

Contents

Contents of the Help file

General

[Introduction](#)

[Getting Started](#)

[ABCs of Modeling](#)

[Using the Mouse](#)

[The Speedbar](#)

[Reading the Screens](#)

[Pictures](#)

[Creating a Project](#)

[Setting Default Values](#)

[Clones](#)

[Compound Nodes](#)

Project planning

[Critical Path](#)

[Task timeline](#)

Building blocks

[Five Node Types](#)

[Three Connector Types](#)

[Resources](#)

Running a model

[Scheduled Activity](#)

[Simulation Control Parameters](#)

Technical Reference

[Basic Concepts](#)

[Node Parameters](#)

[Next Node Choices](#)

[Connections](#)

[How Transactions Move](#)

[The Clock](#)

[Node Statistics](#)

Screen Appearance

[Fonts](#)

[Colors, shadows, etc.](#)

[International](#)

Reporting results

[Exporting Data](#)

[Presentation Graphics](#)

Lookup

[Glossary](#)

Warranty and Disclaimer

This software is sold on an as-is basis. The author (Neptune Systems) specifically disclaims all warranties, expressed or implied. In no event shall the author be liable for any loss of profit or any other damage including but not limited to special, incidental, consequential or other damages including damages to a third party. By using this software you are agreeing to these terms. If these terms are not

agreeable, do not use this software.

Copyright

Copyright© 1995, Neptune Systems, All rights reserved

DynaMetrics is a trademark of Neptune Systems

Windows and Paintbrush are trademarks of Microsoft Corporation

Bringing your diagrams to life See Also ▼

Making the pieces play together



Businesses are dynamic enterprises. They have customers, employees, equipment, supplies, and products, all interacting in ways that are complex and difficult to understand. Times are seldom as scheduled. Customers arrive when they please, not when you expect them. Jobs may go faster or slower than you expect. Employees get called for jury duty. Equipment fails.

Static analysis of an enterprise cannot handle random variations in time or sporadic availability of resources. It can't tell how long it will take to recover from a sudden burst of activity. It can't tell how many customers are lost because of excessive waiting time.

You need dynamic analysis to understand the effect of normal variations or unexpected events on your operation.

What is a simulation?

A simulation is a model of how something works. When you sketch a floor plan and examine it for possible traffic congestion, you are doing a simulation in your imagination. When you lay out a work flow on a piece of paper you are simulating your business.

There is nothing mysterious about simulation

You have been doing it all along. Every work flow diagram you have ever seen is a model of an enterprise. The problem with pencil and paper models is that they get harder to understand as they get bigger. You can only carry a small amount of data in your head.

Generating the work flow diagram on a computer may make the changes easier, but it does not tell you what will happen if an employee calls in sick.

What if ...

Suppose you want to know if a piece of equipment is really needed. With a paper simulation changes are messy, and understanding the impact of a change really means starting over. Static analysis may not show that a piece of equipment is nearing saturation. It can't find a workstation that is essentially idle.

Schedules

I once saw a static work flow diagram that showed a committee review, and allowed a half hour for the process. It didn't mention that the committee met on Wednesday afternoon, and that a document that arrived shortly after adjournment would take a week. The diagram said a half hour, but in practice the average time was over three days. Dynamic analysis can find these.

Perhaps the hardest part of studying a static model is remembering that all of the parts are running simultaneously.

Getting Started

The first steps



First, you need to be familiar with the three basic concepts of the model. You can postpone thinking about resource use until later, but you need to understand the graph and a transaction before you start. There is nothing complicated about either the graph or transactions. The graph is merely the map of the work flow. It looks like any work flow diagram. It consists of boxes connected by arrows. The boxes indicate that something is to be done, and the arrows show how the work flows between boxes. Transactions represent pieces of work moving through the graph.

You could, if you like, draw the graph on a piece of paper and use pennies to represent transactions. You may have already done this in your business. You move the pennies around and look for places on the graph where pennies are starting to pile up - bottlenecks. What the pennies represent is entirely up to you.

The whole purpose of a simulation is to see how the work flows through the graph.

Time and space

The simulation deals only with time. This discussion talks about the graph being fixed and the transactions being movable. This has nothing to do with physical location. The graph is fixed with respect to your organization; it shows how you do business. A transaction is something that flows through or is processed by your organization - your input, output, customers, or whatever you please.

For an assembly line it is the plant that has a fixed location and the work that moves. In the construction industry the business moves and the work is in a fixed place. But the models for both enterprises look very much the same.

There is no concept of physical location anywhere in the program.

Study the examples

Before you plunge in and start building, you should look at some of the examples.

Examples

The classic queuing problem

Examples

Look at some examples



Use the Open Existing Project button on the opening screen to start working with the examples. Start with STEP1.DYN

If you are new to simulations, you should look at some of the examples that came with the software. Use the File | Open menu and pick STEP1.DYN. This is the simplest model possible that actually does anything.

Use the Window | Tile horizontal menu item to tile the windows so you can see all of them. Use the Window | Fit to window menu item to resize the graph.

Getting Help

Press the F1 key. This provides help on the active window.

Detailed help on the graph

Move the cursor over one of the blocks on the graph, press the right mouse button and hold it down. You will get a pop-up menu. Move the cursor to the top menu item, "Help on component". Release the right button. This will open the help file at the topic which explains the box.

Close the help file. Repeat the above step for one of the arrows joining two boxes.

Run the sample



Turn on the animation. When this is pressed, connectors will flash whenever a transaction moves. This gives you a quick look at what is going on.



Click the run button at the top of the screen and watch the results. At any time you can pause the run by clicking anywhere.

Look at the statistics. Place the cursor on one of the blocks and use the right button to "show statistics".

The Classic Queuing Problem

The trade off



No discussion of business simulation would be complete without considering this classic problem. You probably run into it several times a week. Everyone is familiar with standing in line at grocery stores and banks. If your bank doesn't do things this way, try the post office or credit union.

There are two basic ways of waiting for service:

- 1 You get in the shortest line and hope. This is how groceries do it.
- 2 Everyone waits in the same line for the next available clerk. This is how banks do it.

CLASSIC.DYN

The file CLASSIC.DYN compares the two ways of forming waiting lines. Transactions (customers) are generated in the start block and copied into two identical lines. The only difference between the two lines is that the one for the bank has a single line waiting for the next available teller, and the one for the grocery has a line for each clerk.



Run the model and watch the progress gauge in the lower right corner of the screen.

Look at the statistics on the two stop blocks. To look at the statistics, place the cursor on one of the stop blocks, press (and hold) the right mouse button and select "Show Statistics" from the pop up menu. Now look at the other stop block. See the difference?

The Right Button is Critical



Building a model using the mouse

Right button

The right button is used to create, edit and delete objects in the graph window.

Never click the right button -- press it and hold it down.

Left button

The left button is used to rearrange objects in the graph window.

Right Button

Press the right button and hold it down

Help on component	
New	New Label ...
Clone	New Start Point
Make this the default	New Process
	New Sync Point
Edit Node ...	New Assembly Point
Edit Label ...	New End Point
Edit Connection ...	New Compound Node
Delete	
Remove Component	New Connection ...
Show Statistics ...	
Show Resource Use...	

The right mouse button is used in the graph window. When you want to get this menu, press the right mouse button and hold it down. Move the cursor to the item you want and release the button. Action on this menu is on button release, not button press. If you want to get out without doing anything, move the cursor out of the menu before you release the button.

Releasing the right button on one of the items to the left causes the indicated action. All items starting with the word "New" create an item at the location of the cursor when the button was first pressed.

All other items require that the cursor be pointing at something when the mouse button is pressed. The action taken will depend on the indicated object.

In almost all windows applications, action is on mouse release, not mouse press. This gives you a way to look at things without causing any action.

Left Button

Using the left mouse button

The left mouse button is used in the graph window. It is used to rearrange items on the screen, to select a label for font modification, and to connect boxes.

To connect two boxes, hold down a shift key while dragging between them in the direction of work flow.

To move a single object on the graph, point to the object and drag it to the new location.

To move an area of the graph, drag a rectangle around the area from top to bottom, then drag the area to the new location. An object is "caught" by the area rectangle if its center is inside the area. Note: If grid snap is turned on, objects may "snap" out of the area rectangle.

To select a label for font modification, click on the label.

Speedbar

Functional groups



Files - Create, Open, Save, Save as



Graph size - Larger, Smaller, Fit to window. See [Node Size](#)



Copy - Active window to Clipboard



Graphics - Presentation Graphics, Set Summary options, Set Chart options, Set table options



Appearance - Fonts, Colors, Grid snap, Hide pictures



Resource pool maintenance



Run Options - Setup, Fix all times, [Animate](#)



Run - Start run, Continue run, Clear statistics, Reset

New Project

Use this dialog to start a new project

The Examples menu item contains several predefined projects you can use to get started. If you like, you can pick one of these and use it as is, or modify it for your purposes.

New Project

Examples

Clock Units

MilliSeconds

Seconds

Minutes

Hours

Days

Minutes in Block

Blocks in period

These values can be changed later if necessary.

Time Line is 32 blocks of 15 Minutes
(8.00 Hours)

Clock Units

This gives the time units used in this model. All times will be given in the units specified here.

Minutes in Block

Used in scheduling block activity. A process is either on or off for an entire block of time. Example: If this is 15 minutes, then a coffee break is one block and a 45 minute lunch is three blocks. A process can take more or less time than this. This is just the scheduling block.

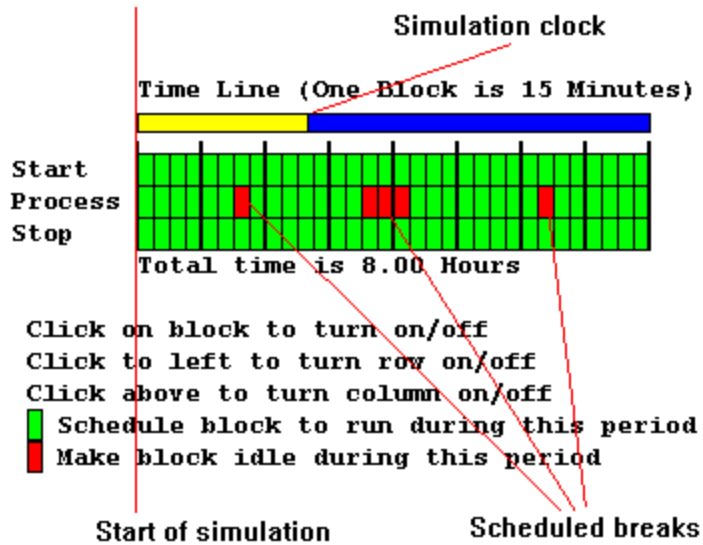
Blocks in period

Number of schedule units in the time line. Example: If Time in block is 15 minutes and you are modeling an eight hour day, there are 32 Blocks in the period. (4 blocks in an hour times 8 hours = 32 blocks)

[See Time Line](#)

Time Line Window

How to schedule block activity



See New Project

The Time Line is defined in the New Project setup.

Nodes

The five node types

Start Point

Generate transactions for the simulation, based on the clock.

Process

The main building block for the model

Sync Point

For work that can proceed as parallel sub tasks. Resynchronize copies of transactions.

Assembly Point

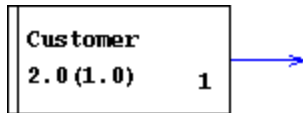
Generate transactions for the simulation, based on events. Often used to build an assembly out of interchangeable parts.

End Point

Remove transactions from the simulation.

Start Point

This is how to start a transaction



A start point generates transactions that will move through the graph. The transactions generated here take on the name of the starting box. They are also stamped with the time of creation. This time stamp will be used later to compute the age of the transaction. The name can be used to route transactions by name.

