	operating
d	Mkt1.Gas	InLink	2	LinkData	Dom.Gas.Rsc	normal	all	operating
d	Mkt1.Gas	InLink	3	LinkData	Reg.Gas	normal	all	operating
d	Mkt2.Gas	InLink	1	LinkData	Tax.Gas	normal	all	operating
d	Mkt.Coal	InLink	1	LinkData	Tax.Coal	normal	all	operating
d	Accum.Elect	InLink	1	LinkData	Mkt.Elect	normal	all	operating
d	Mkt.Elect	InLink	1	LinkData	Oil.Blr.Elect	normal	all	operating
d	Mkt.Elect	InLink	2	LinkData	Dist.Trb.Elect	normal	all	operating
d	Mkt.Elect	InLink	3	LinkData	Gas.Blr.Elect	normal	all	operating
d	Mkt.Elect	InLink	4	LinkData	Gas.Trb.Elect	normal	all	operating
d	Mkt.Elect	InLink	5	LinkData	Coal.Blr.Elect	normal	all	operating
d	Mkt.Elect	InLink	6	LinkData	Nuc.Blr.Elect	normal	all	operating
d	Mkt.Elect	InLink	7	LinkData	Hydro.Elect	normal	all	operating
c								
c	**Conversions**							
d	Pet.Ht.Ind	InLink	1	LinkData	Prod.Mkt.Pet	normal	all	operating
d	Gas.Ht.Ind	InLink	1	LinkData	Mkt2.Gas	normal	all	operating
d	Coal.Ht.Ind	InLink	1	LinkData	Mkt.Coal	normal	all	operating
d	Gas.Cogen.Ind	InLink	1	LinkData	Mkt2.Gas	normal	all	operating
d	Gas.Cogen.Ind	InLink	2	LinkData	Accum.Elect	normal	all	operating

d	Elect.Ht.Ind	InLink	1	LinkData	Elect.Mkt.Ind	normal	all	operating
d	Pet.Ht.Comm	InLink	1	LinkData	Prod.Mkt.Pet	normal	all	operating
d	Gas.Ht.Comm	InLink	1	LinkData	Mkt2.Gas	normal	all	operating
d	Coal.Ht.Comm	InLink	1	LinkData	Mkt.Coal	normal	all	operating
d	Elect.Ht.Comm	InLink	1	LinkData	Elect.Mkt.Comm	normal	all	operating
d	Pet.Ht.Res	InLink	1	LinkData	Prod.Mkt.Pet	normal	all	operating
d	Gas.Ht.Res	InLink	1	LinkData	Mkt2.Gas	normal	all	operating
d	Coal.Ht.Res	InLink	1	LinkData	Mkt.Coal	normal	all	operating
d	Elect.Ht.Res	InLink	1	LinkData	Elect.Mkt.Res	normal	all	operating
d	Avt.Trp	InLink	1	LinkData	Pet.Mkt.Trp	normal	all	operating
d	Trk.Trp	InLink	1	LinkData	Pet.Mkt.Trp	normal	all	operating
d	Per.Trp	InLink	1	LinkData	Pet.Mkt.Trp	normal	all	operating
d	Refin.Pet	InLink	1	LinkData	Crude.Mkt.Pet	normal	all	operating
d	Tax.Pet	InLink	1	LinkData	Refin.Pet	normal	all	operating
d	Tax.Gas	InLink	1	LinkData	Mkt1.Gas	normal	all	operating
d	Tax.Coal	InLink	1	LinkData	Coal.Rsc	normal	all	operating
d	Tax.Nuc	InLink	1	LinkData	Enrich.Nuc	normal	all	operating
d	Tax.Hydro	InLink	1	LinkData	Hydro.Rsc	normal	all	operating
d	Reg.Gas	InLink	1	LinkData	Rgltd.Gas.Rsc	normal	all	operating
d	Enrich.Nuc	InLink	1	LinkData	Nuc.Rsc	normal	all	operating
d	Enrich.Nuc	InLink	2	LinkData	Accum.Elect	loop	all	operating
d	Oil.Blr.Elect	InLink	1	LinkData	Prod.Mkt.Pet	normal	all	operating
d	Dist.Trb.Elect	InLink	1	LinkData	Prod.Mkt.Pet	normal	all	operating
d	Gas.Blr.Elect	InLink	1	LinkData	Mkt2.Gas	normal	all	operating
d	Gas.Trb.Elect	InLink	1	LinkData	Mkt2.Gas	normal	all	operating
d	Coal.Blr.Elect	InLink	1	LinkData	Mkt.Coal	normal	all	operating
d	Nuc.Blr.Elect	InLink	1	LinkData	Tax.Nuc	normal	all	operating
d	Hydro.Elect	InLink	1	LinkData	Tax.Hydro	normal	all	operating
d	Tax.Hydro	InLink	1	LinkData	Hydro.Rsc	normal	all	operating

Components File (Excel format)

c	**Components File for Sample Model**								
c	**Name**	**Comp.**	**Comp. Id.**	**DataSet**					
c	**End-use**				**Start Quant**	**Dem Growth**	**Start Price**	**Price Growth**	**Elast**
d	Ht.Dmd.Ind	na	na	NodeData	1.5E+10	0.020	7.790	0.000	-0.310
d	Elect.Dmd.Ind	na	na	NodeData	1.0E+9	0.020	15.000	0.000	-0.310
d	Ht.Dmd.Comm	na	na	NodeData	4.0E+9	0.020	5.821	0.000	-0.300
d	Elect.Dmd.Co mm	na	na	NodeData	2.0E+9	0.020	15.000	0.000	-0.300
d	Ht.Dmd.Res	na	na	NodeData	5.0E+9	0.020	5.413	0.000	-0.480
d	Elect.Dmd.Res	na	na	NodeData	2.0E+9	0.020	15.000	0.000	-0.480
d	Avt.Dmd.Trp	na	na	NodeData	6.2E+8	0.020	136.240	0.000	-0.400
d	Trk.Dmd.Trp	na	na	NodeData	3.0E+9	0.020	13.250	0.000	-0.400
d	Per.Dmd.Trp	na	na	NodeData	2.9E+9	0.020	59.820	0.000	-1.080
c	**Note: Avt demand in K passenger miles/ yr; Trk demand in K ton miles/ Yr; Per demand in K passenger miles/ Yr**								
c									
c	**Markets**				**Price Sens**				
d	Ht.Mkt.Ind	na	na	NodeData	5				
d	Elect.Mkt.Ind	na	na	NodeData	5				
d	Ht.Mkt.Comm	na	na	NodeData	5				
d	Elect.Mkt.Com m	na	na	NodeData	5				
d	Ht.Mkt.Res	na	na	NodeData	5				
d	Elect.Mkt.Res	na	na	NodeData	5				
d	Pet.Mkt.Trp	na	na	NodeData	5				
d	Prod.Mkt.Pet	na	na	NodeData	5				
d	Crude.Mkt.Pet	na	na	NodeData	5				
d	Mkt1.Gas	na	na	NodeData	5				
d	Mkt2.Gas	na	na	NodeData	5				

d	Mkt.Coal	na	na	NodeData	5									
d	Mkt.Elect	na	na	NodeData	5									
c				**Header for Accum =>**	**Fraction 1**	**Load Factor 1**	**Fraction 2**	**Load Factor 2**	**Fraction 3**	**Load Factor 3**				
d	Accum.Elect	na	na	NodeData	0.60	1.00	0.35	0.40	0.05	0.10				
c														
c	**Conversions**		**Header for NodeData =>**		**First Year Avail**	**Spec Cap Cost**	**Ancil Oper Cost**	**Life**	**Avail. Factor**	**Min Cap Factor**				
c			**Header for TBlockData =>**		**Start Cap**									
c			**Header for InputLink Data =>**		**I-O Coef**									
d	Pet.Ht.Ind	na	na	NodeData	1950	6.81	0.48	25	1	0.7				
d	Pet.Ht.Ind	TBlock	1	TBlockData	1.31E+9									
d	Pet.Ht.Ind	InLink	Prod.Mkt.Pet	InputLinkData	1.23									
c														
d	Gas.Ht.Ind	na	na	NodeData	1950	6	0.39	25	0.9	0.7				
d	Gas.Ht.Ind	TBlock	1	TBlockData	1.37E+10									
d	Gas.Ht.Ind	InLink	Mkt2.Gas	InputLinkData	1.23									
c														
d	Coal.Ht.Ind	na	na	NodeData	1950	14.58	0.54	25	1	0.7				
d	Coal.Ht.Ind	TBlock	1	TBlockData	3.88E+9									
d	Coal.Ht.Ind	InLink	Mkt.Coal	InputLinkData	1.31									
c														
d	Gas.Cogen.Ind	na	na	NodeData	1980	23.19	2.59	25	1	0.7				

d	Gas.Cogen.Ind	TBlock	1	TBlockData	7.53E+8									
d	Gas.Cogen.Ind	InLink	Mkt2.Gas	InputLinkData	1.72									
d	Gas.Cogen.Ind	InLink	Accum.Elect	InputLinkData	-0.1									
c														
d	Elect.Ht.Ind	na	na	NodeData	1950	3.4	0.06	25	1	0.7				
d	Elect.Ht.Ind	TBlock	1	TBlockData	1.83E+7									
d	Elect.Ht.Ind	InLink	Elect.Mkt.Ind	InputLinkData	1.02									
c														
d	Pet.Ht.Comm	na	na	NodeData	1950	40.9	5.1	10	0.9	0.7				
d	Pet.Ht.Comm	TBlock	1	TBlockData	7.68E+8									
d	Pet.Ht.Comm	InLink	Prod.Mkt.Pet	InputLinkData	1.35									
c														
d	Gas.Ht.Comm	na	na	NodeData	1950	51.22	4.41	10	1	0.7				
d	Gas.Ht.Comm	TBlock	1	TBlockData	1.19E+9									
d	Gas.Ht.Comm	InLink	Mkt2.Gas	InputLinkData	1.67									
c														
d	Coal.Ht.Comm	na	na	NodeData	1950	80	15	10	1	0.7				
d	Coal.Ht.Comm	TBlock	1	TBlockData	4.08E+8									
d	Coal.Ht.Comm	InLink	Mkt.Coal	InputLinkData	1.5									
c														
d	Elect.Ht.Comm	na	na	NodeData	1950	15.5	2.9	25	0.9	0.7				
d	Elect.Ht.Comm	TBlock	1	TBlockData	6.98E+8									
d	Elect.Ht.Comm	InLink	Elect.Mkt.Com	InputLinkData	1.02									
c														
d	Pet.Ht.Res	na	na	NodeData	1950	28.7	1.9	20	1	0.7				
d	Pet.Ht.Res	TBlock	1	TBlockData	9.05E+8									

d	Pet.Ht.Res	InLink	Prod.Mkt.Pet	InputLinkData	1.19									
c														
d	Gas.Ht.Res	na	na	NodeData	1950	12.95	1.6	16	1	0.7				
d	Gas.Ht.Res	TBlock	1	TBlockData	3.85E+9									
d	Gas.Ht.Res	InLink	Mkt2.Gas	InputLinkData	1.5									
c														
d	Coal.Ht.Res	na	na	NodeData	1950	80	14	20	0.95	0.7				
d	Coal.Ht.Res	TBlock	1	TBlockData	2.05E+7									
d	Coal.Ht.Res	InLink	Mkt.Coal	InputLinkData	1.5									
c														
d	Elect.Ht.Res	na	na	NodeData	1950	9.24	1.63	20	1	0.7				
d	Elect.Ht.Res	TBlock	1	TBlockData	4.31E+7									
d	Elect.Ht.Res	InLink	Elect.Mkt.Res	InputLinkData	1.02									
c														
d	Avt.Trp	na	na	NodeData	1950	462.5	116.82	15	1	0.7				
d	Avt.Trp	TBlock	1	TBlockData	4.60E+8									
d	Avt.Trp	InLink	Pet.Mkt.Trp	InputLinkData	38.5									
c														
d	Trk.Trp	na	na	NodeData	1900	0.055	0.136	15	1	0.7				
d	Trk.Trp	TBlock	1	TBlockData	3.90E+9									
d	Trk.Trp	InLink	Pet.Mkt.Trp	InputLinkData	1.89									
c														
d	Per.Trp	na	na	NodeData	1900	537.74	44.64	10	1	0.7				
d	Per.Trp	TBlock	1	TBlockData	1.40E+9									
d	Per.Trp	InLink	Pet.Mkt.Trp	InputLinkData	3.85									
c														
d	Refin.Pet	na	na	NodeData	1900	2.78	0.46	30	0.95	0.7				
d	Refin.Pet	TBlock	1	TBlockData	3.60E+10									

