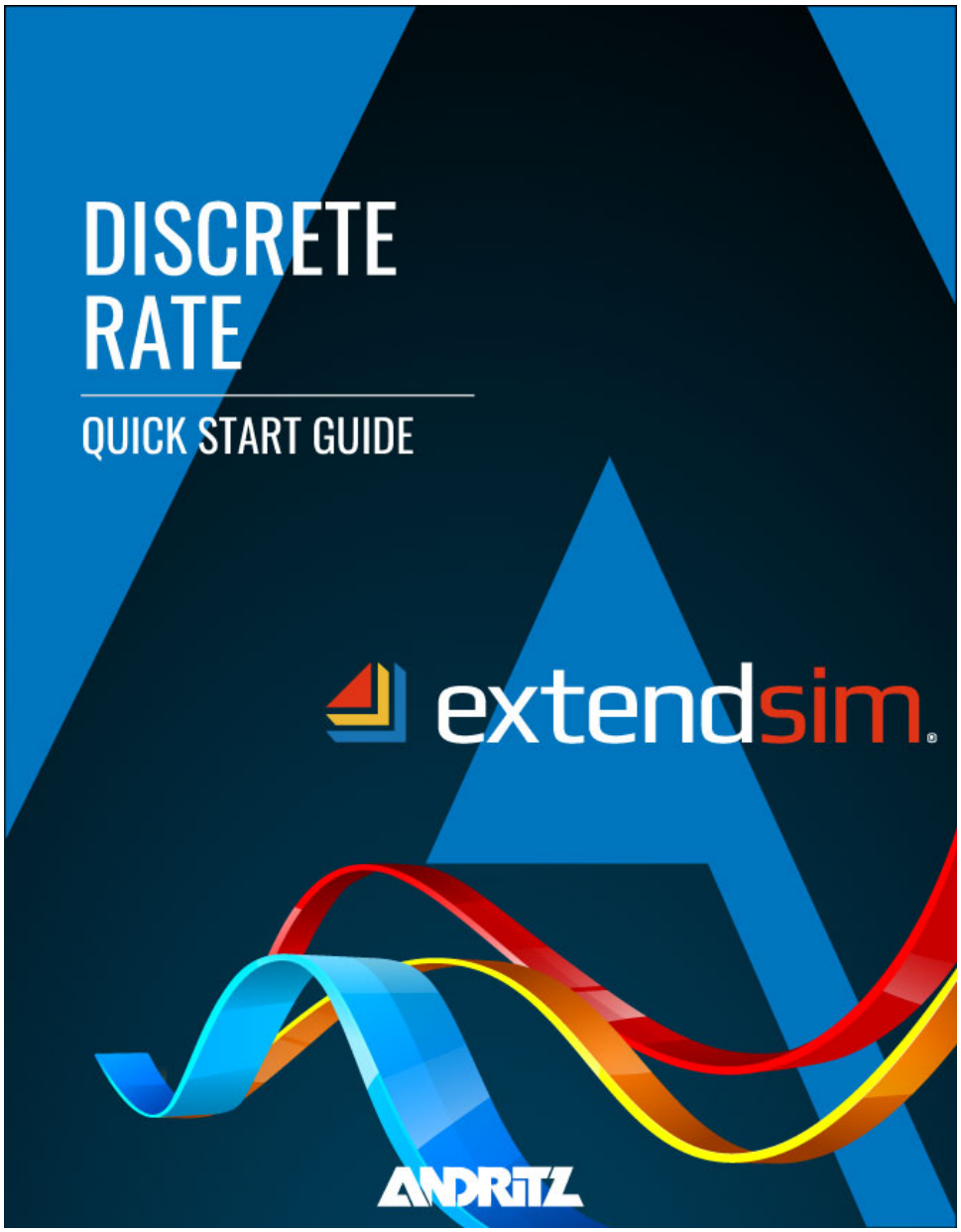
DISCRETE RATE

QUICK START GUIDE



© 2026 ANDRITZ Inc. This program is protected by US and international copyright laws.

You may not copy, transmit, or translate all or any part of this document in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than your personal use without the prior and express written permission of ANDRITZ Inc.

License, Software Copyright, Trademark, and Other Information

The software described in this manual is furnished under a separate license and warranty agreement. The software may be used or copied only in accordance with the terms of that agreement. Please note the following:

> ExtendSim blocks and components (including but not limited to icons, dialogs, and block code) are copyright © by ANDRITZ Inc. and/or its Licensors. ExtendSim blocks and components contain proprietary and/or trademark information. If you build blocks, and you use all or any portion of the blocks from the ExtendSim libraries in your blocks, or you include those ExtendSim blocks (or any of the code from those blocks) in your libraries, your right to sell, give away, or otherwise distribute your blocks and libraries is limited. In that case, you may only sell, give, or distribute such a block or library if the recipient has a valid license for the ExtendSim product from which you have derived your block(s) or block code. For more information, contact ANDRITZ at Info.ExtendSim@Andritz.com or Support.ExtendSim@Andritz.com.

© 2026 ANDRITZ Inc. This program is protected by US and international copyright laws. Microsoft is a registered trademark and Windows is a trademark of Microsoft Corporation. All other product names used in this manual are the trademarks of their respective owners. All other ExtendSim products and portions of products are copyright by ANDRITZ Inc. All right, title and interest, including, without limitation, all copyrights in the Software shall at all times remain the property of ANDRITZ Inc. or its Licensors.

Acknowledgments

Extend was created in 1987 by Bob Diamond; it was re-branded as ExtendSim in 2007.

The contents of this document are the result of years of work by software architects, simulation engineers, and technical writers and editors of ExtendSim products.

ANDRITZ Inc • 13560 Morris Road, Suite 1250 • Alpharetta, GA 30004 USA
408.365.0305 • Info.ExtendSim@Andritz.com
www.ExtendSim.com

Table of Contents

Introduction **3**
- Welcome! 3
- About the Quick Start Guides 3
- Who should read this Quick Start Guide 3
- Chapters in this Quick Start Guide 3
- What is Discrete Rate simulation? 3
- Why simulate using DRS? 4
- What is modeled and where is DRS used? 5
- How ExtendSim is being used for DRS 5
- How DRS compares to other modeling methodologies 8
- Where to get more information 9

Model Basics **11**
- Overview 11
- Opening and exploring a model 11
- Model basics 14
- Blocks 14
- Connectors and connections 16
- Libraries 18
- Blocks for building discrete rate models 19

Tutorial Part I **23**
- Overview 23
- A simple rate-based process 23
- Launch ExtendSim 23
- Set the simulation parameters 24
- Blocks used for the simple Yogurt Production model 25
- Placing and connecting blocks on the model worksheet 26
- Enter dialog parameters and settings 29
- Configure the Line Chart block 30
- Your tutorial model 31
- Running a simulation 31
- Verify and validate results 32
- Save the model 33

Tutorial Part II **35**
- The goal 35
- Model assumptions 35
- Start with the model from the previous chapter 36
- Add a second line for processing the fruit 36
- Mix the flows based on the recipe 37
- Package in containers 38
- Model verification and validation 39
- The final Yogurt Production model 40
- Enhancing the user interface 41
- Next steps 42

Concepts & Terminology **45**
- Overview of discrete rate models 45
- Terminology and architecture 46
- The User Reference and example models 48

Index **49**

Discrete Rate Quick Start

Introduction

Welcome!

Welcome to ExtendSim — an advanced platform for building and analyzing simulation models, with powerful tools for insight and decision-making.

About the Quick Start Guides

The purpose of a Quick Start Guide is to get new users quickly familiar with a specific ExtendSim module and aware of the ExtendSim features and capabilities. There are three Quick Start Guides—Continuous Process Modeling, Discrete Event Simulation, and Discrete Rate Modeling. Depending on the product purchased, one or more of these will be available within ExtendSim via the Help menu, in the installed Documentation folder, and online at the ExtendSim website for the latest updates, examples, and videos.

✦ Each of the Quick Start Guides has similar structure and information, so it is not necessary to read all of them.

For more in-depth information, this guide refers to the User Reference and other documentation available in the Help menu and Documents/ExtendSim/Documentation folder.

Who should read this Quick Start Guide

This guide is for model developers who build *discrete rate models*. These models simulate systems that involve tanks, levels, and valves and the storage and rate-based movement of system components. In these systems, state variables (which describe the system at any point in time) change only when events occur and it is not important to individually identify each system component.

Chapters in this Quick Start Guide

1) Introduction to Discrete Rate Simulation (DRS):
    * What DRS is
    * Where and when to use DRS
    * How it is being used
    * How DRS compares to continuous process and discrete event simulation
    * Where to get more information
2) Model basics and how to run a model
3) Tutorial I: building a simple discrete rate model
4) Tutorial II: expanding the model
5) Concepts and terminology specific to discrete rate modeling

What is Discrete Rate simulation?

Discrete rate simulation (DRS) is the methodology used to simulate the behavior of a rate-based system or process where the rates are affected by events. Typically these rate-based systems either function at a *high speed* or involve the movement of *large volumes* of bulk material and include tanks, levels, and valves. However, discrete rate simulation is useful for modeling the movement and storage of any type of *aggregated*, rather than individually identified, substance or material.

In these types of processes, system components (homogeneous “stuff” or heterogeneous “things”) are stored in and move through the system. Flow rates determine how fast components move and, until an event occurs, the components move at constant rates. When an event happens, the flow rates and the quantities at each storage location are recalculated.

✦ In a DRS model, flow rates remain constant over time until an event occurs. At each event, the rates are recalculated and the state of the system changes.

A common DRS model involves quantities of “flow” (material, product, fluids, gases, data, and so forth) either located in one or more parts of the model or moving between those locations at constant speeds—their effective rates. As events occur during the simulation run, the effective rates are recalculated based on model constraints and other factors. The movement between blocks that store or route the flow follows the paths, rules, and constraints that are set in the model to simulate the real world conditions.

Rate-based models are mainly focused on how quickly the flow moves in different sections of the model and what the yields will be, given the model’s constraints and configurations.

✦ While rate-based systems are typically high speed or involve large volumes of bulk material, almost any system or process that involves a quantity of something that is stored at one or more places, then moves or is transferred to another location at an effective rate per time unit (gallons per second, cans per minute, and so forth), can be simulated using discrete rate modeling.

Why simulate using DRS?

The ExtendSim Discrete Rate module has been specifically designed to simulate the behavior of bulk flows and high speed processes so that engineers and analysts in mining, energy, consumer products, process industries, and many other fields can solve problems and quickly get answers.

A natural fit

A discrete rate approach to modeling is well-suited for simulating high-volume bulk flows, high-speed processes, and aggregated materials or substances because they have a lot in common. They:

- Often require specialized equipment
- Have high operating costs
- Are subject to maintenance issues and shutdowns
- Have complex system interactions which are further compounded by the presence of discrete and continuous flows of material at different steps in a process

✦ For a comparison of discrete rate, discrete event, and continuous process methodologies, see “How DRS compares to other modeling methodologies” on page 8.