Count

A start block is activated, that is, generates transactions based on the clock. Each time the block "fires", it generates the number of transactions given in the Count parameter of the Output section of node parameters. This count is the number in the lower right corner of the above box. If the count is preceded by a "P", then the number is the mean of a Poisson distribution.

Timing

The timing numbers for a start block determine the time between successive block activations. In this example, new customers enter the store two minutes apart, give or take a minute.

Priority

Node Output

Cargo

Any transaction can act as a carrier. The cargo capacity of transactions are determined in the Start block.

Loading Dock

General

Most models have several start points. A factory gets parts from several suppliers. A hospital gets normal patients at receiving, and emergency patients, with a higher priority, through the emergency room.

A model must have at least one start point, or it won't do anything.

Node Output

Output Parameters

The screenshot shows a software interface with five tabs: 'Input', 'Processing', 'Output', 'Block Size', and 'Pictures'. The 'Output' tab is active. It contains two main sections:

- Transaction Generation Parameters:**
 - How many:** A sub-section containing a 'Count' field with the value 'N/A' and two radio buttons: 'Fixed count' (selected) and 'Poisson distribution'.
 - Priority:** A field with the value 'N/A'.
 - Min Cargo:** A field with the value 'N/A'.
 - Max Cargo:** A field with the value 'N/A'.
- Next block choice:** A list of radio buttons: 'Shortest waiting line' (selected), 'Wait for an idle node', 'Pick one at random', 'Rotation', 'By name', 'Spill over' (with a small input field containing '0'), and 'Copy All'.

Transaction Generation Parameters

These values are for Start and Assemble blocks only

Count - Number of transactions generated each time the block "fires"

Fixed count - generate the same number every time. For example, you get 12 eggs in a carton.

Poisson distribution - generate a random number of transactions each time. The number will be chosen from the Poisson distribution with a mean of "count". This is an approximation of the number of pieces in your incoming mail on any particular day, or the number of incoming phone calls in an hour.

Priority - Priority of transactions generated at this block. The higher the number, the more important the transaction

Min Cargo - A transaction will not leave a loading dock without at least this many transactions on board.

Max Cargo - Maximum number of transactions that can "piggyback" on this transaction.

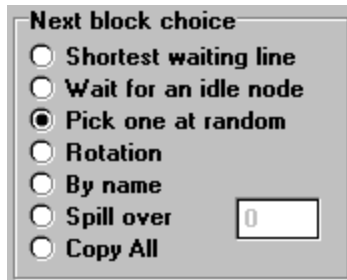
Next block choice

Where to send the transaction if a block has more than one exit.

Choices

Choices

Choice of next node



Next block choice

- Shortest waiting line
- Wait for an idle node
- Pick one at random
- Rotation
- By name
- Spill over
- Copy All

There are several ways to pick the next node if more than one are available. All of these are common in business. Here are some examples of the choices and where they are commonly used.

Shortest waiting line

This is what you find in grocery stores. You get in the shortest line.

Wait for an idle node

This is what you find in some banks and post offices. Everyone waits in the same line for the next available clerk. Many factories also use this method; work goes to the first idle workstation.

Pick one at random

This is how you model a split in the traffic based on experience. You know from watching your business that one customer in ten visits a certain counter, or that one part in a hundred is rejected, or that one check in fifty bounces. You know the ratio, but you don't know in advance which ones.

Rotation

This is like dealing cards at a table, or assigning work on a rotation basis. The weights are used in the same way as in Random, but the distribution is strictly in order.

By name

This sends the transaction to the node that has a name matching the transaction name. If no such block is found, the transaction is sent to the last node. Transactions have the same name as the originating block.

Spill over

This is how to tell when to add a node. Some transactions are perishable. A seafood packing plant has a limited time to get the catch processed.

You are often balancing customer patience with number of employees available. Maybe you need to add another checkout line, or add a phone operator during certain hours. Very few transactions are as perishable as a phone customer on hold.

You model this by adding more services than you think you will need and setting the maximum line length acceptable for a transaction. You then run the model and watch how often the "extra" nodes are active.

Copy All

A copy of the transaction is sent to each following node. Only copies produced in this way can be resynchronized in a Sync block.

By name, Rotation and Spill over - the placement of the nodes on the page is critical. See Node Order.

Node Order
Dispatcher

Dispatcher

How transactions move

The simulator has an internal procedure called the Dispatcher. You will never have to deal with it directly, but you need to know how it operates.



Transactions are moved from node to node by the Dispatcher. If there is only one following block, there is no choice to make. If there are several exits from a block, the dispatch rules are followed. The dispatch rules are imposed in the following order:

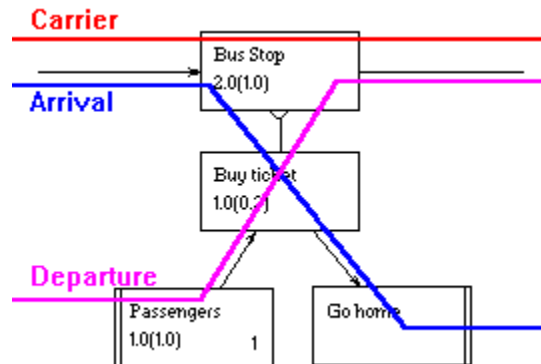
- 1 For a split node, copies of the transaction are sent to all following nodes. The Dispatcher then exits.
- 2 If *By name* is checked: Transactions carry the name of the originating (Start or Assemble) block. NOTE: An Assembly block does NOT change the name of the Master transaction. If a name match between the transaction and one of the following nodes is found, the transaction is moved to that node. If no match is found, the transaction moves to the last node in order. This is the equivalent of the *case* statement in Pascal or the *switch* statement in C. The Dispatcher then exits.
- 3 The Dispatcher chooses the next node according to the method checked. If the choice is for the first idle node, and all following nodes are busy the dispatcher does nothing. If none of the following nodes are accepting input, the Dispatcher does nothing. If the choice is Spill over and all following node queues are full, the Dispatcher does nothing.

The first two moves are unconditional; the 'Always Accept Input' check box has no effect.

Loading Dock Connector

Any transaction can be a carrier

When you check the Load/Unload box in the Connection dialog, the arrow head on the connector changes to a circle to indicate that this is a loading connector. The circle is on the carrier end of the connection. A loading connector is the only bidirectional connector in the simulator.



This shows the transaction path at a loading point. This example is for a bus stop. The flow would be exactly the same for trucks carrying cargo or planes carrying mail or ships carrying cars.

At the dock

When the carrier stops at a loading dock several things happen.

First the arrivals unload. The percent to unload is specified in the connector dialog.

Next, waiting "passengers" get on the carrier, up to the maximum capacity.

The carrier waits for the specified time or the minimum load, whichever takes longer.

Passengers arriving during this period can get on if there is room.

The carrier moves on to the next node.

Any delays, next node choices etc. on the carrier node (the one with the circle) apply to the carrier itself. For a normal queue this is analogous to the line of taxicabs you see at airports. If Can Interrupt is checked this is like an ambulance; the highest priority carrier gets handled first. If Multiple transactions is checked, this means that several carriers can load or unload simultaneously; a bank of elevators is an example.

On the other end of the connector, delays are applied to the passengers before they move to the waiting line for the carrier. The next node choice is applied to arrivals.

Transactions maintain their identity while on a carrier, but are invisible while on board. While on board they play no part in the scheduling. They become visible when they get off the carrier.

For example

If you bundle orders and ship them to a factory you have to unbundle them at the factory to process the orders. If you get a shipment from the factory you have to unpack it before you can match up the contents with the orders.

In general

Any transaction can "piggyback" on any other transaction. A transaction can be a carrier and a passenger

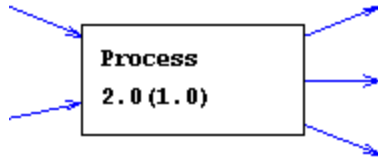
at the same time -- containerized freight. You must keep track of the role being played by a transaction at each place in the model.

Restrictions

You can not split or stop a loaded transaction.

Process

The basic building block



The process is the basic building block of the model. All other blocks play a support role to the process. This is where most of your delays are specified. This is where you have the most choices of parameters. This is where you can specify queuing strategy, interrupt processing and next node choice. The process is the only block that can use resources. The process is the only box that can act as a loading dock.

[Node Details](#)

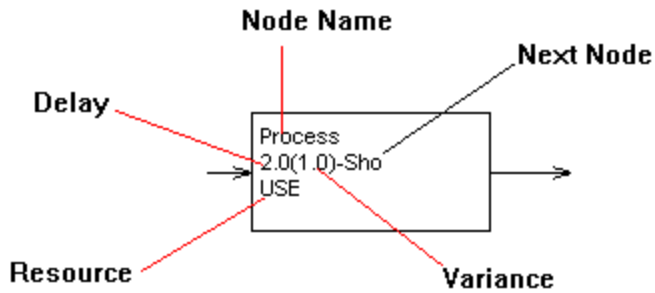
[Clone](#)

Node Details

Two different displays

Display during design

While you are defining the model the nodes show the following information.



In addition, during design the following is shown:

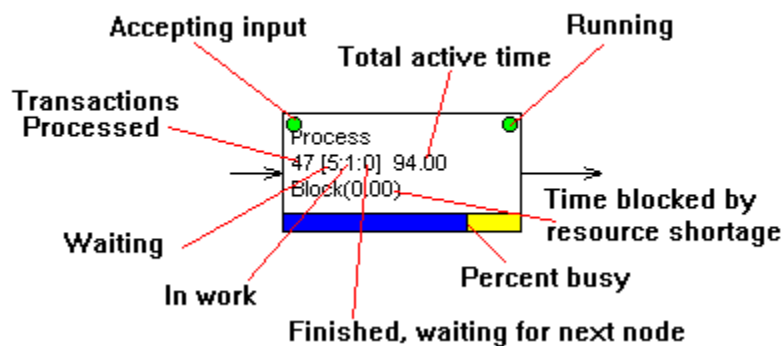
If there is a choice of following nodes, the following will be shown on the line showing the delay:

- "Sho" - Shortest
- "Wai" - Wait for idle node
- "Ran" - Pick one at random
- "Spi" - Spill over
- "Rot" - Rotate
- "Nam" - By name
- 'Cpy" - Send copy to each following node

A "*" on the same line indicates that the process can be interrupted.

Display during run

While the simulation is running, the nodes show the following information.



In addition during a run the following is shown:

There will be a large "X" across the box if there is work to do, but resources are not available.

For a loading dock, the number waiting in line to get on the carrier is shown in braces {}.

[See Pictures](#)

Default Values

You can change the default values

When you define a new node default values are given to all parameters for the node. There is a different set of default values for each node type. These settings may not be appropriate for your purposes. If you find that you are spending too much time editing nodes you may want to change these values.

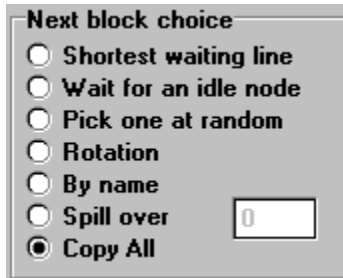
To change the default values for a particular node type, simply point to a node that you want to use as the default and pick Make this the default from the pop up menu you get using the right mouse button. From that time on all new nodes of that type will be copies of the one you selected. The only restriction is that the **default node cannot use resources**.

The default values are for all projects, not just the current one.