d	Refin.Pet	InLink	Crude.Mkt.Pet	InputLinkData	1.05									
c														
d	Tax.Pet	na	na	NodeData	1990	0	0.6	5	1.00	0.5				
d	Tax.Pet	TBlock	1	TBlockData	1.00E+10									
d	Tax.Pet	InLink	Refin.Pet	InputLinkData	1.00									
c														
d	Tax.Gas	na	na	NodeData	1990	0	0.26	5	1.00	0.5				
d	Tax.Gas	TBlock	1	TBlockData	1.00E+10									
d	Tax.Gas	InLink	Mkt1.Gas	InputLinkData	1.00									
c														
d	Tax.Coal	na	na	NodeData	1990	0	0.26	5	1.00	0.5				
d	Tax.Coal	TBlock	1	TBlockData	1.00E+10									
d	Tax.Coal	InLink	Coal.Rsc	InputLinkData	1.00									
c														
d	Tax.Nuc	na	na	NodeData	1990	0	0.26	5	1.00	0.5				
d	Tax.Nuc	TBlock	1	TBlockData	1.00E+10									
d	Tax.Nuc	InLink	Enrich.Nuc	InputLinkData	1.00									
c														
d	Tax.Hydro	na	na	NodeData	1900	0	0.26	100	1	0.5				
d	Tax.Hydro	TBlock	1	TBlockData	1.0E+9									
d	Tax.Hydro	InLink	Hydro.Rsc	InputLinkData	1.00									
c														
d	Reg.Gas	na	na	NodeData	1900	0	0	5	1	0.9				
d	Reg.Gas	TBlock	1	TBlockData	1.0E+11									
d	Reg.Gas	InLink	Rgltd.Gas.Rsc	InputLinkData	1.00									
c														
d	Enrich.Nuc	na	na	NodeData	1900	3.3	0.35	30	0.8	0.1				
d	Enrich.Nuc	TBlock	1	TBlockData	3.29E+9									
d	Enrich.Nuc	InLink	Nuc.Rsc	InputLinkData	1.2									
d	Enrich.Nuc	InLink	Accum.Elect	InputLinkData	0.1									

c														
d	Oil.Blr.Elect	na	na	NodeData	1930	26.48	0.75	20	0.8	0.4				
d	Oil.Blr.Elect	TBlock	1	TBlockData	1.84E+8									
d	Oil.Blr.Elect	TBlock	2	TBlockData	1.33E+8									
d	Oil.Blr.Elect	TBlock	3	TBlockData	6.24E+6									
d	Oil.Blr.Elect	InLink	Prod.Mkt.Pet	InputLinkData	2.7									
c														
d	Dist.Trb.Elect	na	na	NodeData	1950	12.58	2.28	30	0.92	0.4				
d	Dist.Trb.Elect	TBlock	1	TBlockData	3.30E+7									
d	Dist.Trb.Elect	TBlock	2	TBlockData	6.30E+7									
d	Dist.Trb.Elect	TBlock	3	TBlockData	2.94E+7									
d	Dist.Trb.Elect	InLink	Prod.Mkt.Pet	InputLinkData	4.2									
c														
d	Gas.Blr.Elect	na	na	NodeData	1930	23.84	0.75	30	0.8	0.4				
d	Gas.Blr.Elect	TBlock	1	TBlockData	1.38E+9									
d	Gas.Blr.Elect	TBlock	2	TBlockData	7.11E+8									
d	Gas.Blr.Elect	TBlock	3	TBlockData	2.28E+7									
d	Gas.Blr.Elect	InLink	Mkt2.Gas	InputLinkData	3.03									
c														
d	Gas.Trb.Elect	na	na	NodeData	1950	12.31	2.25	30	0.92	0.4				
d	Gas.Trb.Elect	TBlock	1	TBlockData	4.45E+8									
d	Gas.Trb.Elect	TBlock	2	TBlockData	5.77E+8									
d	Gas.Trb.Elect	TBlock	3	TBlockData	1.26E+8									
d	Gas.Trb.Elect	InLink	Mkt2.Gas	InputLinkData	4.35									
c														
d	Coal.Blr.Elect	na	na	NodeData	1930	42.65	1.2	30	0.85	0.4				
d	Coal.Blr.Elect	TBlock	1	TBlockData	1.29E+9									

d	Coal.Blr.Elect	TBlock	2	TBlockData	4.16E+8									
d	Coal.Blr.Elect	TBlock	3	TBlockData	4.37E+6									
d	Coal.Blr.Elect	InLink	Mkt.Coal	InputLinkData	3.03									
c														
d	Nuc.Blr.Elect	na	na	NodeData	1955	59.99	1.13	30	0.62	0.3				
d	Nuc.Blr.Elect	TBlock	1	TBlockData	5.61E+8									
d	Nuc.Blr.Elect	TBlock	2	TBlockData	1.92E+8									
d	Nuc.Blr.Elect	TBlock	3	TBlockData	7.88E+6									
d	Nuc.Blr.Elect	InLink	Tax.Nuc	InputLinkData	3.22									
c														
d	Hydro.Elect	na	na	NodeData	1900	0	0.1	100	1	0.5				
d	Hydro.Elect	TBlock	1	TBlockData	1.05E+8									
d	Hydro.Elect	TBlock	2	TBlockData	2.35E+8									
d	Hydro.Elect	TBlock	3	TBlockData	1.36E+8									
d	Hydro.Elect	InLink	Tax.Hydro	InputLinkData	1.05									
c														
c	**Resources**			**Header for Resource Node =>**	**Price 0**	**Price 1**	**Commit 1**	**Price 2**	**Commit 2**	**Rsrc Type**	**Init Prod**	**Prod/ Rsrves**	**Resrc Life**	**Disc Rate**
c				**Header for Recource Price Node =>**	**Start Price**	**Price Growth**								
d	Imp.Pet.Rsc	na	na	NodeData	2.5	0.02	0.00							
d	Dom.Pet.Rsc	na	na	NodeData	2.2	5.00	1.0E+11	10.00	1.5E+11	well	2.11E+10	0.1	10	0.15
d	Imp.Gas.Rsc	na	na	NodeData	2	0.02	0.00							
d	Dom.Gas.Rsc	na	na	NodeData	1.5	3.00	1.0E+11	6.00	2.0E+11	well	1.71E+10	0.1	10	0.15
d	Rgltd.Gas.Rsc	na	na	NodeData	1.5	3.50	1.0E+11	7.00	2.0E+11	well	1.70E+10	0.1	10	0.15
d	Coal.Rsc	na	na	NodeData	1.5	2.00	3.0E+11	2.50	6.0E+11	mine	1.09E+10	0.1	30	0.15
d	Nuc.Rsc	na	na	NodeData	0.74	1.22	1.0E+11	2.44	1.0E+12	mine	2.94E+9	0.1	30	0.15
d	Hydro.Rsc	na	na	NodeData	0.5	0.00	0.0E+0							

Constraints File (Excel format)

c	Constraints File for sample Model																
c	Name	Comp.	Comp. Id.	DataSet	Constr SensA	Constr Sens B	Max Constr Ratio	1	2	3	4	5	6	7	8	9	10
d	Reg.Gas	TBlock	1	PriceConstraint	3	3	2	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
d	Tax.Hydro	TBlock	1	QuantityConstraint	1	0.4	2	5.0E+8	5.0E+8	5.0E+8	5.0E+8	5.0E+8	5.0E+8	5.0E+8	5.0E+8	5.0E+8	5.0E+8

Sample Results

Note that except for the transportation sector, all quantities are in mmBtus, annual quantity flows are in mmBtus per year and prices are in $/mmBtus. In the transportation sector truck demand is in thousands of ton-miles per year and prices are in $ per thousand ton miles. Aviation and Personal transportation demands are in thousands of passenger miles and prices are in $ per thousand passenger miles.