Advantages

The advantage of discrete rate simulation is that it allows you to:

- Perfect processes and products
- Plan resource capacity by determining the rate at which products are being processed or sold
- Test various schedules to maximize process efficiency
- Identify bottlenecks and determine throughput

- Evaluate system reliability
- Fine-tune distribution control strategies
- Study system constraints and the impact of rate changes
- Evaluate plant configurations
- Analyze the effect of processes on the internal and external environment

What is modeled and where is DRS used?

Discrete rate modeling is used to model things in the aggregate, rather than as individually identified entities.

Commodities

Discrete rate simulation is most often used to model two diverse types of commodities:

- Homogeneous "stuff"—commodities where the components are indistinguishable from each other, such as *powders*, *liquids*, *data streams*, *gases* and other *fluids*
- Heterogeneous "things"—commodities whose components are so numerous that it would be inconvenient or overwhelming to model them individually, typically in industries that produce, mix, fill, or package products on high-volume or high-speed lines

Typical industries and their commodities

Industry	Stuff (bulk flows)	Industry	Things (high-speed/high-volume)
Petrochemical	Petroleum oil, natural gas	Food & Beverage	Tea bags, boxes of cereal, soda cans, yogurt containers
Mining	Coal, gravel, potash	Pharmaceutical	Bottles of aspirin, containers of salve
Hydrology	Fresh or sea water, waste water	Health & Beauty	Bottles of lotion or shampoo, diapers, tubes of lipstick
Pharmaceutical	Pills, tablets, capsules	Milling	Carpet, paper, cardboard, textiles
Manufacturing	Laundry detergent, chemicals	Network traffic	Samples, messages, packets
Power transmission	Electricity	Manufacturing	Metal cans, glass jars, buttons, tires
Communications	Information, data		

In other words, DRS is used to model pretty much anything that can be expressed as a volumetric flow rate (quantity divided by time), such as gallons per second, cans per minute, cubic feet per day, and so forth.

How ExtendSim is being used for DRS

The ExtendSim Discrete Rate Simulation technology was developed two decades ago. It was a revolutionary advancement at that time and continues to be the leading discrete rate simulation technology available today. Building upon numerous customer experiences and taking advantage of newer methodologies, the ExtendSim DRS technology is now in its third generation. As you might expect, it has been used extensively in industry, especially for:

- Pit-to-Port mining operations
- Pulp and paper mills
- High-speed manufacturing
- Water and wastewater management
- Petrochemical processes

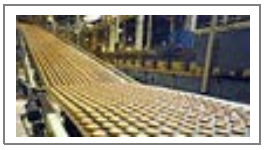
✦ As will be seen in the Discrete Rate Tutorial, it is common for discrete rate models to also include portions that represent discrete event processes.

Some examples where DRS has been used include:

- Major chemical companies use ExtendSim DRS to simulate chemical plant operations and identify the processes that limit plant production. DRS captures plant dynamics and successfully identifies different bottlenecks for each product. The models are also used to evaluate the costs and benefits of proposed solutions and to improve the production capability of the plants.

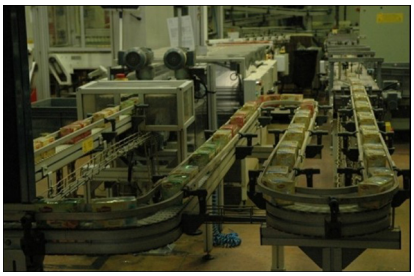
- Mining companies use ExtendSim DRS to develop pit-to-port and mine-to-market models of coal supply chains. A model at one international mining company tests stockpile management strategies in a materials handling facility (MHF) and analyzes competing capital investment options for moving coal from the MHF to a rail head.

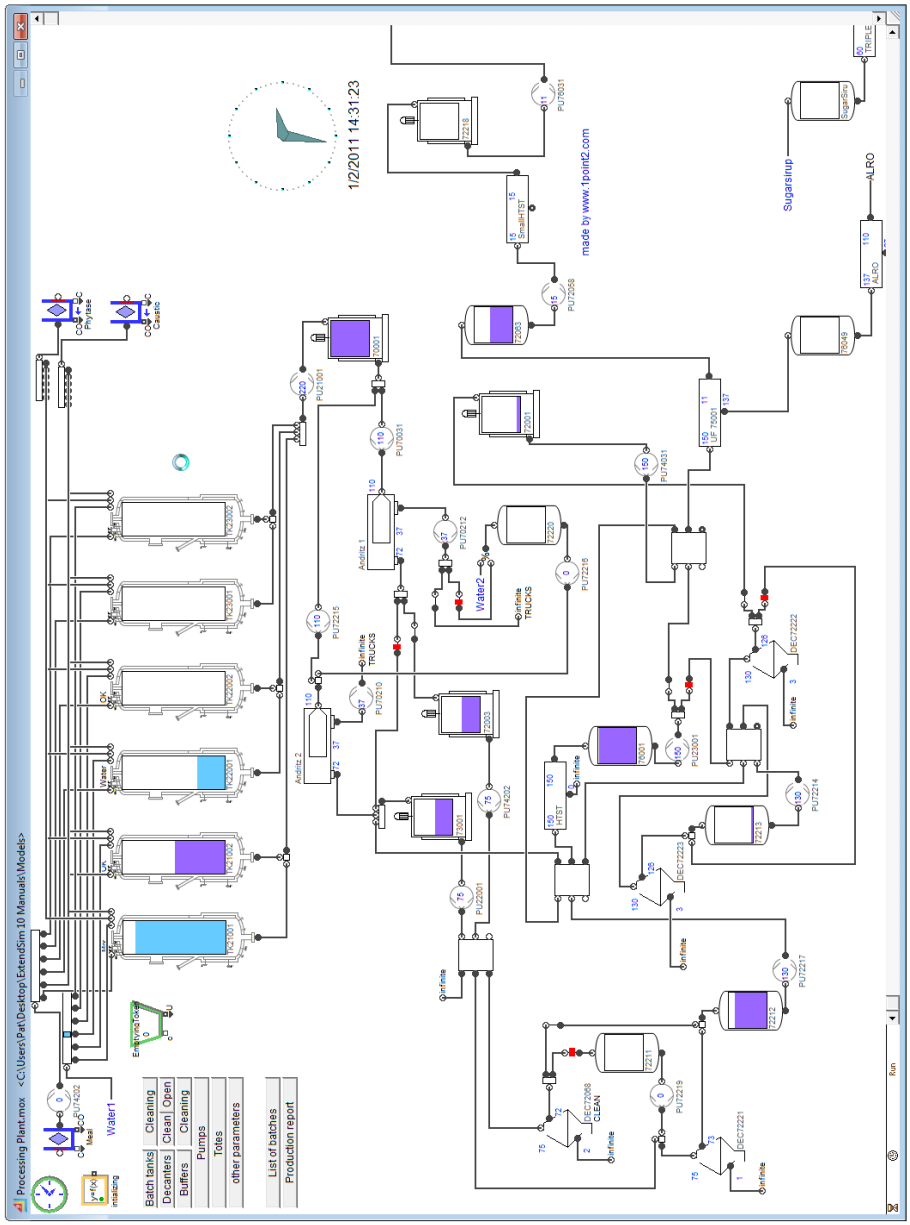
- Consumer products companies use ExtendSim DRS to model and analyze their high speed production lines and to test production scheduling as well as different production speeds for bulk material. One multinational company simulates the high speed production and packaging of laundry detergent. This allows them to test production schedules when adding new products and helps them manage intermediate stock between production and packaging. Another company performs reliability studies and tests production policies (start and stop times, batch levels, rework, etc.) based on the type of bulk material.



- A commodities producer uses ExtendSim DRS to simulate production, harboring, and shipping routes for liquid natural gas. The model helps determine the number and size of ships required as well as the amount of on-land storage needed. The impact of the external environment, including wind levels, visibility, and so forth, is included as part of the simulation.

- A major American food company uses DRS to simulate the production of peanut butter. The model explores various configurations for the company's conveyor belt system, including expected and unexpected shutdowns, and conveyor length, speed, and capacity. The goal is to reduce bottlenecks and maximize throughput. Different systems supplying and demanding material at various rates, quantities, and operational shifts introduces complexities the model readily solves.





Processing Plant model, developed for a customer by 1Point 2 of Seyssinet-Pariset, France
This plant's processes are extremely water-intensive, so it has its own internal water recycling system. When necessary, the plant can get water from the local community, but at a high cost. 1Point2 used the ExtendSim Rate library to optimize the plant's processes so that a minimum amount of outside water is required.

How DRS compares to other modeling methodologies

In some situations, simulating a system using discrete rate modeling is a more natural fit compared to using continuous or discrete event modeling:

- Processes that are event driven, rather than time driven, do not lend themselves to continuous simulation.
- Systems where there is no "entity" that can be identified, or when there are so many entities that individual identification is meaningless or unwanted, can be more naturally represented using a rate-based approach. In addition, rate-based models run faster than discrete event models and are better suited to thinking in terms of flows, tanks, rates, and so forth.

Most importantly, discrete rate modeling allows you to simulate the performance of high speed processes and bulk material flows in a way that can't be duplicated using discrete event or continuous modeling.

Comparison chart

The most important aspects of the three main modeling methodologies—continuous, discrete event, and discrete rate—are shown below.

	System State Changes When	Models	Attributes for Tracking	Queued Order Method
Continuous Process	Time changes	Generic stuff	No	N/A
Discrete event	Event occurs	Identifiable things	Yes	FIFO, LIFO, Priority, or a custom order based on Item Attributes
Discrete rate	Event occurs	Generic or aggregated stuff	Yes	FIFO, LIFO, or a custom order based on Flow Level Attributes

- State change. Models are either event-based or time-based. Discrete rate models only change state when an event occurs; the same is true for discrete event models. Continuous process models have a fixed time step where time advances in equal increments and the model changes state at each time step.
- What is modeled. Discrete rate and continuous models both model homogeneous or heterogeneous "stuff", while discrete event models track individually identifiable "things"—entities or items.
- Tracking. Quantities of flow in discrete rate simulations can be assigned attributes—properties that allow a value to be individually identified and manipulated; this is also true for the entities in discrete event models. However, values in continuous models do not have attributes.
- Order method. The order in which the flow in discrete rate models or the things in discrete event models leave a queue which is holding more than one quantity or thing. The stuff (values) in continuous models cannot be put into any order.

✦ For more information about how discrete rate modeling differs from continuous and discrete event modeling, see the User Reference chapter on Modeling Methodologies.

Where to get more information

The ExtendSim documentation, example models, and the video files and documents on the www.ExtendSim.com website provide comprehensive help.