Sync Points

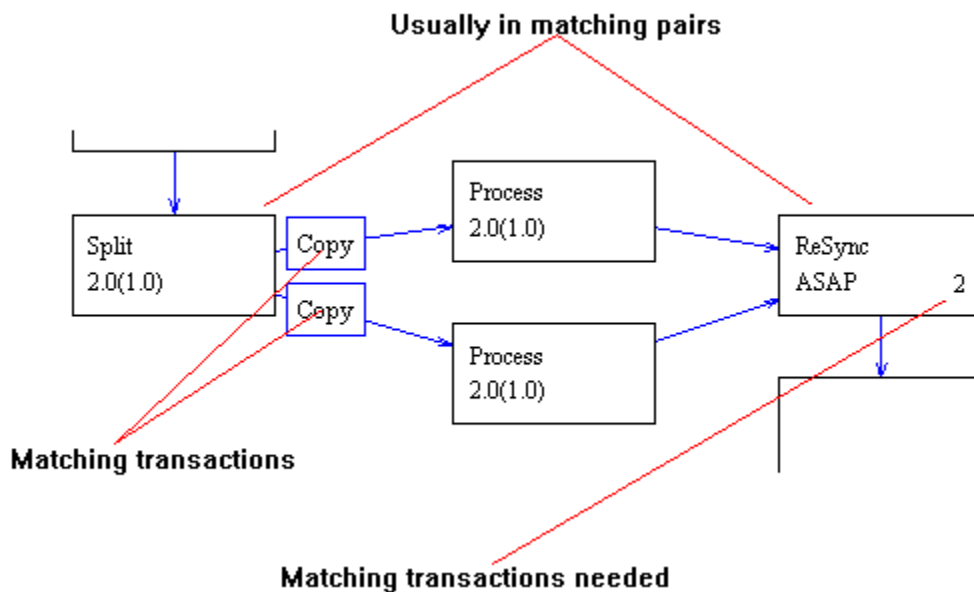
Dividing the work

A Sync point is used to gather multiple copies of a transaction. These copies must have been produced by the Copy All option on a process.



A split point is a process that sends a copy of a transaction to each following node. You create a split point by checking the Copy All button in the Next block choice for a Process. A split point is used to model a place where work breaks into sub tasks that can be run in parallel.

At some place down the line you will probably want to get the sub tasks back together. That is, to resynchronize them.



The Sync block waits until the required number of matching transactions have arrived. When the count is reached, all transactions but one are destroyed, and that one is sent on to the next block.

How it works

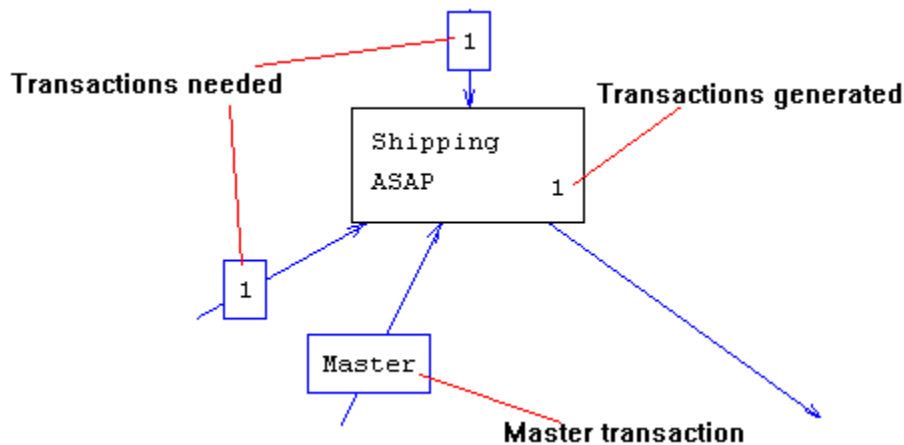
Each transaction generated by either a Start or Assembly block is assigned a unique identifier. A split point makes copies of the transaction, including the unique identifier. This identifier is used in a Sync block to determine which transactions match.

Assembly

Putting parts together

An assembly point waits for a specified number of transactions from one or more inputs and then generates an output. You set the number of transactions needed in the "Parts needed for assembly" box of the connection edit box.

Usually all incoming transactions are destroyed and a new transaction is created. You may want to change this to make one of the incoming transactions the output transaction. This one is called the master transaction. The main reason for doing this is so the output can be used in a synchronization block later on.



Priority and cargo

You can change the priority and cargo capacity of a transaction using Assembly blocks. The Master transaction keeps all of its properties except priority, min cargo and max cargo. These it takes from the Assembly block. NOTE: An Assembly block does NOT change the name of the Master transaction.

Repackaging

Another common use for assembly blocks is in repackaging products. Suppose you receive supplies by the carton and use them one at a time. In that case the input count is one and the transactions generated count is the number of pieces in the carton.

If you are packing your product for shipping, the input count is the number of units per carton, and the output count is one.

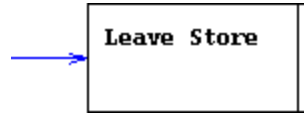
This is another start block

Notice that assembly points have the same transaction count, priority and cargo parameters as a start block.

An assembly block is just a start block that is triggered by events, not the clock.

End Point

This is how to end a transaction



An end point removes transactions from a simulation. It also gathers statistics on how long it took the transaction to work its way through the model. These statistics are valuable if you want to know whether your entire operation is running fast enough.

For example, if transactions represent customers, a very long time may mean that you are losing sales. If transactions represent paperwork, you may find that some forms are taking an unacceptable time to move. If they represent perishable goods, you may be getting spoilage.

Connections

The transaction paths

Connections are the transaction paths that tie the graph together. They appear as arrows pointing in the direction of transaction flow.

Connections do not usually have properties, but there are three cases where they do:

Random Choice and Rotation

When the next node is a random choice, the weight is used to compute the probability.
When the next node is taken in rotation, the weight is the count given that node before moving on to the next. In the display the weights are shown in rectangles for random choice and ellipses for rotation.

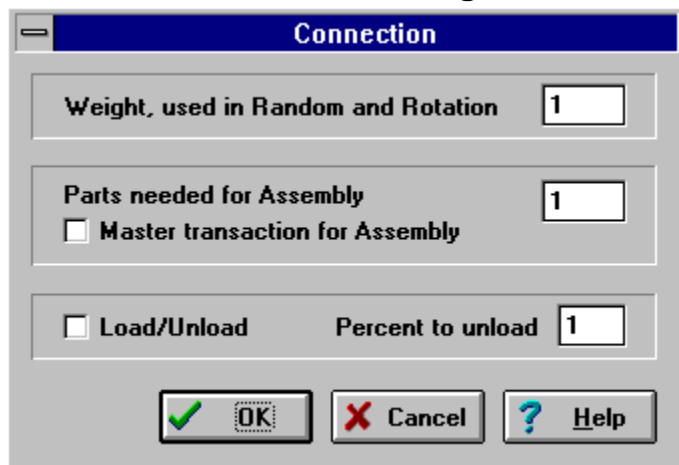
Assembly

When the next node is an assembly point, the count tells how many transactions from the connector are needed to satisfy the assembly.

Loading Dock

The connection may also be used to specify a loading dock.

The connection edit dialog



The screenshot shows a dialog box titled "Connection" with a blue header bar. It contains three sections, each with a label and a text input field:

- Weight, used in Random and Rotation**: Input field contains "1".
- Parts needed for Assembly**: Input field contains "1". Below this is a checkbox labeled **Master transaction for Assembly**, which is currently unchecked.
- Load/Unload**: Input field contains "1". To its left is a checkbox labeled **Load/Unload**, which is currently unchecked.

At the bottom of the dialog are three buttons: **OK** (with a green checkmark icon), **Cancel** (with a red X icon), and **Help** (with a blue question mark icon).

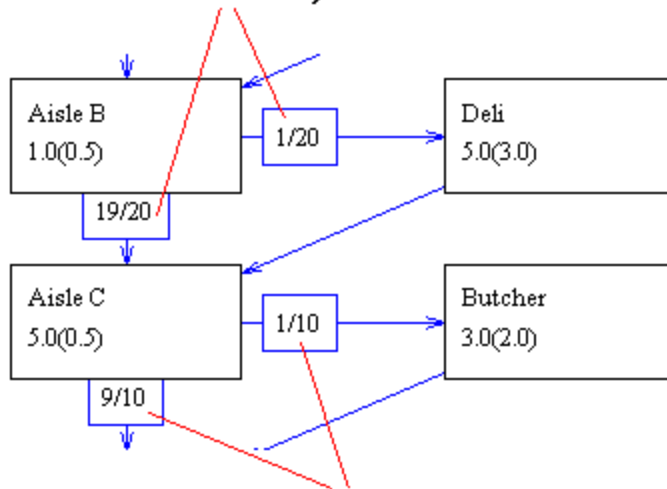
Random Choice Connector

The odds of a move



If the Pick one at random button is selected, you can use connection weight to affect the probability that a particular path will be taken. This is how you model transactions scattering at random. One customer in twenty visits the Deli in a supermarket. One part in fifty is junked by final inspection. One form in five is sent back for rework. One check in twenty bounces.

One customer in twenty visits the Deli



One customer in ten visits the Butcher

What the weights mean

A weight can be assigned to a connector. For example, if a node has two outputs and you do not give a weight to either, then each has a fifty-fifty chance of being selected. You may assign a weight to either or both connectors to affect the odds. If you assign a weight of 2 to one of them and a weight of 3 to the other, then the first has a 2/5 chance while the second has a 3/5 chance.

In general, the probability of a connector being selected is its weight divided by the sum of all of the weights.

Resources

What are resources?

Resources are the people, tools or supplies needed for a process to work. They can be assigned to a process for the duration of a job. If a process needs resources it waits until everything it needs is available before starting a job. The simulator uses tokens to allocate resources.

What are tokens?

Tokens are used to allocate resources to a process. Each resource has a limited number of tokens. A process may require one or more tokens from one or more resources. A process is stopped until it can get every token it needs.

When a process that needs resources has work to do, it first checks the resource pool to see if the resources are available. If not, it does nothing. If the resources are available, it locks everything it needs and continues. At the end of a job it releases its locked resources to the resource pool. Once the resources have been locked, they will not be released until the job in progress is complete. To avoid deadlock, resource allocation is all or none; if a node cannot get everything it needs, it gets nothing.

Example

The simplest example of a resource is a piece of equipment that cannot be shared. It has one token. If you have three of these, there are three tokens -- up to three processes that need the equipment may run simultaneously.

A more complex example would be a construction crew of six. Two jobs needing three people can run in parallel. If another job requires four, it cannot run in parallel with either of the others.

Resource Pool

Node resources

Resource Pool

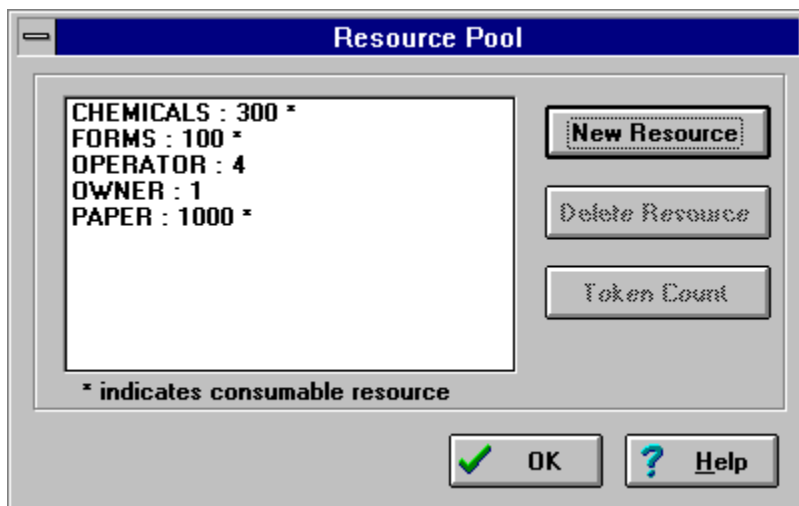
All resources belong to the pool

The resource pool is a list of all the resources available to a simulation. The resource pool does not belong to any node. You can think of it as a pile of tools stacked off to the side.

Changing the resource pool



Click this button on the speed bar and use the following dialog box to edit the resource pool.