c	Node	Comp	CompID	DataSet	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9	Period 10
c	Sample 1.2													
d	Ht.Dmd.Ind	na	na	Prices	3.78E+00	4.52E+00	5.08E+00	5.48E+00	5.85E+00	6.27E+00	6.73E+00	7.25E+00	7.81E+00	8.43E+00
d	Ht.Dmd.Ind	na	na	Quantities	1.95E+10	2.04E+10	2.17E+10	2.34E+10	2.53E+10	2.74E+10	2.96E+10	3.19E+10	3.44E+10	3.71E+10
d	Elect.Dmd.Ind	na	na	Prices	1.28E+01	1.50E+01	1.66E+01	1.77E+01	1.87E+01	1.99E+01	2.11E+01	2.25E+01	2.40E+01	2.56E+01
d	Elect.Dmd.Ind	na	na	Quantities	1.09E+09	1.15E+09	1.23E+09	1.33E+09	1.44E+09	1.56E+09	1.70E+09	1.84E+09	1.99E+09	2.15E+09
d	Ht.Dmd.Comm	na	na	Prices	1.66E+01	1.78E+01	1.85E+01	1.91E+01	1.96E+01	2.02E+01	2.09E+01	2.17E+01	2.25E+01	2.34E+01
d	Ht.Dmd.Comm	na	na	Quantities	3.04E+09	3.29E+09	3.59E+09	3.92E+09	4.29E+09	4.70E+09	5.14E+09	5.61E+09	6.13E+09	6.68E+09
d	Elect.Dmd.Comm	na	na	Prices	1.28E+01	1.50E+01	1.66E+01	1.77E+01	1.87E+01	1.99E+01	2.11E+01	2.25E+01	2.40E+01	2.56E+01
d	Elect.Dmd.Comm	na	na	Quantities	2.18E+09	2.30E+09	2.46E+09	2.67E+09	2.89E+09	3.14E+09	3.40E+09	3.69E+09	3.99E+09	4.32E+09
d	Ht.Dmd.Res	na	na	Prices	6.43E+00	7.42E+00	8.21E+00	8.73E+00	9.20E+00	9.75E+00	1.04E+01	1.10E+01	1.18E+01	1.26E+01
d	Ht.Dmd.Res	na	na	Quantities	4.79E+09	4.94E+09	5.19E+09	5.57E+09	5.99E+09	6.43E+09	6.90E+09	7.40E+09	7.91E+09	8.44E+09
d	Elect.Dmd.Res	na	na	Prices	1.28E+01	1.50E+01	1.66E+01	1.77E+01	1.87E+01	1.99E+01	2.11E+01	2.25E+01	2.40E+01	2.56E+01
d	Elect.Dmd.Res	na	na	Quantities	2.24E+09	2.30E+09	2.42E+09	2.59E+09	2.78E+09	2.98E+09	3.20E+09	3.43E+09	3.67E+09	3.93E+09
d	Avt.Dmd.Trp	na	na	Prices	3.21E+02	3.45E+02	3.61E+02	3.75E+02	3.90E+02	4.06E+02	4.24E+02	4.44E+02	4.66E+02	4.90E+02
d	Avt.Dmd.Trp	na	na	Quantities	4.58E+08	4.91E+08	5.32E+08	5.79E+08	6.29E+08	6.84E+08	7.42E+08	8.04E+08	8.71E+08	9.42E+08
d	Trk.Dmd.Trp	na	na	Prices	7.48E+00	8.62E+00	9.43E+00	1.01E+01	1.08E+01	1.16E+01	1.25E+01	1.35E+01	1.46E+01	1.58E+01
d	Trk.Dmd.Trp	na	na	Quantities	3.92E+09	4.09E+09	4.36E+09	4.68E+09	5.03E+09	5.40E+09	5.79E+09	6.20E+09	6.64E+09	7.10E+09
d	Per.Dmd.Trp	na	na	Prices	1.27E+02	1.41E+02	1.43E+02	1.45E+02	1.46E+02	1.48E+02	1.49E+02	1.51E+02	1.54E+02	1.56E+02
d	Per.Dmd.Trp	na	na	Quantities	1.34E+09	1.31E+09	1.43E+09	1.57E+09	1.71E+09	1.87E+09	2.03E+09	2.21E+09	2.41E+09	2.61E+09
c														
d	Ht.Mkt.Ind	Allocator	1	Prices	3.78E+00	4.52E+00	5.08E+00	5.48E+00	5.85E+00	6.27E+00	6.73E+00	7.25E+00	7.81E+00	8.43E+00
d	Ht.Mkt.Ind	Uplink	Ht.Dmd.Ind	Quantities	1.95E+10	2.04E+10	2.17E+10	2.34E+10	2.53E+10	2.74E+10	2.96E+10	3.19E+10	3.44E+10	3.71E+10
c														
d	Elect.Mkt.Ind	Allocator	1	Prices	1.28E+01	1.50E+01	1.66E+01	1.77E+01	1.87E+01	1.99E+01	2.11E+01	2.25E+01	2.40E+01	2.56E+01
d	Elect.Mkt.Ind	Uplink	Elect.Dmd.Ind	Quantities	1.09E+09	1.15E+09	1.23E+09	1.33E+09	1.44E+09	1.56E+09	1.70E+09	1.84E+09	1.99E+09	2.15E+09
d	Elect.Mkt.Ind	Uplink	Elect.Ht.Ind	Quantities	1.47E+07	1.49E+07	1.69E+07	1.92E+07	2.16E+07	2.43E+07	2.74E+07	3.10E+07	3.52E+07	3.99E+07
c														
d	Ht.Mkt.Comm	Allocator	1	Prices	1.66E+01	1.78E+01	1.85E+01	1.91E+01	1.96E+01	2.02E+01	2.09E+01	2.17E+01	2.25E+01	2.34E+01
d	Ht.Mkt.Comm	Uplink	Ht.Dmd.Comm	Quantities	3.04E+09	3.29E+09	3.59E+09	3.92E+09	4.29E+09	4.70E+09	5.14E+09	5.61E+09	6.13E+09	6.68E+09
c														
d	Elect.Mkt.Comm	Allocator	1	Prices	1.28E+01	1.50E+01	1.66E+01	1.77E+01	1.87E+01	1.99E+01	2.11E+01	2.25E+01	2.40E+01	2.56E+01
d	Elect.Mkt.Comm	Uplink	Elect.Dmd.Comr	Quantities	2.18E+09	2.30E+09	2.46E+09	2.67E+09	2.89E+09	3.14E+09	3.40E+09	3.69E+09	3.99E+09	4.32E+09
d	Elect.Mkt.Comm	Uplink	Elect.Ht.Comm	Quantities	6.36E+08	5.66E+08	5.48E+08	5.36E+08	5.52E+08	5.55E+08	5.57E+08	5.61E+08	5.65E+08	5.70E+08
c														
d	Ht.Mkt.Res	Allocator	1	Prices	6.43E+00	7.42E+00	8.21E+00	8.73E+00	9.20E+00	9.75E+00	1.04E+01	1.10E+01	1.18E+01	1.26E+01
d	Ht.Mkt.Res	Uplink	Ht.Dmd.Res	Quantities	4.79E+09	4.94E+09	5.19E+09	5.57E+09	5.99E+09	6.43E+09	6.90E+09	7.40E+09	7.91E+09	8.44E+09
c														
d	Elect.Mkt.Res	Allocator	1	Prices	1.28E+01	1.50E+01	1.66E+01	1.77E+01	1.87E+01	1.99E+01	2.11E+01	2.25E+01	2.40E+01	2.56E+01
d	Elect.Mkt.Res	Uplink	Elect.Dmd.Res	Quantities	2.24E+09	2.30E+09	2.42E+09	2.59E+09	2.78E+09	2.98E+09	3.20E+09	3.43E+09	3.67E+09	3.93E+09

c	Node	Comp	CompID	DataSet	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9	Period 10
d	Elect.Mkt.Res	Uplink	Elect.Ht.Res	Quantities	3.72E+07	3.59E+07	4.02E+07	4.41E+07	4.70E+07	5.07E+07	5.52E+07	5.95E+07	6.49E+07	7.20E+07
c														
d	Pet.Mkt.Trp	Allocator	1	Prices	3.88E+00	4.49E+00	4.92E+00	5.28E+00	5.66E+00	6.07E+00	6.54E+00	7.05E+00	7.63E+00	8.26E+00
d	Pet.Mkt.Trp	Uplink	Avt.Trp	Quantities	1.76E+10	1.89E+10	2.05E+10	2.23E+10	2.42E+10	2.63E+10	2.86E+10	3.10E+10	3.35E+10	3.63E+10
d	Pet.Mkt.Trp	Uplink	Trk.Trp	Quantities	7.41E+09	7.73E+09	8.24E+09	8.84E+09	9.50E+09	1.02E+10	1.09E+10	1.17E+10	1.25E+10	1.34E+10
d	Pet.Mkt.Trp	Uplink	Per.Trp	Quantities	5.15E+09	5.06E+09	5.52E+09	6.03E+09	6.59E+09	7.19E+09	7.83E+09	8.52E+09	9.27E+09	1.01E+10
c														
d	Prod.Mkt.Pet	Allocator	1	Prices	3.88E+00	4.49E+00	4.92E+00	5.28E+00	5.66E+00	6.07E+00	6.54E+00	7.05E+00	7.63E+00	8.26E+00
d	Prod.Mkt.Pet	Uplink	Pet.Mkt.Trp	Quantities	3.02E+10	3.17E+10	3.43E+10	3.72E+10	4.03E+10	4.37E+10	4.73E+10	5.12E+10	5.53E+10	5.97E+10
d	Prod.Mkt.Pet	Uplink	Pet.Ht.Ind	Quantities	1.35E+09	1.30E+09	1.64E+09	1.99E+09	2.07E+09	2.30E+09	2.54E+09	2.80E+09	3.08E+09	3.36E+09
d	Prod.Mkt.Pet	Uplink	Pet.Ht.Comm	Quantities	1.00E+09	1.29E+09	1.58E+09	1.82E+09	1.96E+09	2.17E+09	2.41E+09	2.66E+09	2.93E+09	3.21E+09
d	Prod.Mkt.Pet	Uplink	Pet.Ht.Res	Quantities	9.24E+08	7.83E+08	9.04E+08	1.13E+09	1.20E+09	1.37E+09	1.57E+09	1.77E+09	2.01E+09	2.30E+09
d	Prod.Mkt.Pet	Uplink	Oil.Blr.Elect	Quantities	7.87E+08	1.08E+09	1.35E+09	1.64E+09	1.74E+09	1.94E+09	2.14E+09	2.36E+09	2.60E+09	2.83E+09
d	Prod.Mkt.Pet	Uplink	Dist.Trb.Elect	Quantities	5.40E+08	6.43E+08	7.58E+08	8.73E+08	8.93E+08	9.56E+08	1.02E+09	1.08E+09	1.14E+09	1.19E+09
c														
d	Crude.Mkt.Pet	Allocator	1	Prices	2.42E+00	3.00E+00	3.40E+00	3.75E+00	4.11E+00	4.51E+00	4.95E+00	5.44E+00	5.99E+00	6.59E+00
d	Crude.Mkt.Pet	Uplink	Refin.Pet	Quantities	3.65E+10	3.86E+10	4.25E+10	4.68E+10	5.06E+10	5.51E+10	5.99E+10	6.50E+10	7.04E+10	7.63E+10
c														
d	Mkt1.Gas	Allocator	1	Prices	1.61E+00	2.23E+00	2.72E+00	3.00E+00	3.32E+00	3.66E+00	4.02E+00	4.43E+00	4.88E+00	5.39E+00
d	Mkt1.Gas	Uplink	Tax.Gas	Quantities	4.01E+10	3.73E+10	3.64E+10	3.72E+10	4.07E+10	4.32E+10	4.57E+10	4.84E+10	5.11E+10	5.39E+10
c														
d	Mkt2.Gas	Allocator	1	Prices	1.87E+00	2.49E+00	2.98E+00	3.26E+00	3.58E+00	3.92E+00	4.28E+00	4.69E+00	5.14E+00	5.65E+00
d	Mkt2.Gas	Uplink	Gas.Ht.Ind	Quantities	1.70E+10	1.45E+10	1.36E+10	1.36E+10	1.49E+10	1.57E+10	1.65E+10	1.74E+10	1.82E+10	1.91E+10
d	Mkt2.Gas	Uplink	Gas.Cogen.Ind	Quantities	1.16E+09	1.01E+09	1.08E+09	1.28E+09	1.54E+09	1.83E+09	2.16E+09	2.53E+09	2.95E+09	3.43E+09
d	Mkt2.Gas	Uplink	Gas.Ht.Comm	Quantities	2.28E+09	2.66E+09	2.97E+09	3.22E+09	3.61E+09	3.96E+09	4.35E+09	4.76E+09	5.19E+09	5.65E+09
d	Mkt2.Gas	Uplink	Gas.Ht.Res	Quantities	5.94E+09	6.35E+09	6.57E+09	6.84E+09	7.38E+09	7.82E+09	8.24E+09	8.72E+09	9.17E+09	9.56E+09
d	Mkt2.Gas	Uplink	Gas.Blr.Elect	Quantities	7.84E+09	7.55E+09	7.38E+09	7.57E+09	8.37E+09	8.89E+09	9.37E+09	9.87E+09	1.04E+10	1.09E+10
d	Mkt2.Gas	Uplink	Gas.Trb.Elect	Quantities	5.94E+09	5.24E+09	4.81E+09	4.68E+09	4.89E+09	4.94E+09	5.02E+09	5.10E+09	5.18E+09	5.25E+09
c														
d	Mkt.Coal	Allocator	1	Prices	1.76E+00	1.97E+00	2.26E+00	2.53E+00	2.76E+00	3.04E+00	3.35E+00	3.70E+00	4.07E+00	4.48E+00
d	Mkt.Coal	Uplink	Coal.Ht.Ind	Quantities	5.19E+09	9.07E+09	1.14E+10	1.31E+10	1.39E+10	1.52E+10	1.67E+10	1.83E+10	2.01E+10	2.20E+10
d	Mkt.Coal	Uplink	Coal.Ht.Comm	Quantities	4.62E+08	2.69E+08	1.49E+08	1.83E+08	2.13E+08	2.55E+08	3.05E+08	3.67E+08	4.44E+08	5.37E+08
d	Mkt.Coal	Uplink	Coal.Ht.Res	Quantities	2.17E+07	1.89E+07	1.44E+07	1.86E+07	2.26E+07	2.93E+07	3.86E+07	5.03E+07	6.62E+07	8.88E+07
d	Mkt.Coal	Uplink	Coal.Blr.Elect	Quantities	3.46E+09	4.30E+09	5.45E+09	6.38E+09	6.89E+09	7.63E+09	8.42E+09	9.32E+09	1.03E+10	1.14E+10
d	Accum.Elect	na	na	Prices	1.28E+01	1.50E+01	1.66E+01	1.77E+01	1.87E+01	1.99E+01	2.11E+01	2.25E+01	2.40E+01	2.56E+01
d	Accum.Elect	Uplink	Elect.Mkt.Ind	Quantities	1.11E+09	1.16E+09	1.25E+09	1.35E+09	1.47E+09	1.59E+09	1.72E+09	1.87E+09	2.02E+09	2.19E+09
d	Accum.Elect	Uplink	Elect.Mkt.Comm	Quantities	2.82E+09	2.86E+09	3.01E+09	3.20E+09	3.45E+09	3.69E+09	3.96E+09	4.25E+09	4.56E+09	4.89E+09
d	Accum.Elect	Uplink	Elect.Mkt.Res	Quantities	2.28E+09	2.33E+09	2.46E+09	2.63E+09	2.83E+09	3.03E+09	3.25E+09	3.49E+09	3.73E+09	4.00E+09
d	Accum.Elect	Uplink	Gas.Cogen.Ind	Quantities	-6.76E+07	-5.88E+07	-6.29E+07	-7.46E+07	-8.96E+07	-1.06E+08	-1.25E+08	-1.47E+08	-1.72E+08	-1.99E+08