Tutorial & Reference documents

In addition to the Quick Start Guides, there are three Tutorial & Reference documents that are included as eBooks in the Documents/ExtendSim/Documentation folder:

- *ExtendSim Database*. This internal relational database provides model developers with a systematic way to manage information for the model and makes models scalable.
- *Advanced Resource Management*. ARM is a sophisticated architecture for systematically dealing with multiple types of resources and a model's complex resource requirements.
- *Reliability*. Graphically capture and validate complex availability behavior. Determine when scheduled and unscheduled downs occur for individual resources and what impact that has on the availability of the system as a whole.

User Reference

The ExtendSim User Reference has a lot of information you will find helpful when building, using, and presenting models.

How To chapters cover general modeling and simulation topics

- Using libraries and blocks
- Performing analysis
- Enhancing presentations
- Creating a user interface
- Using equation-based blocks
- And much more

Discrete Rate chapters in the User Reference discuss specific topics:

- Creating flow
- Rates, constraints, and movement
- Merging/diverging, and routing flow directly and remotely
- Delaying flow using goals, hysteresis, and the Convey Flow block
- Mixing flow and items
- Advanced topics such as LP technology, upstream supply and downstream demand, and messaging in discrete rate models.

Appendices list menu commands and the ExtendSim libraries and blocks

Every menu command is explained; the main libraries are described block by block.

Technical Reference

You probably won't build your own discrete rate block, but it's very common to use an Equation block (Value library) in a model. The Technical Reference lists over 1,000 functions and has information about using include files and other programming tools.

- The eBooks ship with the appropriate ExtendSim product. To access these documents, see the Documents/ExtendSim/Documentation/folder or launch the books from the Getting Started model that opens when ExtendSim launches. The User Reference and Technical Reference are also available if you select the Help menu when using ExtendSim.

Example models and videos show you how

ExtendSim includes numerous tutorial models as well as videos and example models that explain concepts discussed in the User Reference. For example models, see the Documents/ ExtendSim/Examples/ Discrete Rate folder. For videos, see the ExtendSim website.

Discrete Rate Quick Start

Model Basics

Overview

This chapter shows how to open and run a model of a simple discrete rate system and discusses the components of a model.

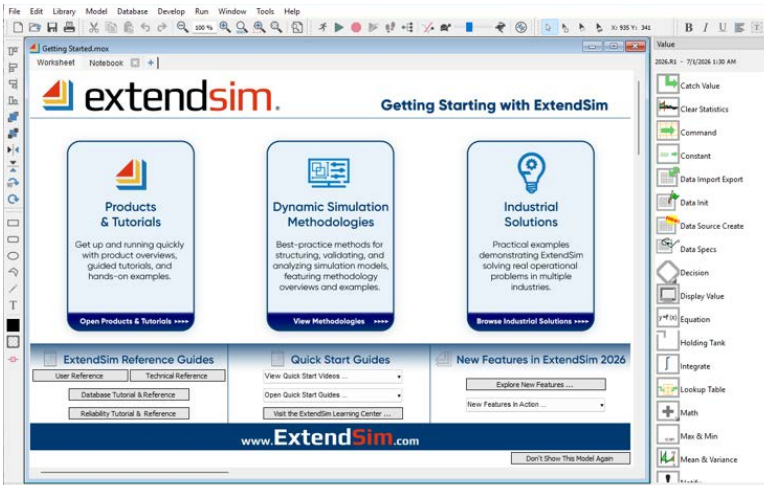
⚠ If you already know how to open, run, and save an ExtendSim model and are familiar with blocks, connectors, and model layout, skip this chapter and go to the next one.

Opening and exploring a model

▸ Launch ExtendSim

As seen on the right, when ExtendSim launches it opens an application window with:

- A model (on the left)
- Library windows (stacked on the right) for every open library
- Toolbars on the top and sides



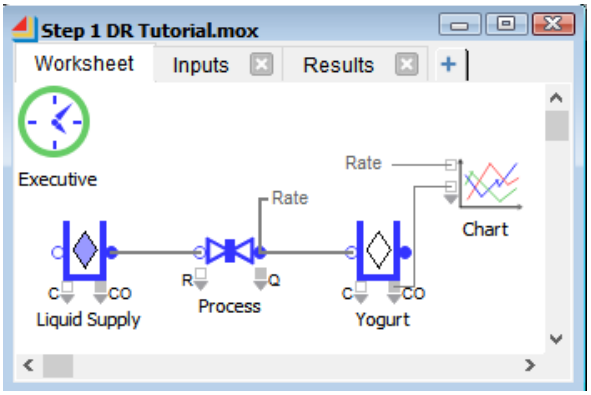
By default, the model that opens will be the Getting Started model. If that option has been removed from the Edit > Options > Model tab, a new blank model will open instead.

- The Getting Started model has information about ExtendSim products as well as links to the manuals, video tutorials, and more.
- If instead a new model opens, in most situations it will open with an Executive block in the upper left corner. This is discussed in “Launch ExtendSim” on page 23.

This tutorial will focus on the model window. Other aspects of the application window are discussed later in this Guide or in the User Reference.

Open the example model

- Select File > Open
- Browse to Documents/ ExtendSim/Examples/**Tutorials**/ Discrete Rate
- Select the model named *Step 1 DR Tutorial.mox*
- Click Open



This opens the *model window* as shown on the right.

About the model

This is a simple example of yogurt production. The model uses five blocks (labeled Executive, Liquid Supply, Yogurt Process, Yogurt, and Chart) to simulate a simple discrete rate system where a liquid mix is processed into yogurt.

The model window

Notice that there are tabs on the top of the model window:

- The *Worksheet* is where the model resides.
- The other tabs are *notebooks* that provide central places for storing inputs, outputs, and so forth; notebooks are discussed on page 13.

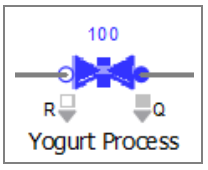
Run the example model

- Select Run > Run Simulation or click the Run Simulation button in the tool bar.

As the simulation runs, progress information will be displayed in the status bar at the bottom left of the primary window. (With a small model like this, and with animation turned off, the messages may go by too quickly to be read.) Since it has been set to do it, at the end of the simulation run the Line Chart block opens and displays the graph.

Run with animation

ExtendSim provides built-in and customizable 2D animation on block icons and on the worksheet. To see 2D animation in this model:



- Select Show 2D Animation in the Run menu or select the Show 2D Animation tool
- Run the simulation again

With animation on, the three Rate library blocks in this model (labeled Liquid Supply, Yogurt Process, and Yogurt) display appropriate aspects of the flow, such as the direction it takes, the amount that is present, or the rate of flow as shown on the Valve block labeled Yogurt Process.

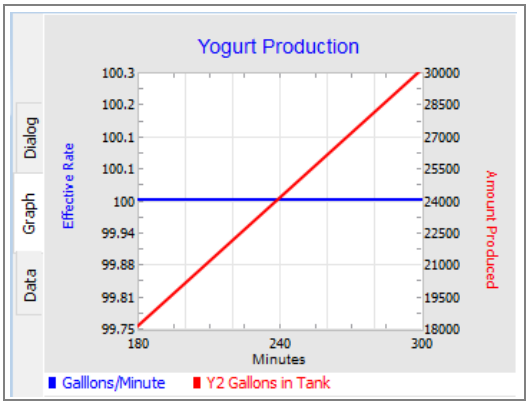
See results

In addition to what animation shows, there are many ways to see information during and at the end of the simulation run. For this model, information is reported on graphs and data tables, in notebooks, in block dialogs, and (as seen above) through animation.

On graphs and in data tables

ExtendSim comes with a number of flexible graphs and charts to use in models; this model uses a Line Chart block.

The example model runs for a simulated 480 minutes. At the end of the simulation, ExtendSim opens the Line Chart block and displays the history of the values over time on the block's Graph tab.



The Line Chart block in this model has information entering two of its input connectors, so the Graph tab displays two lines. The graph also has two value axes, each using a different scale. The graph's legend indicates which axis is used for each line (Y2 indicates the values are plotted on the right axis) and what each line represents.

In this case, the blue line is matched with the left axis to show the Yogurt Process's (Valve block) effective rate over time. The red line uses the right axis to display the number of gallons of yogurt produced.

480.00000000	100.00000000	480.00000000	48000.0000.

The Line Chart's Data tab shows the data points, seen at left, which produce the line.

The Line Chart's Dialog tab has options for when to show the graph, when to autoscale, and so forth.

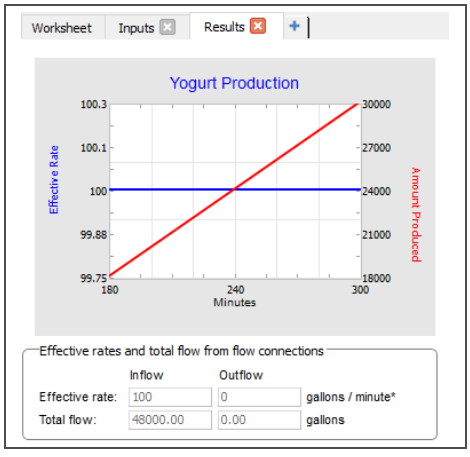
✦ To access windows for customizing how the graph is displayed and how the lines are drawn on it, right click on the Graph tab. To tear off one of the Line Chart's tabs so it is in a separate window, click and drag the tab by its name. Click a torn-off tab's close box to return it to the Line Chart.

In Notebooks

Notebooks are customizable tear-off windows that can be used as a "dashboard" for the model - to control model parameters, report simulation results, and document the model.

Some important parameters and tables from the dialogs of blocks in the yogurt model have been *cloned* (duplicated) onto two notebooks, one for inputs and one for results.

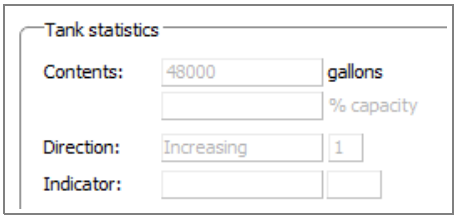
The graph and some of the results from the Tank block labeled Yogurt have been cloned onto the Results notebook shown here.



✦ Cloning dialog items is discussed in the How To: Custom Interfaces section of the User Reference.

From the model's blocks

Information can often be found on block icons, in block dialogs, and at block connectors. For example, when the icon of the Tank block (labeled Yogurt) is double-clicked, its dialog opens and its Results tab indicates that 48,000 gallons of yogurt were produced. And if 2D animation is on when the simulation is run, the icon of the Valve block (labeled Yogurt Process) indicates that the output is 100 gallons/minute.



Using other blocks

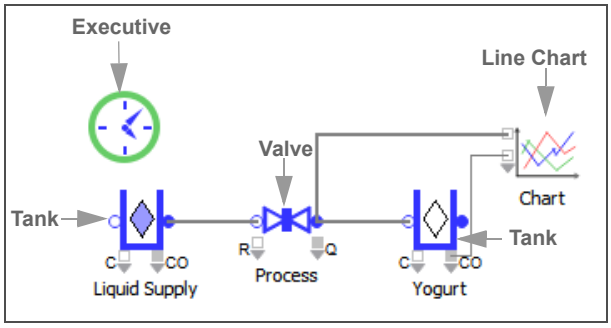
This model is so simple that it doesn't need further analysis. However, for larger and more complex models, the blocks in the Report library gather information as the model runs and display and calculate statistics on the results.