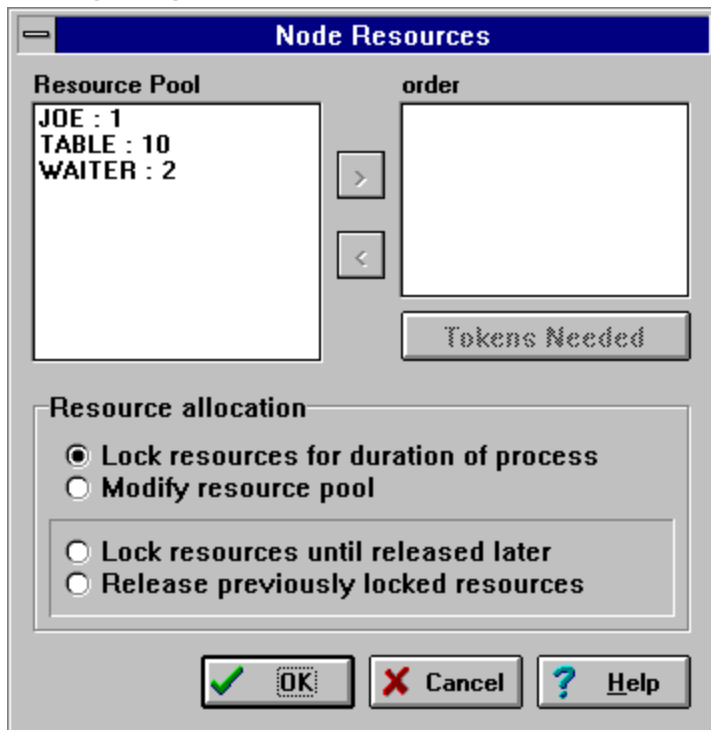
When a process needs resources to run, it checks the pool to see if the resources are available. If so, the process borrows the resources from the pool. When the process is finished, it returns the resources to the pool.

What about consumed resources?

Consumed resources are indicated by * following the number. They are not returned to the pool. They simply vanish.

Node Resources

Assigning resources to nodes



This dialog is used to assign resources to a node. Only resources that have already been defined in the resource pool can be used.

You may add or remove resources to the list needed by this node. You may also change the number of tokens needed by the node.

To add a resource, click on the resource in the Resource Pool window. The button with the > character will be enabled. Click the button to add the resource.

To remove a resource, click on the resource in the right window. The button with the < character will be enabled. Click the button to remove the resource. You may also change the tokens needed when the button is enabled.

Resource allocation

Lock resources for duration of process

This causes the resources to be locked only for the duration of the process. This is a safe allocation method. It cannot cause a deadlock.

Modify resource pool

You can have the node modify the resource pool. The most common uses of this are to model the restocking of supplies or sudden equipment failure. The resource pool will be modified by adding or subtracting the tokens shown in the right window.

You indicate a subtraction by making the token count negative.

Lock resources until released later

At times it may be necessary to hold a resource for several processes. For example, in a restaurant a table is locked for the duration of a meal. This is independent of whatever resources are needed during the meal.

This can cause a deadlock. Suppose you have two chains of processes, each needing the same two resources. If these two chains do not lock the resources in the same order, you can get into a "wait forever" state.

Suppose chain 1 locks resource A, then later tries to lock resource B. If, in the meantime, chain 2 has locked resource B and is now trying to lock resource A, the two chains will be waiting for each other. Forever.

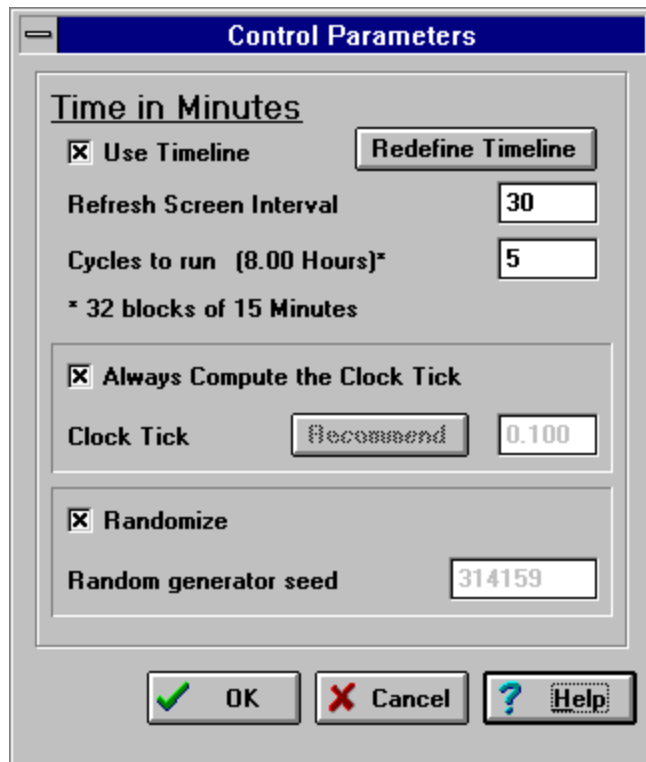
Release previously locked resources

This releases that were locked in another process. You will get a run time error if you release resources that have not been locked.

It is entirely up to you to make sure the locks and releases match.

Run Parameters

These parameters control the running model



Use Time Line - If this box is checked the time line will control when blocks are running. If it is not checked, all blocks are always running.

Redefine Time Line - This allows you to completely redefine the time line. It is the same as new project, except that the graph is saved.

Refresh Screen Interval - How often to redraw the screen when running. The time is in units of the simulation clock. If this is too fast, the screen flicker will be irritating. If it is too slow, you may miss something interesting.

Simulation Duration - How many time line cycles to go through in the simulation run. This is normally 1, but you may want to simulate several periods back to back. For example, you may want to see how work left over from the previous day affects operations.

Always compute the Clock Tick - The same as pressing the recommend button before each run.

Clock Tick - The rate of the low level "heartbeat" of the simulation.

Randomize - Set the random number seed to a value taken from the internal clock on your computer. Successive runs will use different data. This is a good way to see how different sets of data affect your simulation. If this box is not checked, the random number seed will be set to the given value at the start of each run. This will produce repeatable runs.

[See Clock](#)

[See Clock Tick](#)

Clock

Simulation clock

The Simulation Clock, or clock for short, is an internal counter that keeps all of the nodes working together. It has nothing to do with the internal real time clock on your computer. You may be able to simulate a week of office operation in a few seconds, or it may take several hours to model a second of computer network traffic.

The units on the clock are arbitrary; they mean whatever you want them to. Most business simulations use a clock unit of one minute. This gives familiar times such as 5 to fill out a form, 45 for lunch, and so on.

Clock Tick

Clock Tick

Here there be tigers

For you mathematicians, the Clock Tick, as given in the Control Parameters dialog, is the amount the clock is advanced with each iteration of the simulation. It is the "heartbeat" of the simulation. You might want to think of it as "dt" in a numeric integration. It is usually much smaller than the time unit. It needs to be small enough to resolve the smallest delay, or the narrowest variation of delay. If you use the Recommend button for the Clock Tick, the interval produced is 1/5 of the smallest (non zero) delay or narrowest (non zero) variation, whichever is less.

This is a good place to start. For your simulation you may want to change it. The trade off is between time to run the simulation and resolution of all times.

Use the recommended value unless you have a good reason to change it.

Clock tick ambiguity

Transactions remain in a node for a minimum of one clock tick.

DynaMetrics

The Three Basic Concepts

A simulation is how the following three interact.

The Graph

The graph, or map if you prefer, is the logical layout of the business. It shows blocks where work is done, how long it takes, and where the work flows. It is very similar to a classic work flow diagram.

Transactions

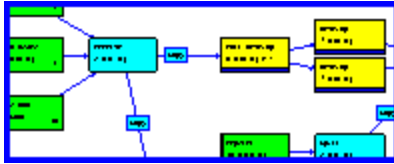
Transactions are the pieces of work flowing through the business.

Resources

Resources are the auxiliary things needed by the graph. A box on the work flow diagram may require one or more resources to run.

The Graph Window

Building the model



You build the model in the graph window. You add, modify and delete objects using the right mouse button. You move objects by dragging them using the left mouse button. You connect boxes by holding down a shift key while dragging between boxes in the direction of work flow.

Notation on boxes



Use the Zoom in, Zoom out and Make fit buttons on the speed bar to size the graph.

The Left Mouse Button

Use the left mouse button to rearrange items on the graph.

The Right Mouse Button

Use the right mouse button to add, modify or delete items on the graph

Labels

Labels are the text blocks you put on a graph. They play no part in the model.

The working part of the graph

Nodes

Nodes are the boxes on the graph that indicate work to be done.

Connections

Connections are the arrows on the graph that show work flow between nodes.

Node details

Node parameters

You need to specify details for the nodes on the graph

See animation

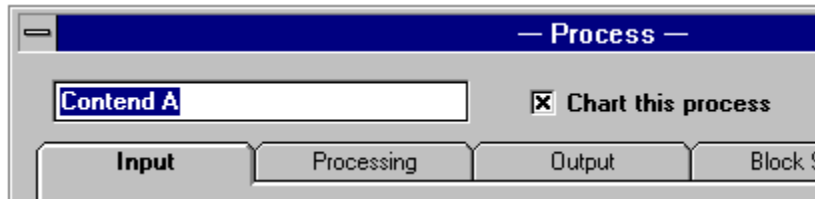
Label

Labeling the graph

Labels are just text that you can put on a graph. They can be placed anywhere. You can set the font of each label.

Node Parameters

The edit node dialog is a tabbed notebook. You click on the tab for the parameters you want to edit.



The text box in the upper left corner is the name of the node.

Chart this process - This check box only appears for process blocks. Very often you will only be interested in reporting a few of the processes in a model. This box tells whether the process is to be included in the summary window and in the chart window.

If you would like this dialog to open every time you create a node, click Automatic Edit under the Options pull down menu.

Details on tabs

[Input parameters](#)

[Processing parameters](#)

[Output parameters](#)

[Block Size](#)

[Node Pictures](#)

Node Input

For transactions entering a block

Input	Processing	Output	Block Size	Pictures
<input checked="" type="checkbox"/> Always Accept Input	This block accepts input during a scheduled break. An office IN basket is a good example.			
Work schedule				
<input checked="" type="radio"/> Normal queue	Wait in line, one transaction at a time			
<input type="radio"/> Can Interrupt	This work can be suspended for a higher priority job.			
<input type="radio"/> Multiple transactions	Can handle many transactions at the same time.			

Always accept input - The block will accept input during a scheduled break. The in basket of an office worker is an example. A counter example is a checkout line at a grocery. People do not get in line if the checkout register is closed.

Work schedule

When a transaction enters a queue, it moves ahead of all transactions of lower priority, but behind all of equal or higher priority.

Normal queue - Transactions are taken from the front of the line and processed one at a time.

Can interrupt - The process can be interrupted for a higher priority task. Example: baking a cake cannot be interrupted, but editing a chapter can. Work in progress will be placed on "hold" while a higher priority transaction is processed.

Multiple transactions - This block can handle multiple transactions at a time. The number of transactions is limited only by the availability of resources. Each transaction is scheduled for release when processing starts, independent of other transactions. For example, several people can walk down an aisle at the same time. They may not leave in the same order they arrived. Many people can eat in a restaurant at the same time, subject to the availability of tables.

Node Processing

Transaction processing in a block

Input	Processing	Output	Block Size	Pictures
Minutes to do job Mean <input type="text" value="0.25"/> Variance <input type="text" value="0.10"/>		Time Distribution <input checked="" type="radio"/> Normal <input type="radio"/> Exponential <input type="radio"/> Uniform <input type="radio"/> OneShot First <input type="text" value="0.00"/>		Resources Assign resources to this process
<input type="checkbox"/> Finish before quitting		Do not leave work in the IN basket. Finish the input queue before taking a scheduled break.		
Item count needed to resynchronize <input type="text" value="N/A"/>		Number of matching transactions needed.		