c	Node	Comp	CompID	DataSet	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9	Period 10
d	Accum.Elect	Uplink	Enrich.Nuc	Quantities	7.01E+07	7.39E+07	7.24E+07	6.57E+07	7.50E+07	9.64E+07	1.30E+08	1.63E+08	2.00E+08	2.51E+08
c														
d	Mkt.Elect	Allocator	1	Prices	1.07E+01	1.28E+01	1.42E+01	1.51E+01	1.60E+01	1.71E+01	1.82E+01	1.95E+01	2.08E+01	2.23E+01
d	Mkt.Elect	Allocator	2	Prices	1.53E+01	1.76E+01	1.93E+01	2.05E+01	2.16E+01	2.28E+01	2.41E+01	2.55E+01	2.70E+01	2.86E+01
d	Mkt.Elect	Allocator	3	Prices	2.06E+01	2.38E+01	2.65E+01	2.87E+01	3.06E+01	3.27E+01	3.50E+01	3.74E+01	4.01E+01	4.29E+01
d	Mkt.Elect	Uplink	Accum.Elect	Quantities	3.72E+09	3.82E+09	4.03E+09	4.30E+09	4.64E+09	4.98E+09	5.36E+09	5.77E+09	6.21E+09	6.68E+09
d	Pet.Ht.Ind	TBlock	1	Quantities	1.10E+09	1.06E+09	1.33E+09	1.61E+09	1.68E+09	1.87E+09	2.07E+09	2.28E+09	2.50E+09	2.73E+09
d	Pet.Ht.Ind	TBlock	1	Capacities	1.31E+09	1.06E+09	1.33E+09	1.61E+09	1.68E+09	1.87E+09	2.07E+09	2.28E+09	2.50E+09	2.73E+09
d	Pet.Ht.Ind	TBlock	1	CapacityAddition	0.00E+00	7.96E+06	5.40E+08	5.43E+08	3.28E+08	4.48E+08	2.02E+08	2.19E+08	7.61E+08	7.76E+08
d	Pet.Ht.Ind	TBlock	1	Prices	5.57E+00	6.68E+00	7.21E+00	7.66E+00	8.12E+00	8.63E+00	9.20E+00	9.84E+00	1.05E+01	1.13E+01
d	Pet.Ht.Ind	TBlock	1	ConstraintPrices	5.57E+00	6.68E+00	7.21E+00	7.66E+00	8.12E+00	8.63E+00	9.20E+00	9.84E+00	1.05E+01	1.13E+01
d	Gas.Ht.Ind	TBlock	1	Quantities	1.38E+10	1.18E+10	1.10E+10	1.10E+10	1.21E+10	1.28E+10	1.35E+10	1.41E+10	1.48E+10	1.55E+10
d	Gas.Ht.Ind	TBlock	1	Capacities	1.53E+10	1.31E+10	1.23E+10	1.23E+10	1.35E+10	1.42E+10	1.50E+10	1.57E+10	1.65E+10	1.73E+10
d	Gas.Ht.Ind	TBlock	1	CapacityAddition	1.62E+09	5.49E+08	1.88E+09	2.72E+09	3.97E+09	3.48E+09	2.35E+09	1.30E+09	2.65E+09	3.51E+09
d	Gas.Ht.Ind	TBlock	1	Prices	3.36E+00	4.12E+00	4.72E+00	5.07E+00	5.46E+00	5.87E+00	6.33E+00	6.83E+00	7.39E+00	8.00E+00
d	Gas.Ht.Ind	TBlock	1	ConstraintPrices	3.36E+00	4.12E+00	4.72E+00	5.22E+00	5.47E+00	5.87E+00	6.33E+00	6.83E+00	7.39E+00	8.00E+00
d	Coal.Ht.Ind	TBlock	1	Quantities	3.96E+09	6.92E+09	8.69E+09	1.00E+10	1.06E+10	1.16E+10	1.28E+10	1.40E+10	1.54E+10	1.68E+10
d	Coal.Ht.Ind	TBlock	1	Capacities	3.96E+09	6.92E+09	8.69E+09	1.00E+10	1.06E+10	1.16E+10	1.28E+10	1.40E+10	1.54E+10	1.68E+10
d	Coal.Ht.Ind	TBlock	1	CapacityAddition	7.95E+07	3.74E+09	2.55E+09	2.10E+09	1.38E+09	1.79E+09	1.22E+09	4.98E+09	3.89E+09	3.57E+09
d	Coal.Ht.Ind	TBlock	1	Prices	4.31E+00	4.59E+00	4.95E+00	5.32E+00	5.62E+00	5.99E+00	6.39E+00	6.84E+00	7.33E+00	7.88E+00
d	Coal.Ht.Ind	TBlock	1	ConstraintPrices	4.31E+00	4.59E+00	4.95E+00	5.32E+00	5.62E+00	5.99E+00	6.39E+00	6.84E+00	7.33E+00	7.88E+00
d	Gas.Cogen.Ind	TBlock	1	Quantities	6.76E+08	5.88E+08	6.29E+08	7.46E+08	8.96E+08	1.06E+09	1.25E+09	1.47E+09	1.72E+09	1.99E+09
d	Gas.Cogen.Ind	TBlock	1	Capacities	7.53E+08	6.02E+08	6.29E+08	7.46E+08	8.96E+08	1.06E+09	1.25E+09	1.47E+09	1.72E+09	1.99E+09
d	Gas.Cogen.Ind	TBlock	1	CapacityAddition	0.00E+00	0.00E+00	1.77E+08	2.68E+08	3.01E+08	3.18E+08	1.91E+08	2.18E+08	4.23E+08	5.43E+08
d	Gas.Cogen.Ind	TBlock	1	Prices	6.13E+00	7.51E+00	8.38E+00	8.75E+00	9.20E+00	9.66E+00	1.02E+01	1.07E+01	1.14E+01	1.21E+01
d	Gas.Cogen.Ind	TBlock	1	ConstraintPrices	6.13E+00	7.51E+00	8.38E+00	8.94E+00	9.21E+00	9.66E+00	1.02E+01	1.07E+01	1.14E+01	1.21E+01
d	Elect.Ht.Ind	TBlock	1	Quantities	1.44E+07	1.46E+07	1.66E+07	1.89E+07	2.12E+07	2.38E+07	2.69E+07	3.04E+07	3.45E+07	3.91E+07
d	Elect.Ht.Ind	TBlock	1	Capacities	1.83E+07	1.46E+07	1.66E+07	1.89E+07	2.12E+07	2.38E+07	2.69E+07	3.04E+07	3.45E+07	3.91E+07
d	Elect.Ht.Ind	TBlock	1	CapacityAddition	0.00E+00	0.00E+00	5.61E+06	5.92E+06	5.96E+06	6.31E+06	3.08E+06	3.54E+06	9.67E+06	1.06E+07
d	Elect.Ht.Ind	TBlock	1	Prices	1.32E+01	1.57E+01	1.73E+01	1.84E+01	1.95E+01	2.07E+01	2.19E+01	2.33E+01	2.48E+01	2.65E+01
d	Elect.Ht.Ind	TBlock	1	ConstraintPrices	1.32E+01	1.57E+01	1.73E+01	1.87E+01	1.95E+01	2.07E+01	2.19E+01	2.33E+01	2.48E+01	2.65E+01
d	Pet.Ht.Comm	TBlock	1	Quantities	7.41E+08	9.58E+08	1.17E+09	1.35E+09	1.45E+09	1.61E+09	1.78E+09	1.97E+09	2.17E+09	2.38E+09
d	Pet.Ht.Comm	TBlock	1	Capacities	8.23E+08	1.06E+09	1.30E+09	1.49E+09	1.61E+09	1.79E+09	1.98E+09	2.19E+09	2.41E+09	2.64E+09
d	Pet.Ht.Comm	TBlock	1	CapacityAddition	5.54E+07	6.25E+08	6.17E+08	2.53E+08	7.44E+08	7.92E+08	4.45E+08	9.50E+08	1.01E+09	6.82E+08
d	Pet.Ht.Comm	TBlock	1	Prices	1.71E+01	1.79E+01	1.85E+01	1.90E+01	1.95E+01	2.00E+01	2.06E+01	2.13E+01	2.21E+01	2.30E+01
d	Pet.Ht.Comm	TBlock	1	ConstraintPrices	1.71E+01	1.79E+01	1.85E+01	1.90E+01	1.95E+01	2.00E+01	2.06E+01	2.13E+01	2.21E+01	2.30E+01
d	Gas.Ht.Comm	TBlock	1	Quantities	1.36E+09	1.60E+09	1.78E+09	1.93E+09	2.16E+09	2.37E+09	2.60E+09	2.85E+09	3.11E+09	3.39E+09
d	Gas.Ht.Comm	TBlock	1	Capacities	1.36E+09	1.60E+09	1.78E+09	1.93E+09	2.16E+09	2.37E+09	2.60E+09	2.85E+09	3.11E+09	3.39E+09
d	Gas.Ht.Comm	TBlock	1	CapacityAddition	1.74E+08	8.26E+08	7.81E+08	3.23E+08	1.05E+09	9.96E+08	5.52E+08	1.30E+09	1.26E+09	8.29E+08