Model basics

In the simplest terms, ExtendSim models are composed of *blocks* and *connections*. This model, for example, has five blocks, seen on the model worksheet below. As the model runs, information goes into a block, is processed and/or modified, and is then sent on to the next block, usually via a connection. Except for some hierarchical blocks, blocks are stored in repositories called *libraries*.

Blocks

Each block in ExtendSim represents a portion of the process or system that is being modeled.



- Blocks have *names*, such as Tank or Valve, that signify the function they perform. A Tank block, for example, will have the same functional behavior in every model you build.
- You can also add a *label* to a block to indicate what it represents in your specific model, such as a Tank block labeled Liquid Supply.
- Blocks come from libraries, which are discussed in "Libraries" on page 18.

Icons

A block's icon is usually a pictorial representation of its function. For instance, in this model the block labeled Liquid Supply is a Tank block. Its icon symbolizes a holding tank that can have quantities added or removed from it.

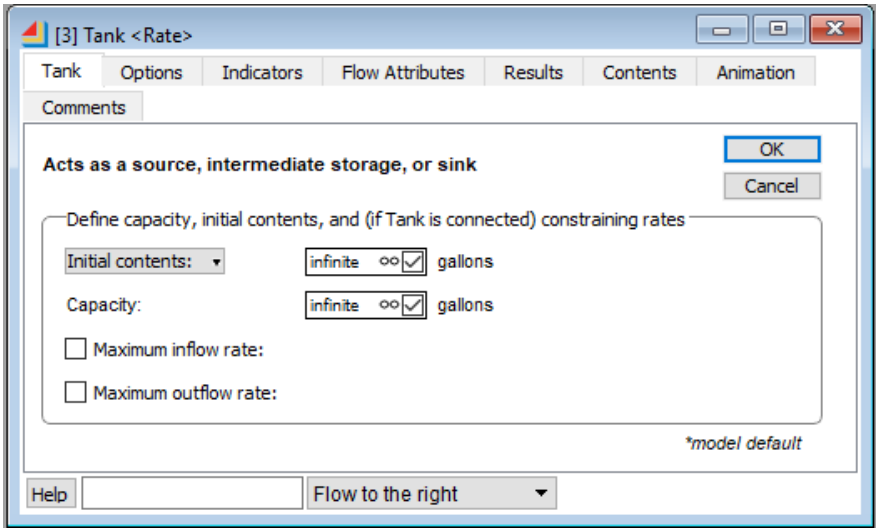
The small circles and squares attached to icons are *connectors*, which are discussed in more detail in "Connectors and connections" on page 16.

✦ Place your cursor over a block's icon on a model worksheet to see a tooltip with the block's object ID, name, and library. If *Include additional block information* is checked in the Edit > Options > Model tab, the tooltip will also include a short description of the block's function.

Dialogs

Most blocks have a dialog associated with them. Dialogs are used to enter values and settings before running simulations and to see results during and after the run.

To access a block's dialog, double-click its icon or right-click the icon and select Open Dialog.



For example, if you double-click the Tank icon, the dialog at right opens.

At the top is the block's global object ID, its name, the name of the library it resides in (inside the angle brackets), and its label (if present). Global object IDs are unique identifiers assigned sequentially to blocks and other objects as they are placed in a model.

✦ When there are multiple models and several open dialogs, it can be difficult to remember which block comes from which model. To see which model a block is in, hover over the block's name in the dialog's title bar.

Dialog tabs

Block dialogs may have tabs on top and/or on the left side.

- Most dialogs have tabs across the top for entering values and settings, getting results, controlling animation, and so forth. As shown above for the Tank block, the tabs are Tank, Options, Indicators, Flow Attributes, Results, Contents, Animation, and Comments.
- If present, tabs on the left side are for switching between windows. For example, the Line Chart has three tabs on the left (Dialog, Graph, and Data).

The tabs can be detached from the block to facilitate viewing the settings.

Interactive dialog items

Some dialogs calculate and display values that are generated as the model runs. If you leave a dialog open during the simulation, it will slow the run but you can watch the impact on different variables. This *interactive simulation* capability means you can even change some of the settings in a dialog during a simulation run, such as choosing different buttons or entering new values.

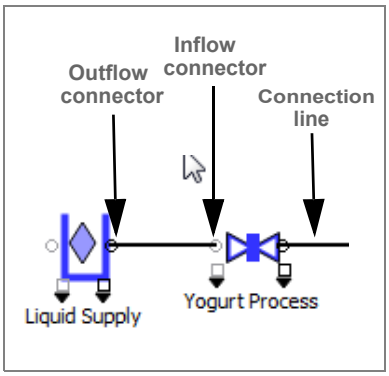
✦ When you click a button while the simulation is running, the block gets that changed value on the next event. However, if you type text or enter numbers into a parameter field or dialog table, the model pauses while you are typing in order to get your entire input.

Help, label, and view



As is true for most blocks, at the bottom of the Tank's dialog is a Help button. It provides information about the block, such as its purpose and use, connector usage, descriptions of each dialog item, and so on. To the right of the Help button is a text box for entering a label for that specific instance of the block in the model. The View popup is for changing the icon's orientation or appearance when you build models—for example, the Tank offers a choice of *Flow to the right* and *Flow to the left*.

Connectors and connections



- *Connectors* are used to get data into and out of a block during a simulation run
- *Connections* are the mechanisms used to transfer data and information between blocks

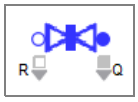
Connectors

Most blocks in ExtendSim have input and output connectors (the small circles or squares attached to the block's icon). As you might expect, information flows into a block at input connectors and out of the block at output connectors.

In ExtendSim, the behavior of most connectors is predefined for each specific block. For example, if you set a Valve block to have a specific maximum rate, it knows what to do with the values that are input into the block. This makes model building easy since you can connect blocks and run simulations without having to write equations or logic to define what each block should do with the inputs or outputs.

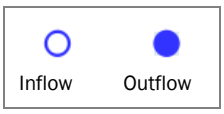
A block can have many input and/or output connectors; some blocks have none. In addition, a block can have single connectors or arrays of connectors depending on how the block is constructed.

Connectors in the Rate blocks



The blocks in the Rate library have primary connectors for flow movement as well as secondary connectors for getting and passing values.

For instance, the Valve block labeled Yogurt Process has two flow connectors and two value connectors:



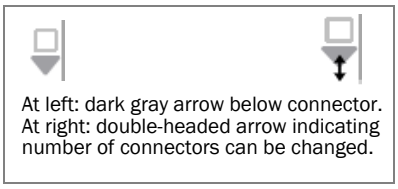
- The blue circles on the left and right sides of the Valve's icon are for *flow input and output*. Each Rate library block has at most one inflow and one outflow connector.
  - The blue outlined *inflow* connector on the left is for flow to enter.
  - The solid blue *outflow* connector on the right is for the flow to exit the block.



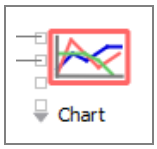
- *Value* connectors are represented by gray squares and are used for getting and reporting values. Unlike flow connectors, they can be a variable connector or a single connector.Variable connectors act like an array of single connectors that can be expanded or contracted to enable a required number of connections. As shown here, a variable connector is usually designated by a dark gray arrow. The Valve has two value-type variable connectors:
  - The gray outlined variable *input* connector on the bottom left of the Valve's icon can be used to control specific block behavior, such as its maximum rate of flow.
  - The solid gray variable *output* connector on the bottom right of the icon is for providing specific results.

Expanding a variable connector

To expand the connector array for a variable connector:



- Hover over the dark gray arrow that is below the bottom input connector on the left side of the Line Chart block. (Don't hover over or click on the input connector, just hover over the gray arrow.) If you are in the correct spot, the cursor will turn into a dragging cursor—a double-headed up/down arrow—indicating that the array of connectors can be resized.



- Once the cursor turns into the double-headed arrow, click and drag down on the arrow until the desired number of connectors are exposed, as shown here. As you drag, more connectors appear and the number of connectors is displayed.

✦ See the User Reference for more information about variable connectors, including how to collapse them.

Connector function

To get information about a connector's function, click the Help button in the bottom left-hand corner of the block's dialog or see the User Reference.

✦ Place your cursor over a connector to see its name and current value; that value will change as the model runs. You may also see additional information depending on how the block is programmed.

Connections

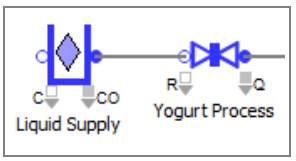
- The connections between inflow and outflow connectors represent the path of product from block to block.
- Connections between value connectors represent the flow of information.

The simulation itself is a series of calculations and actions that occur at each event.

Making connections

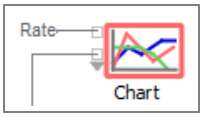
There are three ways to make connections between blocks:

1) *Line connections* physically join the output of one block to the input of another using connection lines.



   - The *Point and Click* method is used to physically draw a connection line between one block's output connector and another block's input connector (or vice versa).
   - When an ExtendSim block's connector is right-clicked, a special *Smart Block technique* can be used to add a block to the model and at the same time automatically connect it to the original block with a connection line.
   - For the Item and Rate libraries only, their primary connectors support *Bump Connect*. To use this feature, click on a block in the library window then bump its itemIn or inflow connector into the itemOut or outflow connector of a block already on the model worksheet. This places the new block on the worksheet and connects the blocks together.

2) *Named connections* use text labels as outputs and inputs, causing data to jump from the output to the input without using connection lines. The model has a named connection titled "Rate" that reports the Valve's effective rate of flow to the Line Chart.



3) The *Throw/Catch* feature allows remote connections, even between hierarchical levels; this is discussed in the User Reference.

Libraries

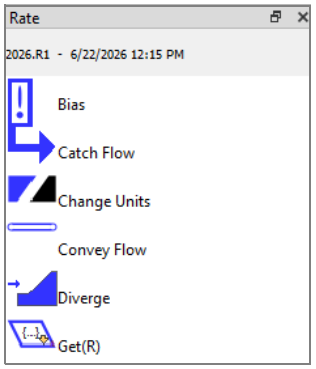
Libraries are repositories for the blocks used to build models. When you open an existing model, ExtendSim automatically opens the libraries that contain the blocks used in the model. Libraries are also automatically opened if listed in the table in the Edit > Options > Libraries tab.

Libraries can be manually opened, closed, and created in the Library menu. Open libraries are listed in alphabetical order at the bottom of the Library menu.

✦ Any data entered in the block's dialog is stored within the model, not within the library.