Time to do job - Mean: usual time; Variance: the expected deviation from the usual time. For a start block, this is the time between successive "fires" of the block.

Time Distribution - Normal: also called Gaussian or Bell curve; Exponential: many outside events follow an Exponential distribution, for example incoming phone calls; Uniform: flat random distribution

OneShot and First are for Start blocks only. OneShot means the block will be activated, or "fire" the first time it is scheduled, and never again. First is the time the start block will "fire" the first time; after that, successive activations will be determined by the mean, variance, and so on.

Finish before quitting - Check if the waiting line must be finished before a scheduled break. Example: Office workers will go to lunch and leave work in the in basket. A clerk will not go to lunch with people left in the checkout line.

Item count needed to resynchronize - Sync blocks only. The number of matching transactions needed to proceed. The number of parallel sub tasks to resume work.

Resources - assign resources to a process.

Animation

Watching the traffic



When the Animate button is pressed, each connector will flash when a transaction moves over it. This gives you a way to watch what is going on in a running model.

Animation does not affect screen refresh rate

The screen is refreshed according to the value given in the Run Parameters. This means that queue sizes, resource conflicts and so on will not be updated with each transaction move. The on screen numbers show the simulation state at the time of the last screen refresh.

Transactions

Transactions mean whatever you want them to mean

Transactions are the movable pieces that flow in a model. Internally, they are packets of data that are created, move from one location to another, and are eventually destroyed. They include information such as time created, priority, name of originating node and so on. You will never have to deal with them directly, but you need to be aware of them; they are what make dynamic modeling possible.

The name of the originating node is important in one case. When the Dispatcher is trying to determine the next node, if the next node name matches the transaction name that node is chosen. If no name match is found, the node selection criterion is used.

Transaction ID

Transaction identification

Identifying information

Name

Each transaction carries the name of the originating Start or Assembly block. This name can be used to route transactions. NOTE: An Assembly block does NOT change the name of the Master transaction.

Unique ID

Transactions are assigned an identification (ID) number when they are created. A Split block copies transactions and gives them the same ID. This is the ID used for matching transactions in a Sync block. NOTE: An Assembly block does NOT change the ID of the Master transaction.

Reading the Screens

What do the displays mean?

There are several displays in the simulation. This section contains details on how to read them.

The Main Windows

Building the model

Graph Window

Time Line Window

Table Window

Results

Resource Use Window

Summary

Charts

Other Windows

Show Statistics

Table Window

Tabular display of node parameters

Table							
Name	Type	<u>Ct, Pr, Min, Max</u>	<u>Delay</u>	<u>+/-</u>	<u>Dist</u>	<u>Next</u>	<u>W A F I</u>
Noon Rush	Start	[1,15,0,0]	2.00	4.00	Exp	Shortest	
Carry out	Start	[1,15,0,0]	20.00	400.00	Exp	Shortest	

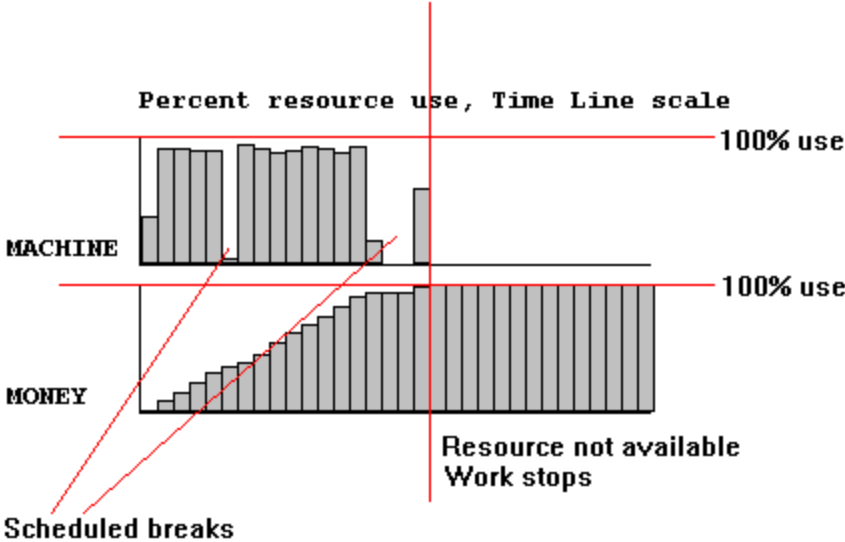
Annotations and their corresponding table columns:

- Count: Points to the **Ct** column.
- Priority: Points to the **Pr** column.
- Cargo: Points to the **Min** column.
- Mean: Points to the **Delay** column.
- Variation: Points to the **+/-** column.
- Wait in line: Points to the **Next** column.
- Always accept: Points to the **W** column.
- Finish: Points to the **A** column.
- Interrupt: Points to the **F** column.
- Transaction: Points to the **Ct** column.
- Delay: Points to the **Delay** column.
- Next node: Points to the **Next** column.
- Process Switches: Points to the **W A F I** columns.

Resource Use Window

How resources are used over time

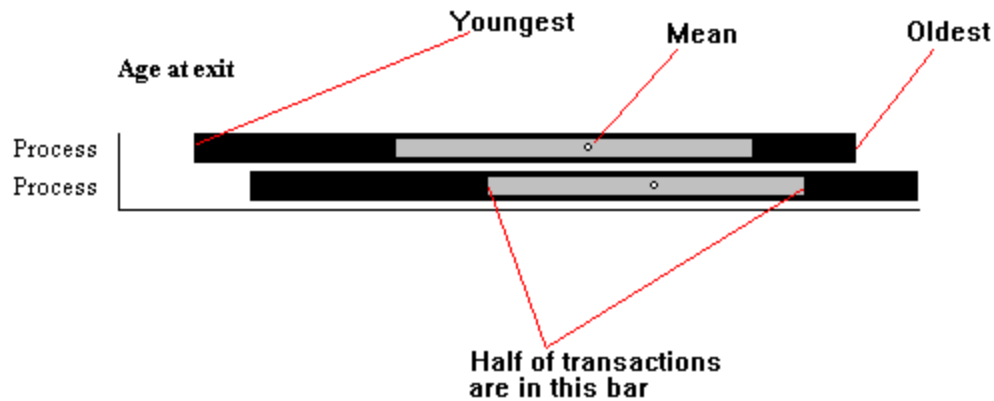
Each vertical bar corresponds to a block on the Time Line.



Resources
Presentation Graphics

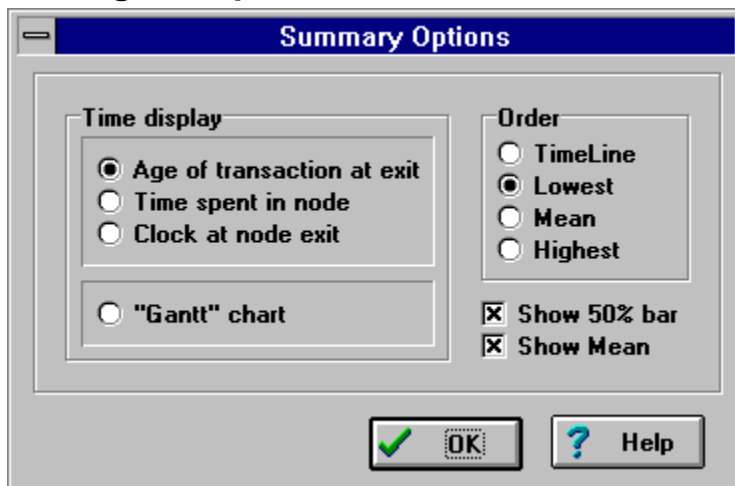
Summary

Statistical summary of processes



Presentation Graphics

Setting the options



Age of transaction at exit

This is the time measured from the creation of the transaction. This tells you "work progress" on transactions since you started work on them.

Time spent in node

This is the total time the transaction spent in the node, counting queue time. This option tells you which nodes are taking the most time.

Clock at node exit

This is the time measured from the beginning of the simulation. This tells you what "time of day" the transaction left the block.

"Gantt" chart

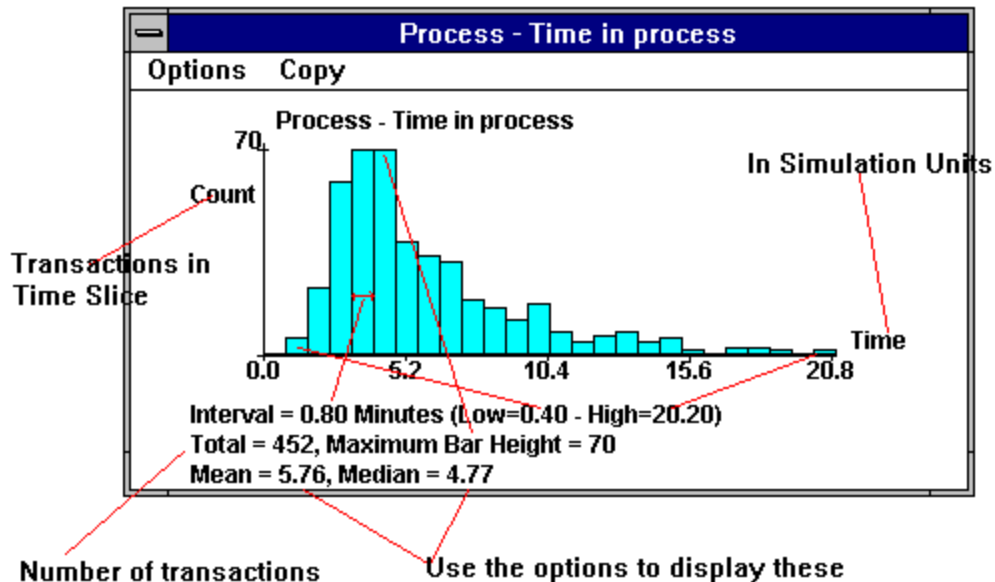
This is the familiar task timeline chart. It differs from the above in that each bar represents two times: task start and task stop.

See Histogram

See 50% Bar

Show Statistics

Relative frequency of transaction times



This diagram shows the time spent by transactions in the process. The time scale is broken into intervals. In this case the interval is 0.80 Minutes. A counter is associated with each interval. When a transaction leaves the process, the length of time spent there is computed and rounded to the nearest interval. The counter for that interval is increased by one.

The above histogram shows the count in each interval. The bar height represents the number of transactions that hit in that interval, and the bar width represents the interval itself.

This is a Pop-Up Window

You get this display by pointing at a box on the graph and using the right mouse button to "Show Statistics"

Three different displays

There are two different sets of statistics and one table given, depending on the type of block chosen.

Start and Stop blocks

Start - Gives the time from creation of a transaction until it reaches any Stop block. If several stop blocks are possible for transactions starting here, they will all be included.

Stop - Gives the time from creation of a transaction until it reaches this Stop block. If several start blocks have contributed they will all be included.

Note: If only one Start and one Stop block are in the model, the above two will be identical.

Assembly blocks

For assembly blocks, the display contains the inventory - the parts on hand.

All others

Gives the time from queue entry for the node to block exit. That is, the time the transaction spent in the node, counting wait time.

Auxiliary information

There are three measurements commonly associated with these diagrams:

Median. This is the one in the middle. Half of the transactions were faster and half were slower.

Mean. This is the average. It is computed by adding all of the times and dividing by the total number.

Mode. This is the value that occurs most often: the location of the tallest bar. There is no provision to show the mode; you can tell at a glance what it is. While a distribution has only one mean, and only one median, it can have several modes. Multi Modal distributions are common in simulations. That is, the histogram will show more than one distinct "peak".

Many consider the median to be the single most informative number in a histogram.

See Summary

See 50% Bar

50% Bar

What does the 50% bar mean?