c	Node	Comp	CompID	DataSet	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9	Period 10
d	Gas.Ht.Comm	TBlock	1	Prices	1.51E+01	1.61E+01	1.70E+01	1.74E+01	1.80E+01	1.85E+01	1.91E+01	1.98E+01	2.06E+01	2.14E+01
d	Gas.Ht.Comm	TBlock	1	ConstraintPrices	1.51E+01	1.61E+01	1.70E+01	1.76E+01	1.80E+01	1.85E+01	1.91E+01	1.98E+01	2.06E+01	2.14E+01
d	Coal.Ht.Comm	TBlock	1	Quantities	3.08E+08	1.79E+08	9.93E+07	1.22E+08	1.42E+08	1.70E+08	2.04E+08	2.45E+08	2.96E+08	3.58E+08
d	Coal.Ht.Comm	TBlock	1	Capacities	4.08E+08	2.04E+08	9.93E+07	1.22E+08	1.42E+08	1.70E+08	2.04E+08	2.45E+08	2.96E+08	3.58E+08
d	Coal.Ht.Comm	TBlock	1	CapacityAddition	0.00E+00	0.00E+00	9.93E+07	2.27E+07	2.02E+07	1.27E+08	5.65E+07	6.15E+07	1.78E+08	1.19E+08
d	Coal.Ht.Comm	TBlock	1	Prices	2.03E+01	2.50E+01	3.02E+01	3.06E+01	3.10E+01	3.14E+01	3.19E+01	3.24E+01	3.29E+01	3.36E+01
d	Coal.Ht.Comm	TBlock	1	ConstraintPrices	2.03E+01	2.50E+01	3.02E+01	3.06E+01	3.10E+01	3.14E+01	3.19E+01	3.24E+01	3.29E+01	3.36E+01
d	Elect.Ht.Comm	TBlock	1	Quantities	6.24E+08	5.55E+08	5.38E+08	5.26E+08	5.41E+08	5.44E+08	5.46E+08	5.50E+08	5.54E+08	5.59E+08
d	Elect.Ht.Comm	TBlock	1	Capacities	6.98E+08	6.16E+08	5.97E+08	5.84E+08	6.01E+08	6.04E+08	6.07E+08	6.11E+08	6.15E+08	6.21E+08
d	Elect.Ht.Comm	TBlock	1	CapacityAddition	0.00E+00	5.80E+07	1.21E+08	1.26E+08	1.57E+08	1.42E+08	3.05E+06	6.18E+07	1.25E+08	1.32E+08
d	Elect.Ht.Comm	TBlock	1	Prices	1.77E+01	1.99E+01	2.16E+01	2.27E+01	2.37E+01	2.49E+01	2.62E+01	2.75E+01	2.91E+01	3.07E+01
d	Elect.Ht.Comm	TBlock	1	ConstraintPrices	1.77E+01	1.99E+01	2.16E+01	2.29E+01	2.37E+01	2.49E+01	2.62E+01	2.75E+01	2.91E+01	3.07E+01
d	Pet.Ht.Res	TBlock	1	Quantities	7.77E+08	6.58E+08	7.60E+08	9.50E+08	1.01E+09	1.15E+09	1.32E+09	1.49E+09	1.69E+09	1.93E+09
d	Pet.Ht.Res	TBlock	1	Capacities	9.05E+08	6.79E+08	7.60E+08	9.50E+08	1.01E+09	1.15E+09	1.32E+09	1.49E+09	1.69E+09	1.93E+09
d	Pet.Ht.Res	TBlock	1	CapacityAddition	0.00E+00	0.00E+00	3.07E+08	4.16E+08	2.88E+08	1.42E+08	1.68E+08	4.77E+08	6.14E+08	5.34E+08
d	Pet.Ht.Res	TBlock	1	Prices	8.24E+00	9.99E+00	1.08E+01	1.13E+01	1.17E+01	1.22E+01	1.27E+01	1.34E+01	1.40E+01	1.48E+01
d	Pet.Ht.Res	TBlock	1	ConstraintPrices	8.24E+00	9.99E+00	1.08E+01	1.13E+01	1.17E+01	1.22E+01	1.27E+01	1.34E+01	1.40E+01	1.48E+01
d	Gas.Ht.Res	TBlock	1	Quantities	3.96E+09	4.23E+09	4.38E+09	4.56E+09	4.92E+09	5.21E+09	5.49E+09	5.81E+09	6.12E+09	6.37E+09
d	Gas.Ht.Res	TBlock	1	Capacities	3.96E+09	4.23E+09	4.38E+09	4.56E+09	4.92E+09	5.21E+09	5.49E+09	5.81E+09	6.12E+09	6.37E+09
d	Gas.Ht.Res	TBlock	1	CapacityAddition	1.12E+08	1.55E+09	1.43E+09	1.46E+09	4.71E+08	1.85E+09	1.72E+09	1.78E+09	7.76E+08	2.10E+09
d	Gas.Ht.Res	TBlock	1	Prices	5.95E+00	6.88E+00	7.62E+00	8.04E+00	8.52E+00	9.02E+00	9.58E+00	1.02E+01	1.09E+01	1.17E+01
d	Gas.Ht.Res	TBlock	1	ConstraintPrices	5.95E+00	6.88E+00	7.62E+00	8.22E+00	8.53E+00	9.02E+00	9.58E+00	1.02E+01	1.09E+01	1.17E+01
d	Coal.Ht.Res	TBlock	1	Quantities	1.45E+07	1.26E+07	9.59E+06	1.24E+07	1.51E+07	1.96E+07	2.57E+07	3.35E+07	4.42E+07	5.92E+07
d	Coal.Ht.Res	TBlock	1	Capacities	2.05E+07	1.54E+07	1.03E+07	1.31E+07	1.59E+07	2.06E+07	2.71E+07	3.53E+07	4.65E+07	6.23E+07
d	Coal.Ht.Res	TBlock	1	CapacityAddition	0.00E+00	0.00E+00	0.00E+00	7.93E+06	7.93E+06	4.73E+06	6.47E+06	8.21E+06	1.91E+07	2.38E+07
d	Coal.Ht.Res	TBlock	1	Prices	1.83E+01	2.20E+01	2.59E+01	2.68E+01	2.71E+01	2.76E+01	2.80E+01	2.85E+01	2.91E+01	2.97E+01
d	Coal.Ht.Res	TBlock	1	ConstraintPrices	1.83E+01	2.20E+01	2.59E+01	2.68E+01	2.71E+01	2.76E+01	2.80E+01	2.85E+01	2.91E+01	2.97E+01
d	Elect.Ht.Res	TBlock	1	Quantities	3.65E+07	3.52E+07	3.94E+07	4.33E+07	4.61E+07	4.97E+07	5.41E+07	5.84E+07	6.36E+07	7.05E+07
d	Elect.Ht.Res	TBlock	1	Capacities	4.31E+07	3.52E+07	3.94E+07	4.33E+07	4.61E+07	4.97E+07	5.41E+07	5.84E+07	6.36E+07	7.05E+07
d	Elect.Ht.Res	TBlock	1	CapacityAddition	0.00E+00	2.90E+06	1.49E+07	1.47E+07	1.35E+07	3.63E+06	7.32E+06	1.92E+07	1.99E+07	2.05E+07
d	Elect.Ht.Res	TBlock	1	Prices	1.52E+01	1.79E+01	1.95E+01	2.07E+01	2.17E+01	2.29E+01	2.41E+01	2.55E+01	2.71E+01	2.87E+01
d	Elect.Ht.Res	TBlock	1	ConstraintPrices	1.52E+01	1.79E+01	1.95E+01	2.09E+01	2.17E+01	2.29E+01	2.41E+01	2.55E+01	2.71E+01	2.87E+01
d	Avt.Trp	TBlock	1	Quantities	4.58E+08	4.91E+08	5.32E+08	5.79E+08	6.29E+08	6.84E+08	7.42E+08	8.04E+08	8.71E+08	9.42E+08
d	Avt.Trp	TBlock	1	Capacities	4.60E+08	4.91E+08	5.32E+08	5.79E+08	6.29E+08	6.84E+08	7.42E+08	8.04E+08	8.71E+08	9.42E+08
d	Avt.Trp	TBlock	1	CapacityAddition	0.00E+00	1.85E+08	1.94E+08	2.00E+08	5.06E+07	2.39E+08	2.53E+08	2.62E+08	1.17E+08	3.10E+08
d	Avt.Trp	TBlock	1	Prices	3.21E+02	3.45E+02	3.61E+02	3.75E+02	3.90E+02	4.06E+02	4.24E+02	4.44E+02	4.66E+02	4.90E+02
d	Avt.Trp	TBlock	1	ConstraintPrices	3.21E+02	3.45E+02	3.61E+02	3.75E+02	3.90E+02	4.06E+02	4.24E+02	4.44E+02	4.66E+02	4.90E+02
d	Trk.Trp	TBlock	1	Quantities	3.92E+09	4.09E+09	4.36E+09	4.68E+09	5.03E+09	5.40E+09	5.79E+09	6.20E+09	6.64E+09	7.10E+09
d	Trk.Trp	TBlock	1	Capacities	3.92E+09	4.09E+09	4.36E+09	4.68E+09	5.03E+09	5.40E+09	5.79E+09	6.20E+09	6.64E+09	7.10E+09

c	Node	Comp	CompID	DataSet	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9	Period 10
d	Trk.Trp	TBlock	1	CapacityAddition	2.25E+07	1.47E+09	1.57E+09	1.62E+09	3.70E+08	1.84E+09	1.96E+09	2.03E+09	8.07E+08	2.30E+09
d	Trk.Trp	TBlock	1	Prices	7.48E+00	8.62E+00	9.43E+00	1.01E+01	1.08E+01	1.16E+01	1.25E+01	1.35E+01	1.46E+01	1.58E+01
d	Trk.Trp	TBlock	1	ConstraintPrices	7.48E+00	8.62E+00	9.43E+00	1.01E+01	1.08E+01	1.16E+01	1.25E+01	1.35E+01	1.46E+01	1.58E+01
d	Per.Trp	TBlock	1	Quantities	1.34E+09	1.31E+09	1.43E+09	1.57E+09	1.71E+09	1.87E+09	2.03E+09	2.21E+09	2.41E+09	2.61E+09
d	Per.Trp	TBlock	1	Capacities	1.40E+09	1.31E+09	1.43E+09	1.57E+09	1.71E+09	1.87E+09	2.03E+09	2.21E+09	2.41E+09	2.61E+09
d	Per.Trp	TBlock	1	CapacityAddition	0.00E+00	6.15E+08	8.19E+08	1.33E+08	7.59E+08	9.75E+08	3.00E+08	9.39E+08	1.17E+09	5.06E+08
d	Per.Trp	TBlock	1	Prices	1.27E+02	1.41E+02	1.43E+02	1.45E+02	1.46E+02	1.48E+02	1.49E+02	1.51E+02	1.54E+02	1.56E+02
d	Per.Trp	TBlock	1	ConstraintPrices	1.27E+02	1.41E+02	1.43E+02	1.45E+02	1.46E+02	1.48E+02	1.49E+02	1.51E+02	1.54E+02	1.56E+02
d	Refin.Pet	TBlock	1	Quantities	3.48E+10	3.68E+10	4.05E+10	4.46E+10	4.82E+10	5.24E+10	5.70E+10	6.19E+10	6.71E+10	7.27E+10
d	Refin.Pet	TBlock	1	Capacities	3.66E+10	3.87E+10	4.26E+10	4.69E+10	5.07E+10	5.52E+10	6.00E+10	6.51E+10	7.06E+10	7.65E+10
d	Refin.Pet	TBlock	1	CapacityAddition	6.20E+08	8.12E+09	9.88E+09	1.03E+10	9.77E+09	1.05E+10	1.08E+10	5.74E+09	1.36E+10	1.57E+10
d	Refin.Pet	TBlock	1	Prices	3.28E+00	3.89E+00	4.32E+00	4.68E+00	5.06E+00	5.47E+00	5.94E+00	6.45E+00	7.03E+00	7.66E+00
d	Refin.Pet	TBlock	1	ConstraintPrices	3.28E+00	3.89E+00	4.32E+00	4.68E+00	5.06E+00	5.47E+00	5.94E+00	6.45E+00	7.03E+00	7.66E+00
d	Tax.Pet	TBlock	1	Quantities	3.48E+10	3.68E+10	4.05E+10	4.46E+10	4.82E+10	5.24E+10	5.70E+10	6.19E+10	6.71E+10	7.27E+10
d	Tax.Pet	TBlock	1	Capacities	3.48E+10	3.68E+10	4.05E+10	4.46E+10	4.82E+10	5.24E+10	5.70E+10	6.19E+10	6.71E+10	7.27E+10
d	Tax.Pet	TBlock	1	CapacityAddition	2.48E+10	1.20E+10	2.85E+10	1.61E+10	3.21E+10	2.04E+10	3.66E+10	2.53E+10	4.18E+10	3.08E+10
d	Tax.Pet	TBlock	1	Prices	3.88E+00	4.49E+00	4.92E+00	5.28E+00	5.66E+00	6.07E+00	6.54E+00	7.05E+00	7.63E+00	8.26E+00
d	Tax.Pet	TBlock	1	ConstraintPrices	3.88E+00	4.49E+00	4.92E+00	5.28E+00	5.66E+00	6.07E+00	6.54E+00	7.05E+00	7.63E+00	8.26E+00
d	Tax.Gas	TBlock	1	Quantities	4.01E+10	3.73E+10	3.64E+10	3.72E+10	4.07E+10	4.32E+10	4.57E+10	4.84E+10	5.11E+10	5.39E+10
d	Tax.Gas	TBlock	1	Capacities	4.01E+10	3.73E+10	3.64E+10	3.72E+10	4.07E+10	4.32E+10	4.57E+10	4.84E+10	5.11E+10	5.39E+10
d	Tax.Gas	TBlock	1	CapacityAddition	3.01E+10	7.22E+09	2.92E+10	7.99E+09	3.27E+10	1.05E+10	3.52E+10	1.31E+10	3.80E+10	1.59E+10
d	Tax.Gas	TBlock	1	Prices	1.87E+00	2.49E+00	2.98E+00	3.26E+00	3.58E+00	3.92E+00	4.28E+00	4.69E+00	5.14E+00	5.65E+00
d	Tax.Gas	TBlock	1	ConstraintPrices	1.87E+00	2.49E+00	2.98E+00	3.38E+00	3.59E+00	3.92E+00	4.28E+00	4.69E+00	5.14E+00	5.65E+00
d	Tax.Coal	TBlock	1	Quantities	9.13E+09	1.37E+10	1.70E+10	1.97E+10	2.10E+10	2.32E+10	2.55E+10	2.81E+10	3.09E+10	3.40E+10
d	Tax.Coal	TBlock	1	Capacities	1.00E+10	1.37E+10	1.70E+10	1.97E+10	2.10E+10	2.32E+10	2.55E+10	2.81E+10	3.09E+10	3.40E+10
d	Tax.Coal	TBlock	1	CapacityAddition	0.00E+00	1.37E+10	3.33E+09	1.64E+10	4.66E+09	1.85E+10	7.01E+09	2.11E+10	9.85E+09	2.42E+10
d	Tax.Coal	TBlock	1	Prices	1.76E+00	1.97E+00	2.26E+00	2.53E+00	2.76E+00	3.04E+00	3.35E+00	3.70E+00	4.07E+00	4.48E+00
d	Tax.Coal	TBlock	1	ConstraintPrices	1.76E+00	1.97E+00	2.26E+00	2.53E+00	2.76E+00	3.04E+00	3.35E+00	3.70E+00	4.07E+00	4.48E+00
d	Tax.Nuc	TBlock	1	Quantities	7.01E+08	7.39E+08	7.24E+08	6.57E+08	7.50E+08	9.64E+08	1.30E+09	1.63E+09	2.00E+09	2.51E+09
d	Tax.Nuc	TBlock	1	Capacities	1.00E+10	7.39E+08	7.39E+08	6.57E+08	7.50E+08	9.64E+08	1.30E+09	1.63E+09	2.00E+09	2.51E+09
d	Tax.Nuc	TBlock	1	CapacityAddition	0.00E+00	7.39E+08	0.00E+00	6.57E+08	9.35E+07	8.71E+08	4.27E+08	1.20E+09	7.95E+08	1.72E+09
d	Tax.Nuc	TBlock	1	Prices	2.91E+00	3.16E+00	3.35E+00	3.46E+00	3.70E+00	3.88E+00	4.01E+00	4.26E+00	4.58E+00	4.88E+00
d	Tax.Nuc	TBlock	1	ConstraintPrices	2.91E+00	3.16E+00	3.35E+00	3.48E+00	3.70E+00	3.88E+00	4.01E+00	4.26E+00	4.58E+00	4.88E+00
d	Tax.Hydro	TBlock	1	Quantities	5.00E+08	5.00E+08	5.00E+08	5.00E+08	5.00E+08	5.00E+08	5.00E+08	5.00E+08	5.00E+08	5.00E+08
d	Tax.Hydro	TBlock	1	Capacities	1.00E+09	9.50E+08	9.00E+08	8.50E+08	8.00E+08	7.50E+08	7.00E+08	6.50E+08	6.00E+08	5.50E+08
d	Tax.Hydro	TBlock	1	CapacityAddition	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
d	Tax.Hydro	TBlock	1	Prices	1.72E+01	1.97E+01	2.18E+01	2.38E+01	2.52E+01	2.71E+01	2.91E+01	3.13E+01	3.37E+01	3.64E+01
d	Tax.Hydro	TBlock	1	ConstraintPrices	1.72E+01	1.97E+01	2.18E+01	2.38E+01	2.52E+01	2.71E+01	2.91E+01	3.13E+01	3.37E+01	3.64E+01
d	Reg.Gas	TBlock	1	Quantities	1.79E+10	1.30E+10	9.33E+09	5.11E+09	3.30E+07	3.79E+05	1.70E+02	0.00E+00	0.00E+00	1.00E-03