Library windows

Each library has an associated *library window*, listing its blocks and their names. By default when you open ExtendSim, a library window for each open library opens and docks on the right side of the primary window; to change that, see the Edit > Options > Libraries tab. The name of each library window is shown on the top of the docked window and in a tab along the right side.



Library windows behave like other docking windows. For example, to detach a library window from its docked position such that it becomes a floating window, double-click its title bar or click and drag to detach it. To re-dock a library window, click its close box, double-click its title bar, or click on its title bar and drag it back to the docking area.

Use the close box to close a docked library window. To easily close all the library windows, use the Library > Close All Library Windows command or the corresponding button in the

toolbar at the top of the windows. The Open All Library Windows command and button re-opens the library window for every open library.

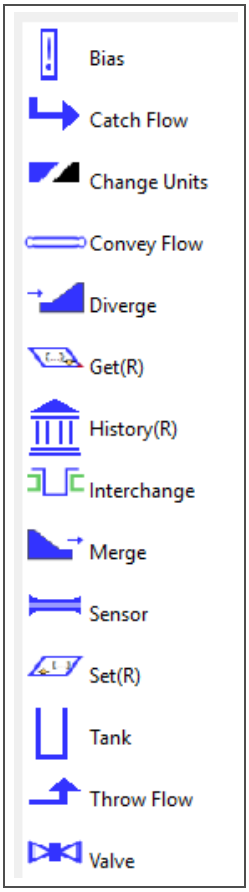
✦ Closing a library window does not close the library; use the Library menu to close a library.

Blocks for building discrete rate models

As discussed more below, to create discrete rate models you will primarily use blocks from the Rate library. However, it is common to also use blocks from other libraries that ship with ExtendSim, especially the Chart, Report, Item, and Value libraries.

Rate library

The Rate library allows you to simulate a wide range of flow systems just by connecting blocks together and entering parameters in block dialogs. The complexity of calculating the effective flow rate and the generation of events that dictate a new rate calculation are handled within the blocks, alleviating the need to do any programming.



Rate library blocks are optimized for modeling anything flowing through a system. They incorporate concepts like constraints, goals, flow prioritizing, mixing, batching, unbatching, level indicators, and so forth. These blocks have been designed to meet most rate-based flow needs so you can quickly and easily model complex high-volume/high-speed systems.

Custom discrete rate blocks

Because of the Rate library's extensive capability, it is unlikely that you would need to create your own discrete rate blocks. If you do want a custom discrete rate block, they are created the same way the blocks in the ExtendSim libraries were created—using the included ExtendSim IDE and its programming tools—then saved in libraries for use in all models. Two important notes:

- If you use an ExtendSim block as a basis for your custom block, be sure to save your block under a different name. Otherwise, your code will be replaced whenever ExtendSim updates.
- Store your custom blocks in a new library, not in an existing ExtendSim library. Otherwise, you will loose your custom blocks whenever ExtendSim is updated.

✦ See the Appendix in the ExtendSim User Reference for a listing and brief description of the blocks in the Rate library.

Item library

Discrete rate models often use discrete event blocks from the Item library; they are helpful for representing the entities—tankers, airplanes, people, and so forth—that interface with flow.

See the Discrete Event Quick Start Guide and the User Reference for more information about blocks in the Item library.

✦ Every discrete rate model must have an Executive block, from the Item library, on the left most side of the model worksheet.

Value library

Most models use one or more continuous blocks from the Value library. They are pre-programmed to simplify data management and model-specific tasks and allow you to perform complex modeling steps often with just the click of a button. For example:

- Use the Random Number block's popup menu to select one of 35 distributions.
- The Equation block is for creating compiled equations, from a simple operation to a full programming segment. It uses formulas and/or ModL code entered in its dialog to calculate values for, and perform operations in, models.
- For advanced modeling and integration with external tools, use the Python Bridge block. This block enables communication between ExtendSim models and Python, allowing you to incorporate Python-based calculations, data processing, and external libraries directly into your simulation workflow. While the Equation block is ideal for calculations within ExtendSim, the Python Bridge block extends this capability by enabling integration with external Python code and analytics.
- The Optimizer block facilitates optimization.
- The Scenario Manager block allows you to strategically explore the outcome of different model configurations and analyze alternatives.

See the Appendix of the User Reference for more information about blocks in the Value library.

✦ Using Value library blocks with Rate library blocks does not change the fundamental architecture of discrete rate models—they will still be event-based rather than use the time-based architecture of continuous models.

Chart library

The blocks in the Chart library are useful for any type of model. The most commonly used block is the Line Chart, which traces the history of values over time during the simulation.

Report library

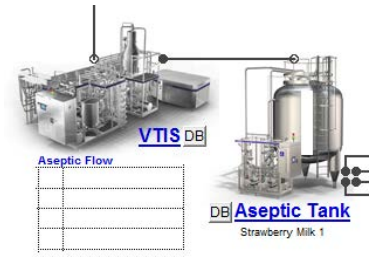
The blocks in the Report library are also useful for any type of model. They gather information as the model runs and display and calculate statistics on the results.

Utilities library

The Utilities library has several blocks that are useful for creating a custom user interface for models and for performing other tasks and activities. See the "How To" section of the User Reference for some examples.

Hierarchical blocks

If your model becomes too cluttered with blocks or you'd like it better organized, encapsulate portions of the model into a hierarchical block. To see the submodel, double-click the hierarchical block. Hierarchical blocks are created using simple menu commands; they can be copied from model to model and stored in libraries for reuse in other models.



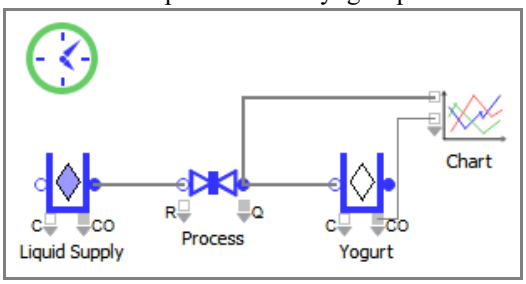
Two hierarchical blocks

Discrete Rate Quick Start

Tutorial Part I

Overview

This tutorial shows how to build the simple rate-based yogurt production model shown below.



⚠ It is important that you have read the previous chapter, or understand ExtendSim basics, before proceeding.

A simple rate-based process

When building any simulation model, it is best to start with a simplified subset of the process. Then continue adding detail, verifying and validating at each stage, until you arrive at a completed representation of the system. This will make the model building process more manageable.

About the model

This example simulates a single line of production, where one Tank holds product that moves to another Tank at a constant rate as set by a Valve. In this case, the model represents the process of making plain yogurt.

✦ This model will be similar to the model you opened and ran in the previous chapter.

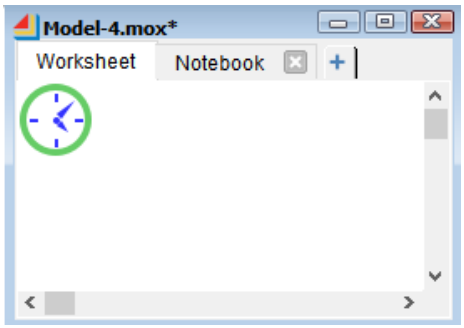
Assumptions

- The Liquid Supply tank has an unlimited supply of liquid for making yogurt
- The liquid is processed at 100 gallons per minute
- The Yogurt tank can store an unlimited amount of yogurt
- The model runs for a simulated time of 480 minutes (8 hours)

Launch ExtendSim

By default, when you launch ExtendSim the Getting Started model opens as well as the major ExtendSim libraries and their library windows.

Open a new model



- Use the toolbar button or the File menu to open a new model.

By default, two things happen:

1) A new model window, named Model-X, opens
2) An Executive block is placed on the left-most side of the worksheet

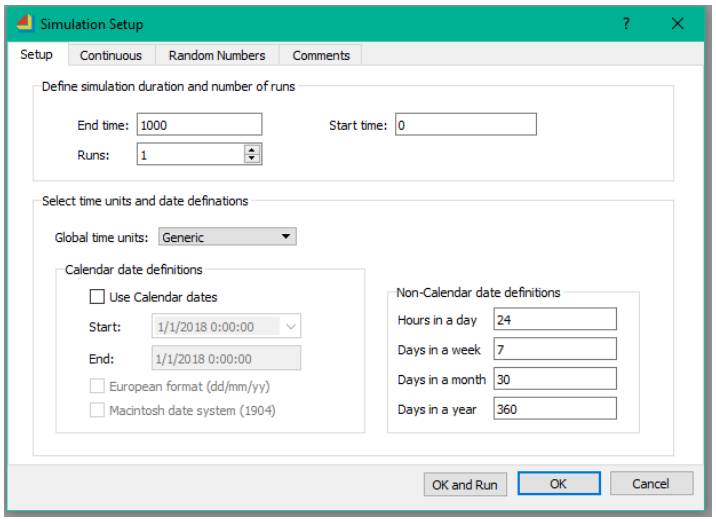
⚠ The Executive block will only be automatically placed in new models if the Item library is open.

If the major libraries have not been set to open in the Edit > Options > Libraries tab, you will need to manually open the Chart, Item, Rate, and Value libraries using the Library menu. Then place an Executive block (Item library) on the worksheet.

Set the simulation parameters

- Select the command Run > Simulation Setup (Ctrl/control J)

This command opens a window for entering global settings for the model, such as how long and how many times the simulation will run.



The most common simulation settings you will need to enter in the Simulation Setup window (and often the only ones) are located on the Setup tab: the *End time*, the number of *Runs,* and *Global time units*. For most purposes, you want the simulation to start at the beginning, so you would use the default start time of 0.

Use the model assumptions to set up the simulation run:

- In the Setup tab, enter the simulation parameters:
  - **End time: 480**
  - **Start time: 0** (default)
  - **Runs: 1** (default)
  - **Global time units: Minutes**
- Leave the other Simulation Setup settings at their defaults

▸ Click **OK** to close the window and save the changes

With these settings, the model will run for a simulated time of 480 minutes.

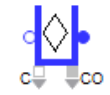
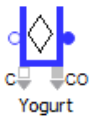
Save the model

▸ Choose File > Save Model As and name the file **My Yogurt Model**.

✦ Models with changes that haven't been saved have an asterisk (*) at the end of their name in the title bar. Each time you save a model, ExtendSim creates a backup, *ModelName.bak*, of your previously saved model. To recover using a backup file, change the name of the backup file to *ModelName.mox*.

Blocks used for the simple Yogurt Production model

The following table lists the five blocks used in the model.