The 50% bar shows the range in which half of the transactions fall. 25% are below this range and 25% are above.

In a normal distribution the mean, median and mode all fall at the same place. A distribution is said to be skewed if there is a difference between these values.

Summary

Each bar in the summary represents many transactions. The big bar runs from the lowest time for a transaction to the highest. But it tells you nothing about the distribution of the transactions inside the bar. The 50% bar contains half of the transactions. It gives you a quick feel for whether the distribution is skewed.

Histogram

In a histogram it is easier to see the skew in a distribution. The 50% bar makes it easier to see the distribution.

Both

The median will be close to the center of the 50% bar. It will always be inside. The mean may be outside. If the mean is outside the 50% bar the distribution is badly skewed. This is not necessarily a sign of trouble, but it does mean that a few transactions are way out of the normal range.

In the case of skewed distributions, you will probably want to look at the statistics for the process in question for details on what is going on.

[Histogram](#)

[Summary](#)

Fonts

Font size, style and color

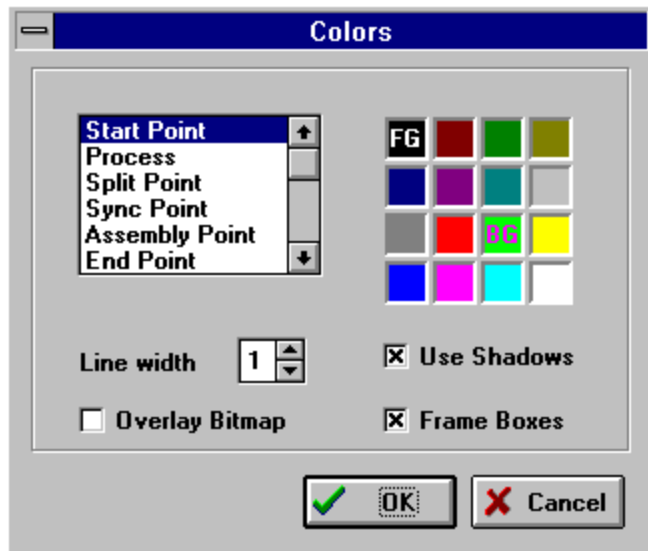
You can change the font for any window. Simply make that window active and use the Font button or menu item. You will get a dialog box that lists the fonts available on your system.

You will usually get more readable screens if you use True Type fonts. On the Table window you should use a monospace font such as Courier. If you use a proportional font such as Arial, the columns will not align correctly.

When adjusting fonts on the Graph window, If a label is selected, the font applies only to that label. If nothing is selected the font choice applies to all text on the screen except labels.

Colors

Setting colors and line widths



Use this dialog to control screen colors. Pick the item you want to set from the list on the left and the colors from the color box on the right. Click on a color with the left button to set the foreground color, the right button for the background.

Set the line widths using the Line width spin control.

Overlay Bitmap

Check this box if you want the normal text superimposed on a bitmap. See [Node Picture](#).

Use Shadows

Check this box if you want the boxes on the graph to cast "shadows".

Frame Boxes

Check this box if you want to place a frame around boxes on the graph.

When you have all of the settings the way you want them, click OK.

Exporting Data



Copy to clipboard

DynaMetrics has no printing facilities. The strategy is to allow any data to be copied to the clipboard, then pasted into your favorite word processor, publisher or slide show generator. This relieves you of the task of having to learn a new publishing system. You simply copy what you want and paste it into something you already know how to use.

There are three formats used to export data. All of these are standard Windows formats that are commonly used in other applications.

Text - simple ASCII text used by almost all applications. The Table and Parts List windows are copied to the clipboard in this format.

Metafile - standard Windows graphic format. All other windows except presentation graphics are copied to the clipboard in this format.

Bitmap - standard Windows graphic format. Presentation graphics are copied to the clipboard in this format.

Printing

PRINT.EXE is a utility distributed with the software that will print a WMF clipboard. You can use this to get a printout of any metafile window. Simply copy the window to the clipboard and use PRINT.EXE to send it on to the printer.

Getting a bitmap image

Some word processors will only work with bitmap images. If you need a bitmap image, there are three easy ways to get one.

Presentation graphics use bitmap images.

The Paintbrush program that comes with Windows will allow you to paste a metafile image. Paintbrush will then save the image as a bitmap file.

You can press the PrintScrn button (usually on the top row of the keyboard, just to the right of F12) at any time. This copies the screen to the clipboard. Or you can press Alt-PrintScrn to copy a window to the clipboard. In either case, you can then paste the clipboard in Paintbrush.

Glossary

50% Bar

A

Animate

ASAP

C

Carrier

Click

Clipboard

Clock Tick

Connector Properties

Connector

D

Deadlock

Drag

E

Exponential

F

F1

G

Gaussian

Graph

Grid Snap

I

Interrupt

M

Master Transaction

Metafile

Model

N

Node

O

OneShot

P

Poisson

Priority

Q

Queue

R

Resource Pool

Resource

S

Simulation

Skewed

T

Three Basic Concepts

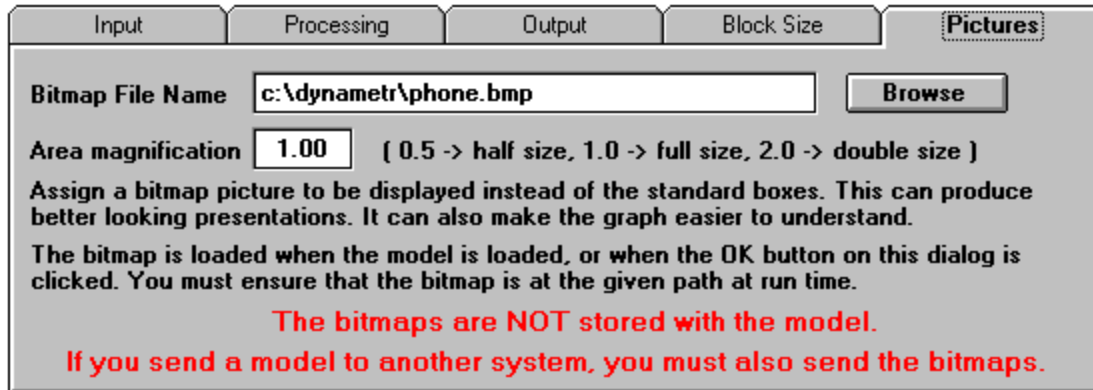
Transaction

W

weight

Node Picture

Showing pictures instead of blocks



Sometime you would like to show pictures instead of the blocks that are the default. Use this dialog page to assign a bitmap to the block. The bitmap will replace the block in a normal display. If the bitmap is not in the specified location, or is in the wrong format, the default block will be shown. Use the Area magnification to change the size of the bitmap picture. Note that this is area magnification; if you want linear magnification, you should square the number. Note also that the magnification is in addition to any zoom in effect.

Hiding pictures



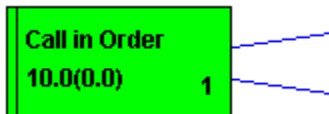
Use the Hide button on the speed bar to show the regular blocks. The pictures are not removed, they are just temporarily hidden.

A start block using a picture



Call in Order

The same start block with the picture hidden



Two ways to show the text

Dialog box under Colors has an item named Overlay Bitmap. When this is checked, the normal contents of the box are superimposed on the bitmap. When this is not checked, the name of the node is shown under the bitmap.

The bitmap can be anything

In this case, the bitmap is an empty frame surrounded by notes. (Maybe Charlie plays the saxophone during breaks.)



The ABCs of Modeling

This was written by one of the field testers, and has been included with her permission. jla

What is DynaMetrics(TM)?

The DynaMetrics system is a business simulation program. It allows you to build a small working model of your business so that you can try experimental solutions to very real problems -- and see on your screen what might happen.

The DynaMetrics system lets you build a simulation or a computer model of your business. Using the building blocks that the program supplies, you tell it about your work processes, the time involved, the competing resources, the usual workload ... all the factors that decide whether your business runs smoothly, overloads, or stalls for lack of resources. Then you take that working model and try improvements. Try different ways of increasing customer satisfaction through tangible measures such as shorter waits, or different ways of increasing productivity and profits.

Your own business has its own particular examples of questions that are hard to answer unless you actually try the solution. Let's take a real-life example. Someone owns a restaurant and is looking for ways to get more business. Would it be better to get more tables, or to hire more wait staff, or to buy another broiler? Without a working model of your business, you may not have any way of answering such questions.

The DynaMetrics program also lets you analyze your existing business for equipment or personnel who are underused - or overused. For example, if you have a retail store with two full-time cashiers and one on-call cashier, how much of the time is the third person used as a cashier? How much of the time is the second cashier idle? How would it affect a customer's average wait if you change the number of cashiers? Again, without a working model of your business, these questions can be difficult to answer.

The ABC's of business simulation:

Have A Question In Mind

A business model can provide many useful answers. The key to building a meaningful business model is to have in mind a question you are trying to answer. How can I keep my shelves better stocked? How can I cut down on delay? How many people do I actually need for a particular function?

Before you begin building your business model, decide the questions you are trying to answer.

Different Questions, Different Models

The business model you build depends on the questions you are asking. For example, if our restaurant owner was trying to determine how to serve more customers during lunch, his model would include the limited number of tables, the limited number of wait staff, and the average time to fill an order in the kitchen. If that same restaurant owner decided that he wanted to concentrate on improving performance in the kitchen, he would build another model that might include the size and number of broilers, the number of cooks, and other details very particular to the kitchen.

Limiting the Scope

In your own business, the models you build will focus on the questions you are trying to answer. You can see from the examples above how asking different questions would lead you to build models using different levels of detail. You might build a very general model of your business showing the "big picture." You may decide to build detail models to work on particular aspects of your business process. We refer to this idea as "limiting the scope." This means that you add just enough detail to answer the specific question you are trying to answer. If you find a model becoming cluttered with detail, consider

whether you have limited the scope of your model to only what is needed to answer the question at hand.

Gather Enough Information

Before you build your business model, you will need to provide the DynaMetrics program with an accurate picture of how your business works. At each step of your business, do you know the normal workload and how long the process takes? You could guess, but guesses may not be accurate. Use some of your existing business documentation to track how long each process takes. If you currently have no way of knowing how long a process takes, consider ways to track the time involved.

There is a rule of thumb for interpreting computer results that is commonly known as "Garbage In, Garbage Out" -- if the information you provide to a computer is meaningless, then the answers provided to you in return will be equally meaningless.

When you build your business model, expect to spend some time gathering the information you need. In the beginning, you may not know exactly how long each process takes, the average time, the average delay. At first you may not know exactly how many transactions go through your business on a busy day, a slow day, and an average day. Analyzing your business is part and parcel of building a meaningful business model. The information you gather will be useful to you in many ways and will help you keep a better handle on your business. If your model is to help you solve and analyze real business problems, it is essential to build the model using realistic information.

Building Blocks

You will create your business model using building blocks that the DynaMetrics program supplies. Here are the basic types of building blocks:

Transactions

When building your business model, the most essential idea to grasp is the transaction. A "transaction" is one unit of work. In the example of a restaurant, the transaction might be a party of customers; in the restaurant kitchen, the transaction might be a food order. In different industries, a transaction might be an order to fill, an item to manufacture, or a customer to serve. The DynaMetrics program even allows you to split a transaction in the event that a transaction in one part of the model is divided into two or more separate pieces that you process separately. For example, in a laundry the paperwork and the dry cleaning follow separate paths, and the delivered product must wait until the matching pieces get back together.

Time

The DynaMetrics program allows you to choose how to track time. For some businesses it is vital to track time in minutes, or perhaps even seconds. In other processes, hours or days may make more sense. You will need to decide for each of your models how you want to track time.