c	Node	Comp	CompID	DataSet	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9	Period 10
d	Reg.Gas	TBlock	1	Capacities	1.00E+11	1.30E+10	1.30E+10	5.11E+09	5.11E+09	3.79E+05	3.79E+05	0.00E+00	0.00E+00	1.00E-03
d	Reg.Gas	TBlock	1	CapacityAddition	0.00E+00	1.30E+10	0.00E+00	5.11E+09	0.00E+00	3.79E+05	0.00E+00	0.00E+00	0.00E+00	1.00E-03
d	Reg.Gas	TBlock	1	Prices	1.56E+00	2.21E+00	2.88E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
d	Reg.Gas	TBlock	1	ConstraintPrices	1.56E+00	2.21E+00	2.88E+00	3.86E+00	1.25E+01	3.44E+01	1.80E+02	4.56E+03	2.58E+04	2.83E+03
d	Enrich.Nuc	TBlock	1	Quantities	7.01E+08	7.39E+08	7.24E+08	6.57E+08	7.50E+08	9.64E+08	1.30E+09	1.63E+09	2.00E+09	2.51E+09
d	Enrich.Nuc	TBlock	1	Capacities	3.29E+09	2.74E+09	2.19E+09	1.64E+09	1.10E+09	1.21E+09	1.62E+09	2.04E+09	2.50E+09	3.14E+09
d	Enrich.Nuc	TBlock	1	CapacityAddition	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.57E+08	9.65E+08	4.17E+08	4.61E+08	6.40E+08
d	Enrich.Nuc	TBlock	1	Prices	2.65E+00	2.90E+00	3.09E+00	3.20E+00	3.44E+00	3.62E+00	3.75E+00	4.00E+00	4.32E+00	4.62E+00
d	Enrich.Nuc	TBlock	1	ConstraintPrices	2.65E+00	2.90E+00	3.09E+00	3.22E+00	3.44E+00	3.62E+00	3.75E+00	4.00E+00	4.32E+00	4.62E+00
d	Oil.Blr.Elect	TBlock	1	Quantities	1.71E+08	2.42E+08	3.03E+08	3.65E+08	3.82E+08	4.22E+08	4.62E+08	5.05E+08	5.51E+08	5.96E+08
d	Oil.Blr.Elect	TBlock	1	Capacities	2.14E+08	3.02E+08	3.78E+08	4.57E+08	4.78E+08	5.27E+08	5.78E+08	6.32E+08	6.88E+08	7.45E+08
d	Oil.Blr.Elect	TBlock	1	CapacityAddition	3.01E+07	1.34E+08	1.22E+08	1.25E+08	6.72E+07	7.91E+07	1.85E+08	1.76E+08	1.81E+08	1.24E+08
d	Oil.Blr.Elect	TBlock	1	Prices	1.48E+01	1.64E+01	1.76E+01	1.85E+01	1.96E+01	2.07E+01	2.19E+01	2.33E+01	2.49E+01	2.66E+01
d	Oil.Blr.Elect	TBlock	1	ConstraintPrices	1.48E+01	1.64E+01	1.76E+01	1.85E+01	1.96E+01	2.07E+01	2.19E+01	2.33E+01	2.49E+01	2.66E+01
d	Oil.Blr.Elect	TBlock	2	Quantities	1.18E+08	1.54E+08	1.94E+08	2.35E+08	2.53E+08	2.85E+08	3.18E+08	3.53E+08	3.90E+08	4.29E+08
d	Oil.Blr.Elect	TBlock	2	Capacities	3.69E+08	4.80E+08	6.07E+08	7.34E+08	7.91E+08	8.91E+08	9.95E+08	1.10E+09	1.22E+09	1.34E+09
d	Oil.Blr.Elect	TBlock	2	CapacityAddition	2.36E+08	1.44E+08	1.60E+08	1.60E+08	9.07E+07	3.36E+08	2.48E+08	2.70E+08	2.75E+08	2.11E+08
d	Oil.Blr.Elect	TBlock	2	Prices	2.01E+01	2.17E+01	2.29E+01	2.38E+01	2.49E+01	2.60E+01	2.72E+01	2.86E+01	3.02E+01	3.19E+01
d	Oil.Blr.Elect	TBlock	2	ConstraintPrices	2.01E+01	2.17E+01	2.29E+01	2.38E+01	2.49E+01	2.60E+01	2.72E+01	2.86E+01	3.02E+01	3.19E+01
d	Oil.Blr.Elect	TBlock	3	Quantities	2.00E+06	3.21E+06	4.69E+06	6.46E+06	7.95E+06	1.01E+07	1.29E+07	1.62E+07	2.03E+07	2.52E+07
d	Oil.Blr.Elect	TBlock	3	Capacities	2.50E+07	4.01E+07	5.86E+07	8.07E+07	9.93E+07	1.27E+08	1.61E+08	2.03E+08	2.54E+08	3.15E+08
d	Oil.Blr.Elect	TBlock	3	CapacityAddition	1.88E+07	1.66E+07	2.01E+07	2.37E+07	2.01E+07	4.63E+07	5.09E+07	6.20E+07	7.45E+07	8.14E+07
d	Oil.Blr.Elect	TBlock	3	Prices	4.66E+01	4.82E+01	4.94E+01	5.04E+01	5.14E+01	5.25E+01	5.37E+01	5.51E+01	5.67E+01	5.84E+01
d	Oil.Blr.Elect	TBlock	3	ConstraintPrices	4.66E+01	4.82E+01	4.94E+01	5.04E+01	5.14E+01	5.25E+01	5.37E+01	5.51E+01	5.67E+01	5.84E+01
d	Dist.Trb.Elect	TBlock	1	Quantities	3.85E+07	5.03E+07	6.02E+07	7.04E+07	7.14E+07	7.64E+07	8.13E+07	8.63E+07	9.13E+07	9.62E+07
d	Dist.Trb.Elect	TBlock	1	Capacities	4.18E+07	5.47E+07	6.55E+07	7.65E+07	7.76E+07	8.30E+07	8.83E+07	9.38E+07	9.93E+07	1.05E+08
d	Dist.Trb.Elect	TBlock	1	CapacityAddition	8.83E+06	1.84E+07	1.63E+07	1.65E+07	6.62E+06	1.09E+07	1.08E+07	1.43E+07	2.39E+07	2.15E+07
d	Dist.Trb.Elect	TBlock	1	Prices	1.99E+01	2.24E+01	2.42E+01	2.58E+01	2.74E+01	2.91E+01	3.11E+01	3.32E+01	3.56E+01	3.83E+01
d	Dist.Trb.Elect	TBlock	1	ConstraintPrices	1.99E+01	2.24E+01	2.42E+01	2.58E+01	2.74E+01	2.91E+01	3.11E+01	3.32E+01	3.56E+01	3.83E+01
d	Dist.Trb.Elect	TBlock	2	Quantities	7.66E+07	8.51E+07	9.78E+07	1.10E+08	1.11E+08	1.16E+08	1.21E+08	1.26E+08	1.30E+08	1.33E+08
d	Dist.Trb.Elect	TBlock	2	Capacities	2.08E+08	2.31E+08	2.66E+08	2.98E+08	3.01E+08	3.16E+08	3.30E+08	3.42E+08	3.53E+08	3.63E+08
d	Dist.Trb.Elect	TBlock	2	CapacityAddition	1.45E+08	3.36E+07	4.48E+07	4.33E+07	1.26E+07	2.62E+07	2.40E+07	1.57E+08	4.46E+07	5.46E+07
d	Dist.Trb.Elect	TBlock	2	Prices	2.19E+01	2.44E+01	2.62E+01	2.78E+01	2.93E+01	3.11E+01	3.30E+01	3.52E+01	3.76E+01	4.03E+01
d	Dist.Trb.Elect	TBlock	2	ConstraintPrices	2.19E+01	2.44E+01	2.62E+01	2.78E+01	2.93E+01	3.11E+01	3.30E+01	3.52E+01	3.76E+01	4.03E+01
d	Dist.Trb.Elect	TBlock	3	Quantities	1.35E+07	1.76E+07	2.24E+07	2.76E+07	3.06E+07	3.50E+07	3.97E+07	4.44E+07	4.92E+07	5.39E+07
d	Dist.Trb.Elect	TBlock	3	Capacities	1.47E+08	1.91E+08	2.43E+08	3.00E+08	3.32E+08	3.81E+08	4.31E+08	4.83E+08	5.34E+08	5.86E+08
d	Dist.Trb.Elect	TBlock	3	CapacityAddition	1.18E+08	4.85E+07	5.75E+07	6.20E+07	3.69E+07	5.31E+07	5.56E+07	1.69E+08	1.00E+08	1.09E+08
d	Dist.Trb.Elect	TBlock	3	Prices	3.18E+01	3.43E+01	3.61E+01	3.77E+01	3.92E+01	4.10E+01	4.29E+01	4.51E+01	4.75E+01	5.02E+01
d	Dist.Trb.Elect	TBlock	3	ConstraintPrices	3.18E+01	3.43E+01	3.61E+01	3.77E+01	3.92E+01	4.10E+01	4.29E+01	4.51E+01	4.75E+01	5.02E+01