Name (Label)	Library	Block Function	Purpose in Model
Executive 	Item	Does event scheduling and provides for simulation control, item allocation, attribute management, and more.	Must be present and on the left-most side in every event-based model. *Note: If the Item library is open, this block will automatically be added to new models as they open.*
Tank 	Rate	Acts as a source, intermediate storage, or sink for whatever is flowing through the model.	Is the source of an unlimited amount of liquids that can be processed into yogurt.
Valve 	Rate	Acts as a constraint on flow. Controls, monitors, and transfers the flow at a specified rate. Every discrete rate model must contain one or more constraining blocks with a non-zero maximum rate.	Regulates the flow of liquid at 100 gallons per minute. (A constraint is required in the model; otherwise, the flow would approach infinity!)
Tank 	Rate	Acts as a source, intermediate storage, or sink for whatever is flowing through the model.	Acts as a sink that can store an unlimited amount of processed yogurt.
Line Chart 	Chart	Displays information about the flow and about model values.	Reports how many gallons of yogurt per minute are processed (the effective rate) and the total amount of yogurt produced.

There is nothing fundamentally different about the structure of these different blocks. Any block may create, modify, or present information, and many blocks perform more than one of

these functions. You can, of course, have multiple instances of the same block (such as the Tank) within a model.

Placing and connecting blocks on the model worksheet

Methods for placing blocks in a model

There are two main methods for placing and connecting blocks in a model:

1) The Point and Click method is useful for adding any block anywhere on the model worksheet:

   - Click a block in the Library menu or library window to select it
   - Click somewhere on the model worksheet to place it
   - As needed, connect it to another block using one of the connection methods discussed on page 17

2) The Smart Block method is a quick way to add a new block to a model and connect it to an existing block at the same time:

   - On the model worksheet, right-click on one block's connector and, from the list, select the new block to add. This will place the new block next to the original and connect them.

You will use both methods for this tutorial. However, to use the right-click technique, a non-Executive block must already be on the worksheet to start the connection chain. So you will place the first block on the worksheet using Point and Click.

✦ A third method, Bump Connect, only works with the primary connectors on Item and Rate library blocks. See the User Reference for more information about Bump Connect.

Use Point and Click to place a Rate library block on the worksheet

To use Point and Click to place a **Tank** block (Rate library) on the model worksheet:



- Bring the **Rate** library window forward by clicking once on its name in the list of library windows on the far right of the application window
- In the Rate library window, click once on the **Tank** block's icon



- Then click on the model worksheet somewhere below and to the right of the Executive

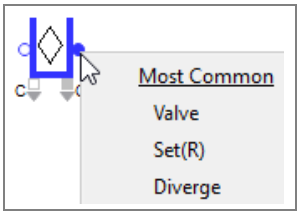
✦ If you select the wrong block, click the Esc (escape) key to unselect it. (You can also place multiple instances of the same block on a worksheet by holding down the Alt or Option key while repeatedly clicking on the worksheet.)

Use the Smart Blocks technique to place the remaining Rate blocks

Now, because it's an easy and fun way to add blocks to a model, use the Smart Block technique to place additional Rate library blocks on the worksheet.

⚠ Since Smart Block involves right-clicking on a connector, you must already have at least one block, other than the Executive, on the model worksheet for it to work. This technique works with the major ExtendSim libraries and, if they have been programmed to use it, with custom libraries.

Valve block



- Go to the model worksheet.
- Right-click on the Tank block's **outflow connector** (the solid blue circle on the right of the icon). A list of the blocks most commonly added to the Tank appears.
- Select the **Valve** from the list of common blocks. Note that the Valve block gets automatically placed on the worksheet and connected to the Tank.

Tank block

- Next, right-click on the *Valve* block's outflow connector and choose a second **Tank**. This places the new Tank on the worksheet and connects it to the Valve.

✦ While not needed here, the Smart Block technique can also be used to connect a second block to an output (hold down the Ctrl/Cmd key while right-clicking on the output) or to insert a new block between two connected blocks (separate the two blocks to provide room for the new block, then right-click on the output or input of one of the existing blocks to insert the new block).

Use Smart Block to place the Line Chart block on the worksheet

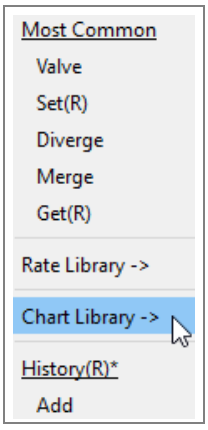
As noted in the table on page 25, there are two pieces of information you want displayed as the simulation runs:

- The effective rate of flow into the second tank—that is, how many gallons of yogurt per minute are actually being produced.
- The contents of that tank, which is the total amount of yogurt produced

The Line Chart block (Chart library) is useful for tracing the history of values during the simulation.

Report the rate of flow

To cause the Line Chart block to report the effective flow rate at the Tank, you will use the Smart Block technique to add the Line Chart block to the model and connect it in parallel with the existing connection between the Valve and the Tank.

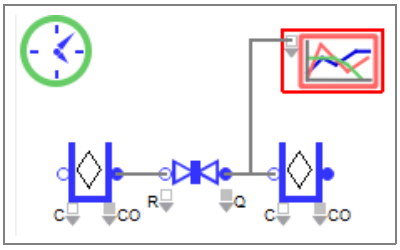


✦ Wondering why the connection isn't made on the outflow connector of the second Tank? Tanks serve as sources, intermediate storage, or sinks. At the end of the line, this Tank block is a sink and nothing flows out. Thus the effective rate at its outflow connector is 0.

- Hold down the **Ctrl/control key** while you right-click on **outflow** connector of the **Valve**.
- *While still holding down the Ctrl/control key,* click the **Chart** library in the popup that appears, as shown at right. This opens a new popup listing all the blocks in the Chart library.
- *While still holding down the Ctrl/control key,* select the **Line Chart** block in the secondary popup. This connects the Line Chart block to the Valve in parallel with its existing connection to the second Tank.

- Move the Line Chart up and to the right. If the connection line goes across the second Tank, you can fix it later.

✦ This process connects a block in parallel with an existing connection. To instead insert and connect a block in *series* between two connected blocks, move the second block away so there is enough space for insertion, then right-click (without using the Ctrl/control key) on the first block's output connector and select a block to insert. (Inserting a block in series requires that there be only one current connection line from the original block.)

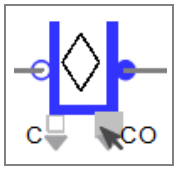


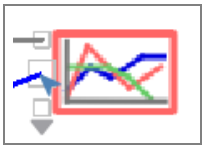
Your model should now appear similar to the one shown above.

Report the contents of the Yogurt tank

You could use the Smart Block technique to add a second Line Chart block to the model. However, you will instead draw a connection line to the existing Line Chart block so that it displays both the rate of flow and the amount of yogurt.

- Hover the cursor over the **CO** (contents) *output* connector, located on the bottom right side of the **Tank's** icon, until that connector enlarges.
- Click once on the CO output connector.
- Then move the cursor and place it over the second input from the top of the **Line Chart** block. That input connector enlarges.
- When the connection line turns thick and blue, click once on the Line Chart's second input connector to make the connection.





This connection line will cause the Chart to report how many gallons are in the Yogurt tank during the simulation.

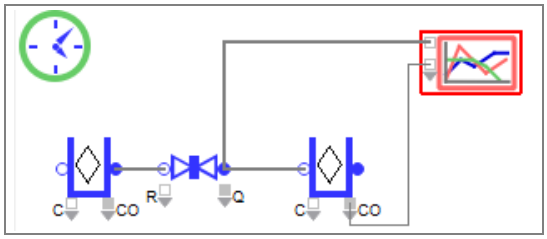
✦ Connection lines will turn solid dark gray when properly connected; they appear as a dashed red line if the connection has not been made correctly.

⚠ The blocks used in this model have variable connectors, but there hasn't been any need to increase the number of connectors that are exposed. For how to expand a connector array, see "Expanding a variable connector" on page 17.

Adjust the connections and blocks

By default, connection lines are drawn using a Smart Connection technique that provides intelligent connection line control when moving blocks around. If Smart Connections don't result in a good appearance for the finished model, you can change the connection line style and adjust the line.

- To change the connection line from the Valve's outflow connector so the line doesn't cross over anything:
  - Right-click on the connection line
  - Choose the Right Angle or Free Form Connection



- Move the line segments as desired by selecting and moving the anchor points.

- Reposition the blocks as desired.
- When finished, the model should look similar to the one shown above.

✦ To change connection line behavior and appearance in a model, go to the Model > Connection Line Style menu or use the right-click menu. To change the default connection line behavior for all models, use the Edit > Options menu.

Save the model

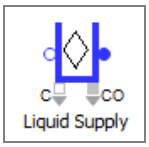
- Choose File > Save Model to save your changes.

Enter dialog parameters and settings

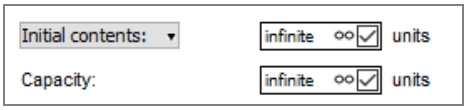
To reflect the basic assumptions for this model, the flow units and constraints need to be defined.

Tank that represents liquid supply

The assumptions state that the supply of liquid is infinite and flow units are expressed in gallons.

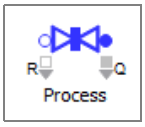


- To open the dialog of the first Tank block, double-click its icon or right-click on the icon and choose Open Dialog
- In the dialog's *Tank* tab:
  - In the field labeled **Initial contents**, check the ∞ (infinite) checkbox. This replaces the initial 0 and puts the word "infinite" in the initial contents field, as seen at right.



  - In the field at the bottom of the dialog, label the block **Liquid Supply**.
- In the Tank dialog's *Options* tab:
  - Leave the default setting as "Use identical flow units for the group and the block".
  - For the popup of the **Flow unit**, select **New Unit** and enter **gallons** as the flow rate.
  - Keep the default time unit "minute".
- Click OK to save changes and close the block's dialog.

Valve that represents the process



- In the dialog of the Valve block, enter **Maximum rate: 100 gallons/minute**
- Label the block **Process**
- Then close its dialog

Tank that represents yogurt

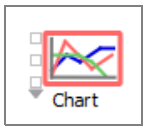
There are no parameters to change in this block because its default settings indicate that it has no initial contents and its maximum capacity is infinite, which follows the model assumptions.



- In its dialog, label the block **Yogurt**
- Then close its dialog

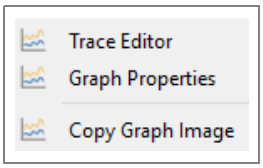
Configure the Line Chart block

As this model is constructed, the Line Chart will get the Valve's effective rate through its top input connector and the number of gallons of yogurt processed on the second input. However, since the scale for these numbers is so different, the graph needs to be adjusted.



Trace Editor

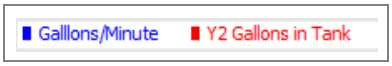
- Open the Line Chart block by double-clicking its icon or by right-clicking and choosing Open Dialog.
- Right-click on the block's Graph window. The two options at the top of the popup are Trace Editor and Graph Properties.
- Choose the Trace Editor. This is where you specify how the lines are drawn on the graph.



  - Name the first trace **Gallons/ Minute**.
  - Name the second input **Gallons in Tank**.
  - For the second input (Gallons in Tank) change the line style (the third tool after the name) to be interpolated. This allows the data points to be connected by diagonal lines, rather than plotting points or right-angle lines.