Once you have decided on a question you are ready to build a business model. Your model will consist of starting points (where a transaction enters the business), processes (where work is performed), and exit points (where a transaction is completed). As we discussed before, you may have transaction splits to divide a transaction into other units of work, and you may have "resynch" points where the different parts of a transaction come back together.

Finally, you will have "queues" which are lines in which a transaction waits its turn. A queue might be a literal line of customers or a stack of orders awaiting processing.

How To Sketch A Rough Model

Now that you have been introduced to the basics of business simulation, it is time to begin creating real business models. We will begin by creating a rough model of your business. You may build a number

different models to examine different aspects of your business.

First, think of a flow chart of your business. For our sample restaurant, you may think of the customer waiting to be seated, placing an order, waiting for food, paying, and leaving.



Now let's consider a problem we would like to tackle: how can I serve more customers during the lunch rush? We also determine the different solutions we would like to try: more tables, more wait staff, or shorter waits in the kitchen. Notice that our original model did not include the number of tables as a limitation, nor the number of wait staff. We need to be sure that our model includes all the factors that we are considering.

The next step in building the business simulation is gathering information. We make a quick list of all the delays involved and a list of the limiting factors such as the limited number of tables.

Process Description	Average Duration	Resources Required
Wait for table	1 minute	table
Wait to place order	2 minutes	wait staff
Wait to receive order	1 minute	wait staff
Eat meal	20 minutes	none
Pay & leave	1 minute	wait staff

Resource Name	How Many?
Wait Staff	5
Tables	25

When gathering the information you need, you will soon see the importance of the "Garbage In, Garbage Out" rule. Remember that the results you get from any computer program are only as accurate as the information you give it. A useful test of your model is to run the very basic model without any "what-if" scenarios to see if you get the results you would expect based on your normal business. Do you get the right number of transactions, the right amount of delay? If any of the results from your business simulation are out of line with what you expected, fine-tune your simulation before going any further. Before trying any special scenarios, you should be able to approximate your current situation.

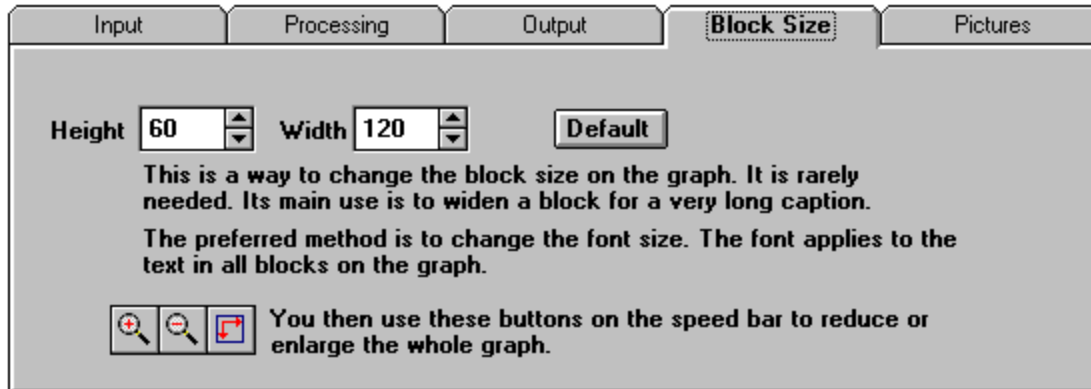
Remember the steps we took to build the simulation. We decided on a question to ask or an improvement to try, and we sketched a simple flow chart. We identified the major pieces of our business model. Then we gathered information about the work load and the different delays involved in completing the work. We decided, for this model, what we will use as the unit of work, and whether to measure time in large units such as days or small ones such as minutes or seconds.

We are now ready to build a meaningful business model.

Node Size

Changing the size of a block

Use this only when nothing else works.



There is very seldom a use for this. Do not try to fit the graph in the window by changing the block sizes. Instead, change the font size to make the text fit in the box. Then use the zoom buttons to change the size of the whole graph.

Notes on ZoomIn and ZoomOut.

The normal zoom does not change the model; it only changed the way it is projected on the screen. Normal zoom is completely reversible. If you hold down the Shift or Alt key during a zoom, the program actually goes through the model and changes the box locations or sizes.

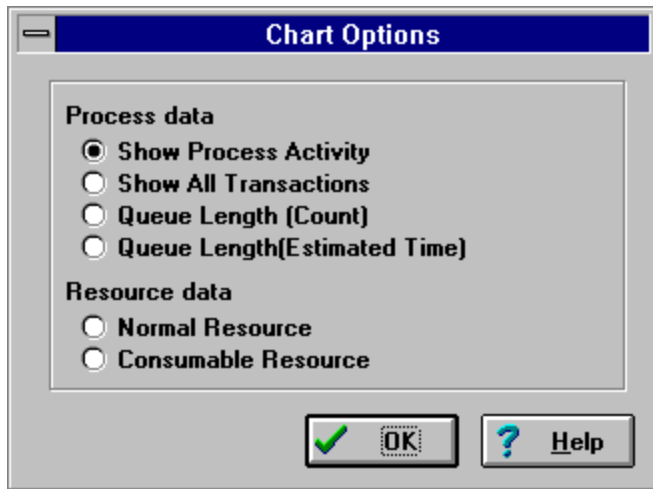
If you hold down the Shift key when you click, the drawing changes size by stretching the connectors and nothing else. If you hold down the Alt key when you click, the drawing changes size by stretching the boxes and nothing else. Fonts and so on remain the same. This gives you an easy way to "open up" or "tighten" the whole graph.

Be careful with these

Since floating point arithmetic is used, the Shift and Alt zooms are not completely reversible. This is especially noticeable if you have grid snap turned on.

Chart Options

Contents of the Chart window



Show Process Activity

Shows total time blocked for lack of resources, busy and idle for each process.

Show All Transactions

The data plotted is the total number of transactions at each process. This includes input queue, work in progress and work that is finished and ready to move to the next node.

Queue Length (Count)

The data plotted is the number of transactions in the input queue for each process.

Queue Length (Estimated Time)

The data plotted is the number of transactions multiplied by the mean processing time for each process.

Normal Resources

This is the percent of time a resource has been busy.

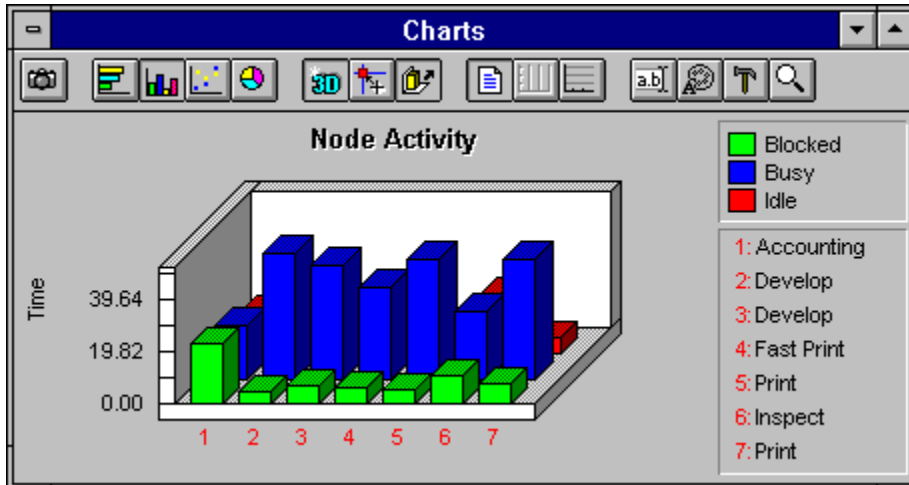
Consumable Resources

This shows the resources available and the total consumed. That is, parts on hand and parts used, or money available and cost so far.

Chart

Presentation Graphics

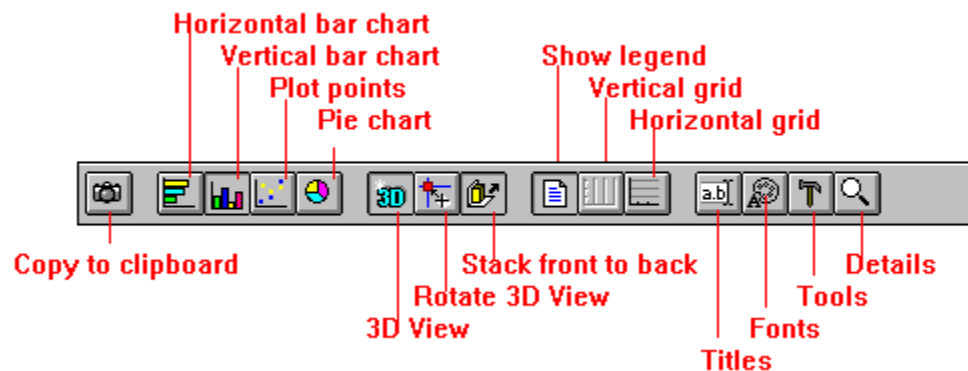
The chart window provides summary information that shows overall system performance. Since preferences vary considerably on how such information should be presented, there are numerous options available to you. You can pick the ones that suit you and your audience.



The best way to learn the presentation graphics window is to simply start playing with it. You can't break anything, so just start pushing buttons to see what happens. The speed bars are not the same on all windows that use this package. For example, some allow line charts instead of pie charts, because a pie chart would make no sense at all. The Charts window allows pie charts because some of the data displayed does make sense when shown in pie chart form.

One word of warning about pie charts: they show only the first series in the graph. For example, if the above chart were shown in pie format, the bars shown in green would be the ones used.

A key to the speed bar



Hints

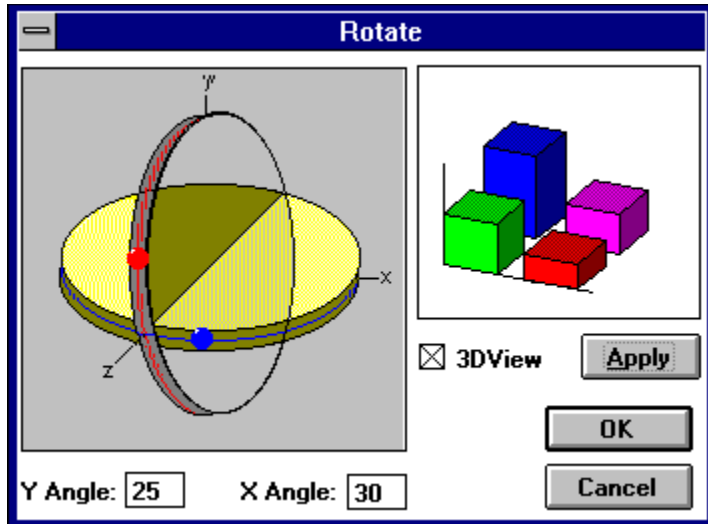
Rotate 3D View - allows you to view a chart from a different angle. For example, you can tilt the chart toward you to see bars hidden behind others.

Tools - Allows you to change the colors and patterns used in filling surfaces. You drag the color or pattern you want onto the surface you want to change.

Details - Allows you to change the scale, maximum values, gap between bars, and so on.

Click the right mouse button on the speed bar for hints.

Rotate 3D View



At times you may want to see a 3D View from a different point in space. You may, for example, want to look at the chart from overhead to see bars hidden behind others. The red and blue balls in the figure show your point of view. Drag the red ball upward (along the red path) to raise your eye level. Drag the blue ball to the right (along the blue path) to move your viewing position to the right. The 3DView box must be checked for this to have any effect.

[See also:Exporting Data](#)

Node Order

When picking the next node

When the dispatcher is looking for the next node, it examines the following nodes in order from top to bottom, and then from left to right. That is, the node nearest the top of the page is examined first, then the next one down, and so on. If two nodes have the same vertical position, the one on the left is examined first.

It usually does not matter where the nodes are located on the graph, but there are three cases where it is very important. These are By name, Rotation and Spill over in the next node choice. For all of these the scan order is important.