c	Node	Comp	CompID	DataSet	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9	Period 10
d	Gas.Blr.Elect	TBlock	1	Quantities	1.75E+09	1.65E+09	1.56E+09	1.56E+09	1.71E+09	1.79E+09	1.86E+09	1.93E+09	2.00E+09	2.07E+09
d	Gas.Blr.Elect	TBlock	1	Capacities	2.19E+09	2.06E+09	1.94E+09	1.95E+09	2.14E+09	2.24E+09	2.32E+09	2.41E+09	2.51E+09	2.59E+09
d	Gas.Blr.Elect	TBlock	1	CapacityAddition	8.05E+08	1.03E+08	1.16E+08	2.40E+08	4.17E+08	3.26E+08	3.15E+08	8.95E+08	1.96E+08	2.01E+08
d	Gas.Blr.Elect	TBlock	1	Prices	9.28E+00	1.12E+01	1.27E+01	1.35E+01	1.45E+01	1.55E+01	1.66E+01	1.78E+01	1.92E+01	2.07E+01
d	Gas.Blr.Elect	TBlock	1	ConstraintPrices	9.28E+00	1.12E+01	1.27E+01	1.39E+01	1.45E+01	1.55E+01	1.66E+01	1.78E+01	1.92E+01	2.07E+01
d	Gas.Blr.Elect	TBlock	2	Quantities	8.30E+08	8.31E+08	8.63E+08	9.13E+08	1.02E+09	1.11E+09	1.19E+09	1.27E+09	1.36E+09	1.44E+09
d	Gas.Blr.Elect	TBlock	2	Capacities	2.59E+09	2.60E+09	2.70E+09	2.85E+09	3.20E+09	3.47E+09	3.72E+09	3.98E+09	4.24E+09	4.49E+09
d	Gas.Blr.Elect	TBlock	2	CapacityAddition	1.88E+09	1.23E+08	2.17E+08	2.74E+08	4.65E+08	3.90E+08	3.72E+08	2.14E+09	3.81E+08	4.72E+08
d	Gas.Blr.Elect	TBlock	2	Prices	1.36E+01	1.55E+01	1.70E+01	1.78E+01	1.88E+01	1.98E+01	2.09E+01	2.22E+01	2.35E+01	2.50E+01
d	Gas.Blr.Elect	TBlock	2	ConstraintPrices	1.36E+01	1.55E+01	1.70E+01	1.82E+01	1.88E+01	1.98E+01	2.09E+01	2.22E+01	2.35E+01	2.50E+01
d	Gas.Blr.Elect	TBlock	3	Quantities	8.18E+06	1.20E+07	1.62E+07	2.11E+07	2.66E+07	3.35E+07	4.18E+07	5.19E+07	6.39E+07	7.80E+07
d	Gas.Blr.Elect	TBlock	3	Capacities	1.02E+08	1.50E+08	2.03E+08	2.64E+08	3.32E+08	4.18E+08	5.23E+08	6.49E+08	7.98E+08	9.75E+08
d	Gas.Blr.Elect	TBlock	3	CapacityAddition	7.95E+07	5.12E+07	5.68E+07	6.51E+07	7.17E+07	9.00E+07	1.09E+08	2.05E+08	2.01E+08	2.34E+08
d	Gas.Blr.Elect	TBlock	3	Prices	3.51E+01	3.70E+01	3.85E+01	3.94E+01	4.03E+01	4.14E+01	4.25E+01	4.37E+01	4.51E+01	4.66E+01
d	Gas.Blr.Elect	TBlock	3	ConstraintPrices	3.51E+01	3.70E+01	3.85E+01	3.97E+01	4.04E+01	4.14E+01	4.25E+01	4.37E+01	4.51E+01	4.66E+01
d	Gas.Trb.Elect	TBlock	1	Quantities	5.51E+08	4.73E+08	4.13E+08	3.97E+08	4.26E+08	4.33E+08	4.37E+08	4.43E+08	4.48E+08	4.53E+08
d	Gas.Trb.Elect	TBlock	1	Capacities	5.99E+08	5.25E+08	4.51E+08	4.31E+08	4.63E+08	4.71E+08	4.75E+08	4.81E+08	4.87E+08	4.92E+08
d	Gas.Trb.Elect	TBlock	1	CapacityAddition	1.54E+08	0.00E+00	0.00E+00	5.42E+07	1.06E+08	8.17E+07	7.89E+07	1.60E+08	6.24E+06	4.81E+06
d	Gas.Trb.Elect	TBlock	1	Prices	1.17E+01	1.43E+01	1.65E+01	1.77E+01	1.91E+01	2.06E+01	2.22E+01	2.40E+01	2.59E+01	2.81E+01
d	Gas.Trb.Elect	TBlock	1	ConstraintPrices	1.17E+01	1.43E+01	1.65E+01	1.82E+01	1.91E+01	2.06E+01	2.22E+01	2.40E+01	2.59E+01	2.81E+01
d	Gas.Trb.Elect	TBlock	2	Quantities	7.49E+08	6.61E+08	6.14E+08	5.92E+08	5.94E+08	5.85E+08	5.85E+08	5.84E+08	5.82E+08	5.81E+08
d	Gas.Trb.Elect	TBlock	2	Capacities	2.04E+09	1.94E+09	1.84E+09	1.75E+09	1.65E+09	1.59E+09	1.59E+09	1.59E+09	1.59E+09	1.59E+09
d	Gas.Trb.Elect	TBlock	2	CapacityAddition	1.46E+09	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.39E+07	9.76E+07	1.45E+09	0.00E+00	0.00E+00
d	Gas.Trb.Elect	TBlock	2	Prices	1.39E+01	1.62E+01	1.82E+01	1.93E+01	2.09E+01	2.25E+01	2.41E+01	2.59E+01	2.79E+01	3.00E+01
d	Gas.Trb.Elect	TBlock	2	ConstraintPrices	1.39E+01	1.62E+01	1.82E+01	1.98E+01	2.10E+01	2.25E+01	2.41E+01	2.59E+01	2.79E+01	3.00E+01
d	Gas.Trb.Elect	TBlock	3	Quantities	6.42E+07	7.04E+07	7.83E+07	8.81E+07	1.03E+08	1.17E+08	1.31E+08	1.45E+08	1.60E+08	1.73E+08
d	Gas.Trb.Elect	TBlock	3	Capacities	6.98E+08	7.65E+08	8.51E+08	9.58E+08	1.12E+09	1.27E+09	1.43E+09	1.58E+09	1.73E+09	1.89E+09
d	Gas.Trb.Elect	TBlock	3	CapacityAddition	5.72E+08	8.81E+07	1.06E+08	1.28E+08	1.86E+08	1.71E+08	1.74E+08	7.27E+08	2.42E+08	2.58E+08
d	Gas.Trb.Elect	TBlock	3	Prices	2.33E+01	2.60E+01	2.81E+01	2.93E+01	3.07E+01	3.22E+01	3.38E+01	3.56E+01	3.75E+01	3.97E+01
d	Gas.Trb.Elect	TBlock	3	ConstraintPrices	2.33E+01	2.60E+01	2.81E+01	2.99E+01	3.08E+01	3.22E+01	3.38E+01	3.56E+01	3.75E+01	3.97E+01
d	Coal.Blr.Elect	TBlock	1	Quantities	9.70E+08	1.14E+09	1.43E+09	1.65E+09	1.77E+09	1.93E+09	2.10E+09	2.29E+09	2.50E+09	2.72E+09
d	Coal.Blr.Elect	TBlock	1	Capacities	1.29E+09	1.34E+09	1.68E+09	1.95E+09	2.08E+09	2.27E+09	2.47E+09	2.70E+09	2.94E+09	3.20E+09
d	Coal.Blr.Elect	TBlock	1	CapacityAddition	0.00E+00	2.68E+08	5.52E+08	4.82E+08	3.45E+08	4.07E+08	4.19E+08	2.26E+08	5.14E+08	8.11E+08
d	Coal.Blr.Elect	TBlock	1	Prices	1.04E+01	1.20E+01	1.29E+01	1.37E+01	1.44E+01	1.53E+01	1.62E+01	1.72E+01	1.84E+01	1.96E+01
d	Coal.Blr.Elect	TBlock	1	ConstraintPrices	1.04E+01	1.20E+01	1.29E+01	1.37E+01	1.44E+01	1.53E+01	1.62E+01	1.72E+01	1.84E+01	1.96E+01
d	Coal.Blr.Elect	TBlock	2	Quantities	1.72E+08	2.78E+08	3.67E+08	4.47E+08	5.04E+08	5.83E+08	6.71E+08	7.71E+08	8.82E+08	1.01E+09
d	Coal.Blr.Elect	TBlock	2	Capacities	5.05E+08	8.17E+08	1.08E+09	1.31E+09	1.48E+09	1.72E+09	1.97E+09	2.27E+09	2.60E+09	2.96E+09
d	Coal.Blr.Elect	TBlock	2	CapacityAddition	8.88E+07	3.81E+08	3.31E+08	3.05E+08	2.38E+08	3.03E+08	3.28E+08	3.81E+08	7.10E+08	7.00E+08
d	Coal.Blr.Elect	TBlock	2	Prices	1.86E+01	1.93E+01	2.01E+01	2.10E+01	2.17E+01	2.25E+01	2.35E+01	2.45E+01	2.56E+01	2.69E+01