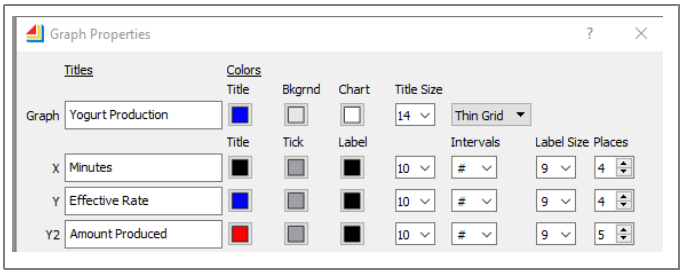


  - Also for the second input, click the Y1 button so that it changes to Y2, as shown here. This will cause the second input to be plotted against the Y2 axis.
  - Then close the Trace Editor window.
  - Notice that the new labels appear in the key on the graph. You can resize the key and place it anywhere you want on the graph.



Graph Properties

- Right-click on the Graph window again and this time choose Graph Properties. This is where you customize the titles, colors, and other properties of the axes, as well as characteristics of the numbers that are displayed.



- For the titles:
  - Name the graph **Yogurt Production**
  - Name the X axis **Minutes**
  - Name the Y axis **Effective Rate**

- Name the Y2 axis **Amount Produced**

- So that the expected number of gallons will display nicely, increase to **5** the number of Places for the numbers on the Y2 axis

- Close the Graph Properties window.

✦ You can also enter titles directly on the graph—click on the text and change it, then press the Enter key.

Dialog tab

- Click the Dialog tab on the left side of the graph. The tabs in the Dialog are for labeling the block and for setting what happens to the graph during or after the simulation run. In the Dialog:

  - In the block's Display tab, choose if you want the graph to not open, or to open at the beginning or the end of the simulation run. Also choose when you want the graph to autoscale.

  - At the bottom of the dialog, label the block **Chart.**

  - Close the block's dialog.

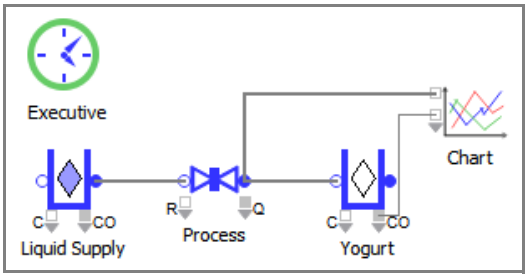
✦ See the Line Chart's Help button or the User Reference for more information about the tools and dialogs that control this block.

Save the model

- Save the model to save your changes

Your tutorial model

If you have followed all the steps, your model should look similar to the model shown below.



Running a simulation

Before running the model, decide if you are going to run a single model one or more times or run different models all at the same time.

Run modes

There are two modes toggled using the Run Mode toolbar button:

1) To run a single model as quickly as possible, even if you are running that model multiple times, use the fastest run mode as shown here. This is the default and what you will use for this tutorial.

2) To run different models all at the same time, use the multi-threaded run mode . This takes advantage of the ExtendSim multi-threading capability—the user interface is pro-

cessed in one thread and the processing for each model occurs in separate threads. In this mode, the Run Simulation button changes to .

Multi-core Runs

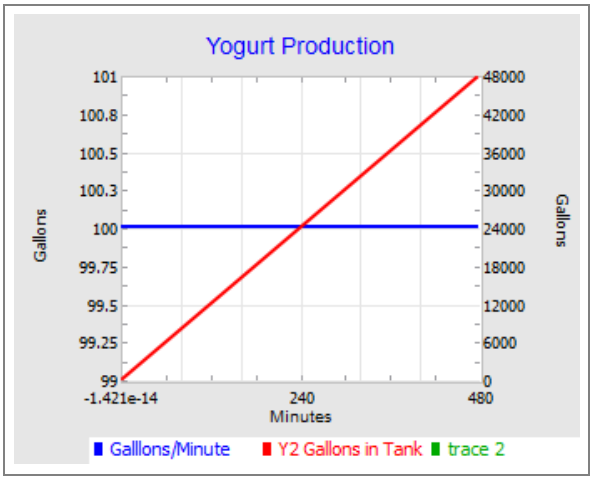
To run a single model multiple times by spawning multiple instances of ExtendSim on a computer's multiple cores, use the Multi-core Analysis Run button on the toolbar.



✦ See "Multicore Analysis" under Running a simulation in the How To chapter in the ExtendSim User Reference for details on how to set up and run models on multiple cores.

Run the simulation

- Click the Show 2D Animation button in the toolbar or choose the Run > Show 2D menu command to enable 2D animation.
- Click the Run Simulation button in the tool bar or choose the Run > Run Simulation command or right-click to run the simulation.
- If it isn't open, double-click the Chart block to see its graph. The block shows the results as a graph, seen at right, and as data in its Data tab.



As seen in the previous chapter, if you run with 2D Animation on, the blocks will automatically display information and status on their icons.

✦ These animations give a lot of useful information for discrete rate models. See the User Reference/Discrete Rate/Miscellaneous chapter for a description of the icon animations and what they mean. For instance, Tanks animate information about their levels under different capacity settings as well as the direction of flow within the Tank.

Verify and validate results

This is a good opportunity to verify and validate the results. Given how the model is constructed and the settings in its dialogs, are the numbers in the graph what you expected to see? Does the model accurately represent the system being modeled?

Note that there is never any change to the rate of flow, so there is no need for the model to recalculate the effective rate. This means that the simulation is finished in two events: the start event and the end event. The graph indicates that the effective rate is 100 and a total of 48,000 gallons of product (shown on the Y2 axis) have been produced in the process. This makes sense because the assumptions were that 100 gallons of yogurt would be produced per minute and the simulation time is 480 minutes.

✦ See the User Reference's Process of Simulation chapter for information about verifying and validating a model.

Save the model

- Save the model; you will use it in the next chapter.

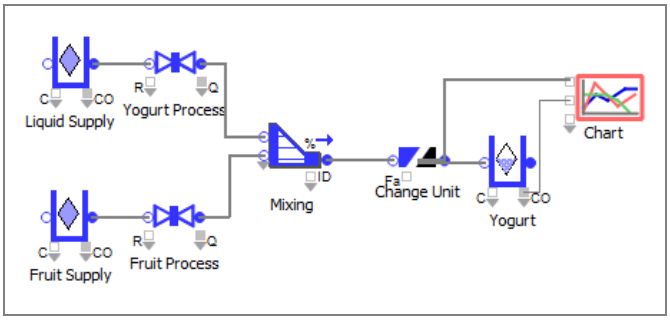
To compare your model to ours, see the example model named *Step 1 DR Tutorial* in the folder Documents/ExtendSim/Examples/Tutorials/Discrete Rate.

Discrete Rate Quick Start

Tutorial Part II

The goal

The previous chapter showed how to build a simple discrete rate model. This chapter adds some details to that model and discusses some ExtendSim features that can be used to enhance the model experience.



Model assumptions

The Yogurt Production model represents a process that takes a supply of liquids, converts it into plain yogurt, and mixes the yogurt with fruit. The fruited yogurt mixture is then poured into individual yogurt containers ready for shipment.

The assumptions for this model are:

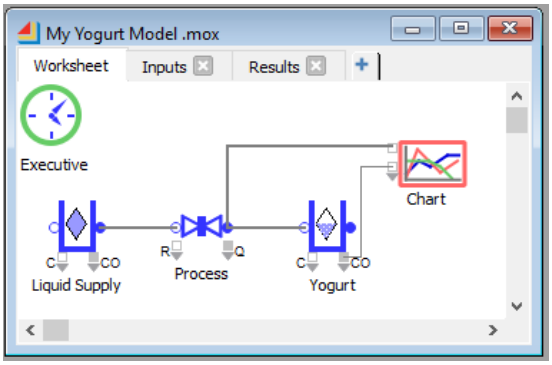
- The supply of liquid to make the yogurt comes from one location and the fruit mix comes from another location. The assumption is that there is no shortage of liquid or fruit, so the supplies are infinite.
- The liquid is processed into yogurt at a rate of 100 gallons/minute.
- The fruit is processed at a rate of 8 gallons/minute.
- The ingredients are mixed such that each 10 gallons of product is composed of 1 gallon of fruit and 9 gallons of plain yogurt.
- The Yogurt tank can store an unlimited amount of yogurt.
- The packaging process yields 12 containers of yogurt per gallon of mix.
- The model runs for a simulated time of 480 minutes (8 hours)

✦ The final Yogurt Production model and the example models that illustrate the steps described in this chapter, are located in the folder Documents/ExtendSim/Examples/Tutorials/Discrete Rate. However, to get the maximum benefit of this tutorial it is recommended that you build the models yourself then compare your models to the example models.

Steps you will take in this chapter

1) Start with an existing model
2) Add a second supply line for processing fruit and merge it with the yogurt line
3) Mix the yogurt and fruit supplies according to a recipe
4) Create a filling operation that puts the yogurt into containers
5) Optional: add user interface and other helpful features to your model

Start with the model from the previous chapter



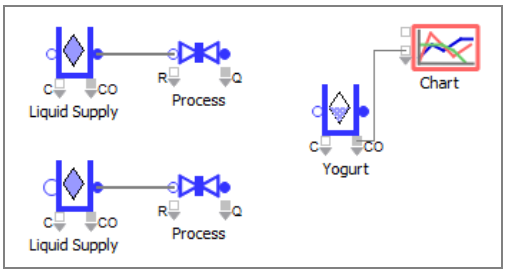
- Open the My Yogurt Model you built in the previous chapter
- Or, open the *Step 1 DR Tutorial* model from the Documents/ExtendSim/Examples/Tutorial/Discrete Rate folder

Add a second line for processing the fruit

So far the model has a single production line for processing the yogurt. The specification for the model indicates that fruit is mixed with the yogurt, requiring a second processing line.

Duplicate some blocks

- Select the connections from the Yogurt Process valve (one going to the Chart and the other going to the Yogurt tank) and delete them.
- Move the Yogurt tank and Chart blocks to the right.
- Frame-select the Liquid Supply tank and Yogurt Process valve so that both blocks and the connection between them are selected.
  - Right-click on the selection and give the Duplicate command. This places copies of the tank and valve below and to the right of the original blocks.
  - Move the two new blocks so they line up under the original blocks. as shown here.



Configure the new blocks

- In the new Liquid Supply tank:
  - Notice that, because this block is a duplicate, the *Initial contents* field in its Tank tab is already checked to be *infinite* and the *Flow group unit* in its Options tab is already set as *gallons*.
  - Change the block's label to **Fruit Supply**.
  - Close its dialog.