The reason for doing it this way is so the order will be clear when you look at the graph, and so you can change the order by simply rearranging blocks.

Getting started

Examples

The ABCs of modeling

How to use help

50% Bar

An optional bar on either the Summary or a Histogram showing where half of the transaction times occur. Roughly, the variance.

Animate

Show traffic by flashing a connector when a transaction moves over it. This helps you see what is going on in the simulation.

ASAP

As Soon As Possible

Carrier

Any transaction can carry other transactions as cargo. When you model a bus, for example, other transactions can "ride" on that transaction.

Click

To press and release the left mouse button without moving the mouse

Clipboard

An area in the Windows system, used for temporary storage of data. Used in most cut, copy and paste operations.

Clock Tick

The rate of the internal clock. For you mathematicians, this is dt in a numeric integration. Use the Recommended value unless you are absolutely sure you have a good reason to change it.

Connector Properties

In some cases, connectors have properties. 1) For a random choice of the next node, the weight of the connector is used to compute the probabilities. 2) For an assembly node, the transaction count needed from the connector can be specified. 3) For a loading point, the percent to unload must be given.

Connector

A line joining two nodes together on the graph, indicating transaction flow. In a traditional work flow diagram, these are the arrows joining two boxes together.

Deadlock

The simulation being stopped because of unresolvable resource conflicts. The most common cause is two chains of procedures locking the same set of resources in a different order.

Drag

To move the mouse while holding down the left mouse button

Exponential

A frequency distribution that often describes the time between certain random events. For example the time between customer arrivals or between incoming phone calls usually follows an exponential distribution.

F1

Help key. Press the F1 key to get help on the active window.

Gaussian

Also called a normal distribution or bell curve

Graph

The graphic representation of the layout. The work flow diagram. A logical layout of a business.

Grid Snap

When moving an object on the graph, always place the center on a grid point. The grid size is specified in the grid snap dialog.

Interrupt

To suspend work on a transaction to handle a transaction of higher priority. This is done in processes where the "Can Interrupt" box has been checked.

Master Transaction

Used in assembly points. If a master transaction is specified, that transaction will be the one sent to the next node when the assembly conditions have been satisfied.

Metafile

A standard Windows format for storing graphics. Can be pasted into most word processors.

Model

The graph, time line, resources and so on needed to describe a business. The static part of a project.

Node

Any box in the graph, or map of the work flow diagram.

OneShot

A timing choice for Start blocks. The block will "fire" the first time it is scheduled, and never again.

Poisson

A common statistical distribution. Often describes the frequency of occurrence of certain random events. The rate of events with exponential inter arrival times.

Priority

The importance of a transaction. Each transaction has a priority. When a transaction enters a queue, it moves ahead of all transactions of lower priority.

Queue

A waiting line. Things waiting for attention on a First-In, First-Out basis.

Resource Pool

The collection of resources that are available to processes during a simulation.

Resource

An auxiliary piece of equipment, or a person, or supplies. A process can require certain resources. If it cannot get these from the resource pool, it remains idle until the resources become available.

Simulation

A running model. The interplay of the graph, resources and transactions.

Skewed

Biased or distorted. Not symmetrical.

Three Basic Concepts

These are the graph, transactions and resources.

Transaction

The basic unit of work that flows through the simulation. The meaning of a transaction is entirely in the mind of the person defining the model. Typically, transactions represent a piece of work, or a customer, or a form or a part. Transactions move between blocks, get delayed, have to wait in line and so on.

weight

The relative probability that a particular path will be selected.

Compound Nodes

Compound nodes are purely visual

These nodes have no effect on the simulation; they are merely a way to group other nodes to make the graph easier to read.

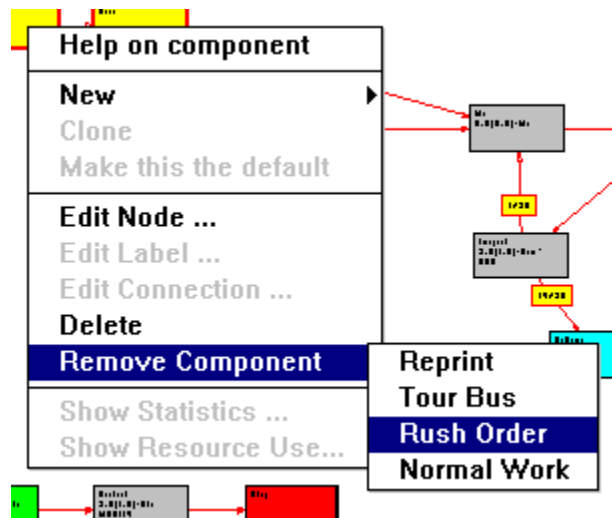
For example, you may have a portion of your model with several nodes that represent accounting. You can simplify the graph by creating a compound node called "Accounting", then dragging the appropriate boxes into the compound box. The original boxes are hidden, but retain their original parameters, connections, sizes, locations and so on.

The simulation runs as if the component nodes were in their original screen positions.

Editing compound nodes

The Edit Node dialog will only allow you to change the Block Size and the Picture.

If you remove a component from a compound node, it will snap back to its original location. If you delete a compound node, all of the component nodes will go back to their original states.



To remove a component, use the right mouse button to get this menu and move the cursor to the Remove Component menu item. You will get a sub menu listing the component nodes. Click on the one you want to remove.

International

Formats

The date, time and number formats used are those given in the International settings in the system control panel.

Split node

A split node is a process in which the Copy All option has been checked. A copy of each transaction is sent to each following node.

Critical Path

The longest path

How the path is computed

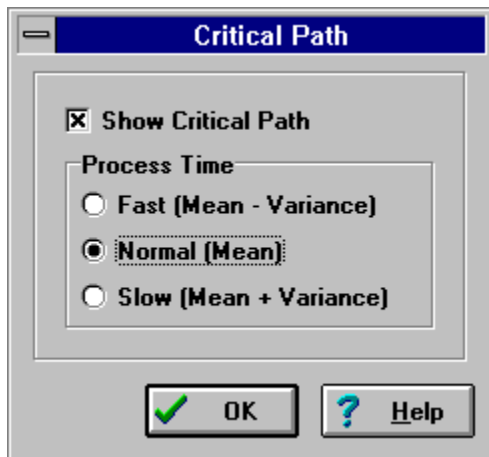
The critical path is the one which takes the longest total time.

Only processes enter the calculation.

All possible paths between Start blocks and Stop blocks are considered.

Any path that contains a loop is rejected.

Passenger load/unload paths are rejected.



Use this dialog to specify the critical path.

For many models the critical path makes no sense. For example, any process that has as its main delay the repetition of some blocks will produce a nonsensical critical path. The Amusement Park example included with this package shows why. Most of the delay for patrons is in the Roller Coaster. But since that is a loop, it is rejected from the critical path calculation.

Why is there more than one critical path?

In the above dialog box, there are three ways to compute the critical path: fast, normal and slow. That is, optimistic, expected and pessimistic. Since some tasks will have larger variances than others, the critical path may change. This is a good way to tell which "non-critical" tasks you need to watch.

Critical paths and the classic PERT diagram

The classic PERT diagram is a very special, limited case of a model. If you want to build a PERT model, you should impose the following constraints.

No loops

No assembly blocks

No loading docks

One start and one Stop block

The start block must be a "one shot" that fires a single transaction

All following node choices must be "Copy All"

All nodes with more than one input must be Sync blocks

All variances must be set to zero



You can temporarily set all variances to zero by pressing the lock button.

Project schedules with uncertainty

PERT was designed to help schedule a project. It is, by its very nature, a one-shot operation; you are only going to do this project once.

Normal planning assumes a fixed time for a task. This is seldom the case. As in all business models, most task durations have some uncertainty.

How do I simulate uncertainty?

The easiest way is to change the start block from a one-shot to one that starts many transactions, and unlock the variances. You can then look at the [statistics](#) to see the results. The summary with the "[Gantt](#)" option checked is also useful.

You should make sure that the transactions cannot interfere with each other; you do not want any queues to become active. You do this by either slowing the start block so that the network can easily keep up with the traffic, or you can check the "[Multiple transactions](#)" radio button in the Edit Node dialog for each process.

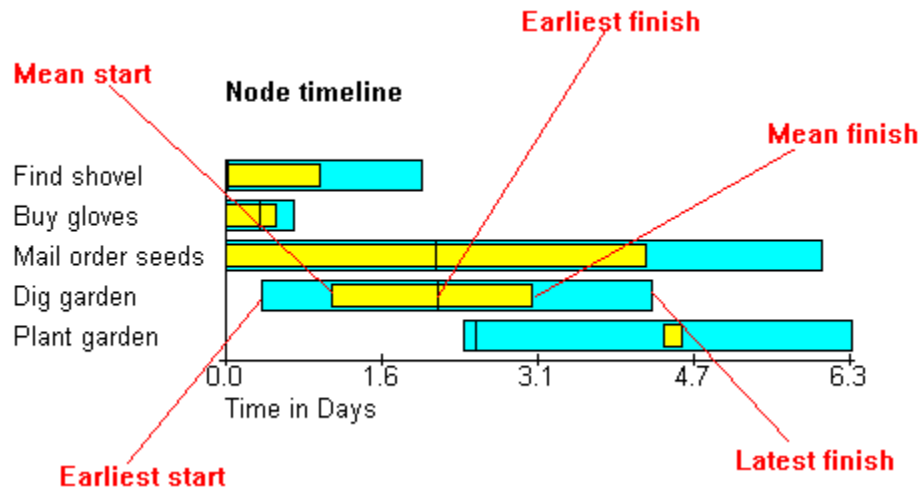
What does a transaction represent in this case?

Nothing. It is just a timing pulse.

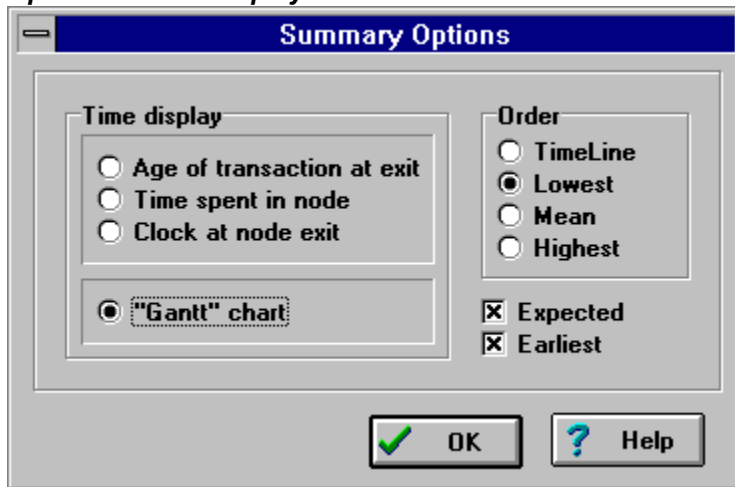
Project Schedules

The Task Timeline

The task timeline differs from the other summary options, in that each bar represents two times: task start and task finish. In this context, task and process are synonyms.



Options for this display



Expected - the inner bar runs from the mean start to the mean finish.

Earliest - a vertical bar is placed at the earliest finish found during the run. This is the earliest date you can't prove will slip. It will, of course, but you can't prove it at the outset.

Clone

Modeling identical processes

Sometimes you want to model several processes that are the same. You could, of course, build identical processes. But this would entail editing all of them if you want to change something; keeping the processes alike would become a problem. This could be a slow, error-prone operation.

Clones provide a way to tell the program that several processes are to be maintained in parallel. Any changes made to one of them should be made to all of them.

To create a clone

Point to the process you want to clone, press the right button and select Clone from the pop-up menu.

What is maintained

The block name, functionality and resource use are maintained in parallel. Block size, picture and timeline schedules can be set separately.

Clones are convenient, not necessary.