c	Node	Comp	CompID	DataSet	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9	Period 10
d	Coal.Blr.Elect	TBlock	2	ConstraintPrices	1.86E+01	1.93E+01	2.01E+01	2.10E+01	2.17E+01	2.25E+01	2.35E+01	2.45E+01	2.56E+01	2.69E+01
d	Coal.Blr.Elect	TBlock	3	Quantities	8.78E+05	1.57E+06	2.41E+06	3.40E+06	4.35E+06	5.75E+06	7.59E+06	9.97E+06	1.30E+07	1.70E+07
d	Coal.Blr.Elect	TBlock	3	Capacities	1.03E+07	1.85E+07	2.83E+07	4.00E+07	5.12E+07	6.76E+07	8.93E+07	1.17E+08	1.54E+08	2.00E+08
d	Coal.Blr.Elect	TBlock	3	CapacityAddition	5.96E+06	8.92E+06	1.05E+07	1.24E+07	1.19E+07	1.72E+07	2.23E+07	3.40E+07	4.51E+07	5.72E+07
d	Coal.Blr.Elect	TBlock	3	Prices	5.49E+01	5.56E+01	5.64E+01	5.73E+01	5.80E+01	5.88E+01	5.98E+01	6.08E+01	6.19E+01	6.32E+01
d	Coal.Blr.Elect	TBlock	3	ConstraintPrices	5.49E+01	5.56E+01	5.64E+01	5.73E+01	5.80E+01	5.88E+01	5.98E+01	6.08E+01	6.19E+01	6.32E+01
d	Nuc.Blr.Elect	TBlock	1	Quantities	1.84E+08	1.97E+08	1.95E+08	1.72E+08	1.95E+08	2.50E+08	3.37E+08	4.22E+08	5.15E+08	6.44E+08
d	Nuc.Blr.Elect	TBlock	1	Capacities	5.61E+08	4.68E+08	3.74E+08	2.81E+08	3.15E+08	4.03E+08	5.44E+08	6.81E+08	8.30E+08	1.04E+09
d	Nuc.Blr.Elect	TBlock	1	CapacityAddition	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.28E+08	1.82E+08	2.34E+08	1.37E+08	1.49E+08	2.08E+08
d	Nuc.Blr.Elect	TBlock	1	Prices	1.46E+01	1.71E+01	1.92E+01	2.15E+01	2.24E+01	2.30E+01	2.34E+01	2.42E+01	2.52E+01	2.62E+01
d	Nuc.Blr.Elect	TBlock	1	ConstraintPrices	1.46E+01	1.71E+01	1.92E+01	2.16E+01	2.24E+01	2.30E+01	2.34E+01	2.42E+01	2.52E+01	2.62E+01
d	Nuc.Blr.Elect	TBlock	2	Quantities	3.31E+07	3.26E+07	2.98E+07	3.14E+07	3.78E+07	4.90E+07	6.55E+07	8.38E+07	1.05E+08	1.35E+08
d	Nuc.Blr.Elect	TBlock	2	Capacities	1.92E+08	1.60E+08	1.28E+08	1.26E+08	-1.52E+08	1.98E+08	2.64E+08	3.38E+08	4.25E+08	5.45E+08
d	Nuc.Blr.Elect	TBlock	2	CapacityAddition	0.00E+00	0.00E+00	0.00E+00	3.05E+07	5.78E+07	7.73E+07	9.86E+07	7.37E+07	8.71E+07	1.20E+08
d	Nuc.Blr.Elect	TBlock	2	Prices	2.59E+01	2.96E+01	3.32E+01	3.56E+01	3.64E+01	3.70E+01	3.74E+01	3.82E+01	3.92E+01	4.02E+01
d	Nuc.Blr.Elect	TBlock	2	ConstraintPrices	2.59E+01	2.96E+01	3.32E+01	3.57E+01	3.64E+01	3.70E+01	3.74E+01	3.82E+01	3.92E+01	4.02E+01
d	Nuc.Blr.Elect	TBlock	3	Quantities	3.11E+05	2.98E+05	2.73E+05	2.30E+05	2.09E+05	2.89E+05	4.05E+05	5.59E+05	7.66E+05	1.06E+06
d	Nuc.Blr.Elect	TBlock	3	Capacities	7.88E+06	6.57E+06	5.25E+06	3.94E+06	3.37E+06	4.67E+06	6.54E+06	9.02E+06	1.23E+07	1.70E+07
d	Nuc.Blr.Elect	TBlock	3	CapacityAddition	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.43E+05	2.61E+06	3.19E+06	2.48E+06	3.33E+06	4.68E+06
d	Nuc.Blr.Elect	TBlock	3	Prices	6.76E+01	7.75E+01	8.72E+01	9.80E+01	1.06E+02	1.07E+02	1.07E+02	1.08E+02	1.09E+02	1.10E+02
d	Nuc.Blr.Elect	TBlock	3	ConstraintPrices	6.76E+01	7.75E+01	8.72E+01	9.81E+01	1.06E+02	1.07E+02	1.07E+02	1.08E+02	1.09E+02	1.10E+02
d	Hydro.Elect	TBlock	1	Quantities	6.07E+07	7.33E+07	7.77E+07	8.11E+07	8.22E+07	8.43E+07	8.66E+07	8.95E+07	9.29E+07	9.63E+07
d	Hydro.Elect	TBlock	1	Capacities	1.05E+08	9.98E+07	9.45E+07	8.93E+07	8.40E+07	8.43E+07	8.66E+07	8.95E+07	9.29E+07	9.63E+07
d	Hydro.Elect	TBlock	1	CapacityAddition	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.51E+06	7.62E+06	8.14E+06	8.58E+06	8.68E+06
d	Hydro.Elect	TBlock	1	Prices	1.82E+01	2.08E+01	2.30E+01	2.51E+01	2.66E+01	2.85E+01	3.07E+01	3.30E+01	3.55E+01	3.83E+01
d	Hydro.Elect	TBlock	1	ConstraintPrices	1.82E+01	2.08E+01	2.30E+01	2.51E+01	2.66E+01	2.85E+01	3.07E+01	3.30E+01	3.55E+01	3.83E+01
d	Hydro.Elect	TBlock	2	Quantities	1.94E+08	1.89E+08	1.87E+08	1.83E+08	1.81E+08	1.78E+08	1.76E+08	1.74E+08	1.73E+08	1.72E+08
d	Hydro.Elect	TBlock	2	Capacities	4.85E+08	4.74E+08	4.67E+08	4.58E+08	4.51E+08	4.46E+08	4.40E+08	4.36E+08	4.32E+08	4.30E+08
d	Hydro.Elect	TBlock	2	CapacityAddition	2.50E+08	0.00E+00	4.75E+06	3.35E+06	4.92E+06	6.18E+06	6.18E+06	7.30E+06	8.20E+06	9.15E+06
d	Hydro.Elect	TBlock	2	Prices	1.82E+01	2.08E+01	2.30E+01	2.51E+01	2.66E+01	2.85E+01	3.07E+01	3.30E+01	3.55E+01	3.83E+01
d	Hydro.Elect	TBlock	2	ConstraintPrices	1.82E+01	2.08E+01	2.30E+01	2.51E+01	2.66E+01	2.85E+01	3.07E+01	3.30E+01	3.55E+01	3.83E+01
d	Hydro.Elect	TBlock	3	Quantities	2.21E+08	2.14E+08	2.12E+08	2.12E+08	2.13E+08	2.14E+08	2.13E+08	2.12E+08	2.10E+08	2.08E+08
d	Hydro.Elect	TBlock	3	Capacities	2.21E+09	2.21E+09	2.20E+09	2.19E+09	2.19E+09	2.18E+09	2.17E+09	2.16E+09	2.16E+09	2.15E+09
d	Hydro.Elect	TBlock	3	CapacityAddition	2.08E+09	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
d	Hydro.Elect	TBlock	3	Prices	1.82E+01	2.08E+01	2.30E+01	2.51E+01	2.66E+01	2.85E+01	3.07E+01	3.30E+01	3.55E+01	3.83E+01
d	Hydro.Elect	TBlock	3	ConstraintPrices	1.82E+01	2.08E+01	2.30E+01	2.51E+01	2.66E+01	2.85E+01	3.07E+01	3.30E+01	3.55E+01	3.83E+01
d	Imp.Pet.Rsc	na	na	Prices	2.63E+00	2.90E+00	3.20E+00	3.54E+00	3.90E+00	4.31E+00	4.76E+00	5.25E+00	5.80E+00	6.40E+00
d	Imp.Pet.Rsc	na	na	Quantities	1.26E+10	2.30E+10	3.26E+10	4.03E+10	4.60E+10	5.15E+10	5.69E+10	6.24E+10	6.81E+10	7.41E+10
d	Dom.Pet.Rsc	na	na	Prices	2.31E+00	3.13E+00	4.07E+00	5.09E+00	6.18E+00	7.36E+00	8.61E+00	9.96E+00	1.14E+01	1.30E+01

c	Node	Comp	CompID	DataSet	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9	Period 10
d	Dom.Pet.Rsc	na	na	Quantities	2.39E+10	1.56E+10	9.86E+09	6.53E+09	4.63E+09	3.56E+09	2.93E+09	2.55E+09	2.32E+09	2.18E+09
d	Dom.Pet.Rsc	na	na	Commitments	2.80E+10	3.68E+10	2.05E+10	1.60E+10	1.37E+10	1.24E+10	1.15E+10	1.09E+10	1.05E+10	1.02E+10
d	Imp.Gas.Rsc	na	na	Prices	2.10E+00	2.32E+00	2.56E+00	2.83E+00	3.12E+00	3.45E+00	3.81E+00	4.20E+00	4.64E+00	5.12E+00
d	Imp.Gas.Rsc	na	na	Quantities	3.98E+09	1.03E+10	1.68E+10	2.41E+10	3.39E+10	3.75E+10	4.06E+10	4.37E+10	4.66E+10	4.94E+10
d	Dom.Gas.Rsc	na	na	Prices	1.55E+00	2.18E+00	2.83E+00	3.53E+00	4.32E+00	5.03E+00	5.78E+00	6.57E+00	7.40E+00	8.29E+00
d	Dom.Gas.Rsc	na	na	Quantities	1.83E+10	1.40E+10	1.02E+10	7.94E+09	6.74E+09	5.66E+09	5.04E+09	4.69E+09	4.52E+09	4.46E+09
d	Dom.Gas.Rsc	na	na	Commitments	1.17E+10	4.91E+10	3.20E+10	2.83E+10	2.77E+10	2.29E+10	2.21E+10	2.17E+10	2.17E+10	2.20E+10
d	Rgltd.Gas.Rsc	na	na	Prices	1.56E+00	2.21E+00	2.88E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
d	Rgltd.Gas.Rsc	na	na	Quantities	1.79E+10	1.30E+10	9.33E+09	5.11E+09	3.30E+07	3.79E+05	1.70E+02	0.00E+00	0.00E+00	1.00E-03
d	Rgltd.Gas.Rsc	na	na	Commitments	8.69E+09	4.07E+10	2.83E+10	4.42E+09	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
d	Coal.Rsc	na	na	Prices	1.50E+00	1.71E+00	2.00E+00	2.27E+00	2.50E+00	2.78E+00	3.09E+00	3.44E+00	3.81E+00	4.22E+00
d	Coal.Rsc	na	na	Quantities	9.13E+09	1.37E+10	1.70E+10	1.97E+10	2.10E+10	2.32E+10	2.55E+10	2.81E+10	3.09E+10	3.40E+10
d	Coal.Rsc	na	na	Commitments	0.00E+00	1.29E+11	1.68E+11	1.66E+11	1.38E+11	1.69E+11	1.86E+11	2.05E+11	2.26E+11	2.48E+11
d	Nuc.Rsc	na	na	Prices	7.40E-01	7.40E-01	7.40E-01	7.40E-01	7.40E-01	7.40E-01	7.40E-01	8.38E-01	9.81E-01	1.10E+00
d	Nuc.Rsc	na	na	Quantities	8.42E+08	8.87E+08	8.69E+08	7.88E+08	9.00E+08	1.16E+09	1.56E+09	1.96E+09	2.40E+09	3.01E+09
d	Nuc.Rsc	na	na	Commitments	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.52E+09	2.31E+10	3.04E+10
d	Hydro.Rsc	na	na	Prices	5.00E-01	5.00E-01	5.00E-01	5.00E-01	5.00E-01	5.00E-01	5.00E-01	5.00E-01	5.00E-01	5.00E-01
d	Hydro.Rsc	na	na	Quantities	5.00E+08	5.00E+08	5.00E+08	5.00E+08	5.00E+08	5.00E+08	5.00E+08	5.00E+08	5.00E+08	5.00E+08

Technical Information Department • Lawrence Livermore National Laboratory
University of California • Livermore, California 94550