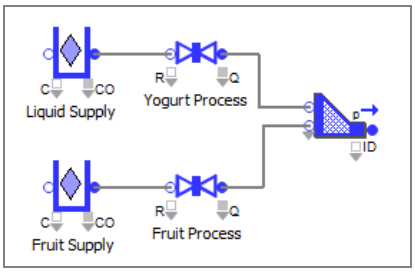
- In the new Yogurt Process valve:
  - In its Valve tab, change to **Maximum rate: 8 gallons/minute**.
  - Change the block's label to **Fruit Process**.
  - Close its dialog.

Mix the flows based on the recipe

The assumptions state that each 10 gallons of mix is composed of 9 gallons of plain yogurt and 1 gallon of fruit.

Add a Merge block

To merge the two flows according to the recipe:



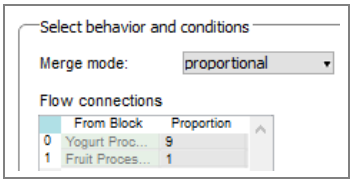
- Add a Merge block (Rate library) to the model and place it to the right and between the Yogurt Process and the Fruit Process blocks.

✦ You can add the Merge block from the library window or right-click the outflow connector of the Yogurt Process block and choose the Merge block.

  - If you added the Merge from the library window, connect from the outflow connector of the *Yogurt Process* valve to the top inflow connector on the Merge.

Connect the blocks

- Draw a connection line from the outflow connector of the Fruit Process valve to the second inflow connector on the Merge.
- Draw a connection line between the Merge block's outflow connector and the Yogurt tank's inflow connector.
- Draw a second connection line from the Merge block's outflow connector to the top input of the Chart.
- In the Merge block's Merge tab:

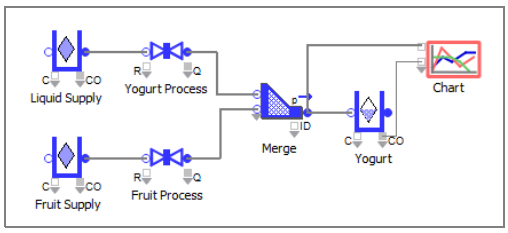


  - Select **Merge mode: proportional** from the popup menu.
  - Notice that the *From Block* column in the flow connections table shows the labels of the connected blocks.
  - In the Proportion column of the flow connections table, enter **9** for the Yogurt Process and **1** for the Fruit Process.
  - Label the Merge block **Mixing** and close its dialog.

Save and run the model

- Save, then run the model.

Your model should show that the effective rate is 80 gallons/minute and that 38,400 gallons have been produced.



This is an interesting model to run with 2D animation on since it helps with model verification. Notice that the rate displayed at the top of the Yogurt Process icon is displayed as the fraction 72/100. This is the ratio of the Valve's effective rate (72) to its maximum rate (100). Since the maximum output of the fruit process is only 8 gallons per minute, and the mixing process requires a ratio of 9 portions of plain yogurt to 1 portion of fruit, the amount of plain yogurt that can be taken is only 72 (8*9) gallons per minute. And with the 8 gallons of fruit per minute, the effective rate at the Mixing outflow is 80 gallons/minute.

✦ A complete description of the animation information shown on the Rate library block icons is given in the Discrete Rate Miscellaneous chapter of the User Reference.

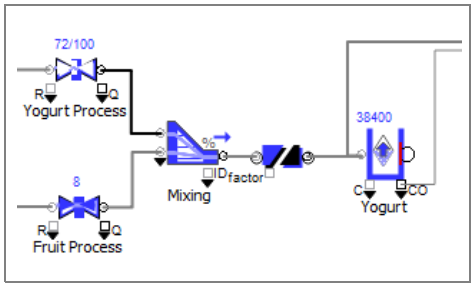
Your model should now be similar to the *Step 2 DR Tutorial* model located at Documents/ExtendSim/Examples/Tutorial/Discrete Rate.

Package in containers

The model assumptions state that the yogurt mix is packaged into containers at a ratio of 12 containers per gallon of liquid. To represent this, the units of flow need to be changed.

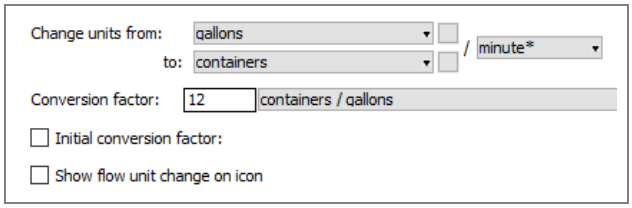
Insert a block to change the flow units

- Move the Yogurt and Chart blocks to the right to provide space for inserting a block.
- Right-click on the outflow of the Mixing/Merge block, select the Rate library, and choose a **Change Units** block. This Smart Block technique inserts the Change Units block on the connection line between the Mixing/Merge and the Yogurt tank.



The model should now look like the screenshot to the right.

- In the dialog of the Change Units block:
  - Keep the default setting for *Change units from: gallons.*
  - In the popup for the *second* unit setting (*to: gallons*), select **New Unit** and name the new flow unit **containers**.
  - So that each gallon will result in 12 containers, enter **Conversion factor: 12 containers/gallons**. (Be sure to select **containers/gallons** from the popup menu, not gallons/container.)



  - If you check the box for **Show flow unit change on icon** in the Change Units dialog, the area above the icon will display the text *gallons=>containers*.

- Label the block **Change Unit** and close its dialog.

At this point the dialog of the Change Units block should look like the screen shot shown above.

Change the connection line

Since you want to know the final effective rate:

- Delete the connection between the Mixing/Merge block and the Chart.
- Draw a new connection from the outflow of the Change Unit block to the top input of the Chart.

Update the Line Chart

The model is now reporting the rate in which gallons are packaged into containers as well as the number of containers the Yogurt tank holds.

- Right-click on the Line Chart's graph and choose the Trace Editor:
  - Change the name of the trace from Gallons/Minute to **Containers/Minute**
  - For the Y2 axis, change the trace name from Gallons in Tank to **Containers**
- Right-click on the Line Chart's graph and choose Graph Properties:
  - Since the units plotted on the Y2 axis have increased by a factor of 12, increase the number of places for the Y2 axis to **6**

Save the model

- Save the model to save your changes

Your model should now be similar to the *Step 3 DR Tutorial* model located at Documents/ExtendSim/Examples/Tutorial/Discrete Rate.

Model verification and validation

There is no randomness in this model, so when the model is run, the number of yogurt containers produced each minute, and thus the total number of containers processed, will be the same each time.

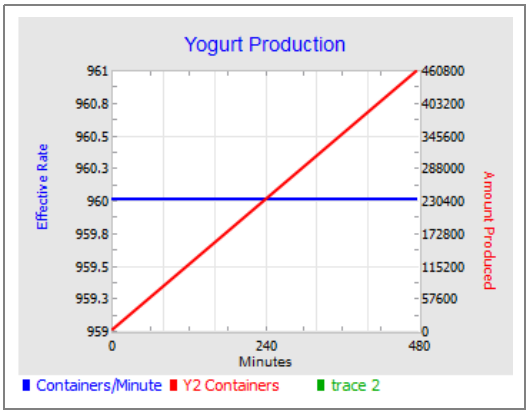


Chart block

The Graph and Data tabs of the Chart block should indicate that 460,800 containers of yogurt were produced. This makes sense because, as the Results tab in the Mixing/Merge block shows, the total flow was 38,400 gallons of fruited yogurt and each gallon results in 12 containers.

Model worksheet

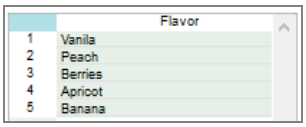
At the end of the simulation run, holding the cursor over each outflow connector will show the final rate at that connector. For example, the Yogurt Process has an effective rate of 72 gallons per minute, the Fruit Process will have a rate of 8 gallons per minute, the Mixing block will

indicate that 80 gallons of mix were produced that minute, and the Change Units block's output will show that 960 (80 * 12) containers of yogurt were packaged per minute.

The final Yogurt Production model

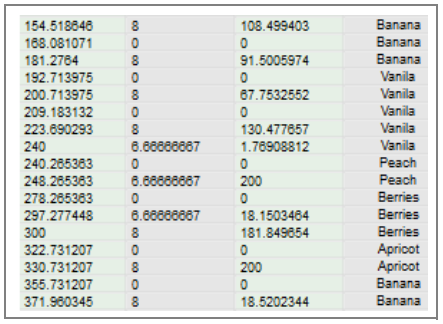
The Documents/ExtendSim/Examples/Tutorial/Discrete Rate folder contains a Y*ogurt Production Final* model has additional features and capabilities.

- Fruit supply:
  - Differentiates the fruit supply into five separate flavors using flow layer attributes.
  - Uses a Shutdown block (Item library) to represent a changeover period on the fruit supply process, so that the equipment can be emptied and cleaned between flavors.



	Flavor
1	Vanila
2	Peach
3	Berries
4	Apricot
5	Banana

- Liquid supply:
  - Adds a Lookup Table block (Value library) that places a dynamic constraint on the liquid supply's production rate, such that the rate is 60 gallons/minute rather than 100 for a period of time.
- Adds a Convey Flow block (Rate library) to simulate a cooling cycle. You could also connect a Shift block (Item library) to the Convey Flow block, stopping the cooling cycle process at the end of the day and causing the Convey Flow to be emptied of product.
- Packaging and palletizing:
  - Packages the containers into cartons.
  - Adds a palletizing area where cartons are stored for shipment, using the Rate library's Interchange block and discrete event blocks from the Item library.
  - Configures the pallets so that all the cartons on a pallet contain only one flavor of yogurt.
  - Includes the time it takes to unload a full pallet and load an empty pallet.
- Uses History(R) blocks from the Rate library to verify that the effective rate and the type of flavor produced are as expected.



154.518646	8	108.499403	Banana
168.081071	0	0	Banana
181.2764	8	91.5005974	Banana
192.713975	0	0	Vanila
200.713975	8	67.7532552	Vanila
209.183132	0	0	Vanila
223.690293	8	130.477657	Vanila
240	6.66666667	1.76908812	Vanila
240.265363	0	0	Peach
248.265363	6.66666667	200	Peach
278.265363	0	0	Berries
297.277448	6.66666667	18.1503464	Berries
300	8	181.849654	Berries
322.731207	0	0	Apricot
330.731207	8	200	Apricot
355.731207	0	0	Banana
371.960345	8	18.5202344	Banana

Your model could have other enhancements such as:

- If the model were more complex or required a lot of data, the ExtendSim internal database could be used to track inputs and outputs and make the model scalable.
- The Python Bridge block (Value library) can be used to integrate Python code into your model for advanced calculations, data processing, external analytics, or connectivity with other systems.
- The Scenario Manager (Value library) could evaluate and compare different model configurations.

- For additional model verification and validation, you could use the Reports Manager block (Report library) to get detailed information about every block and event in the model.

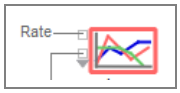
✦ See the User Reference for more details on how to implement these and other model enhancements.

Enhancing the user interface

The Yogurt Production model you built in these tutorials adheres to the assumptions and runs as expected. But it could use some enhancements to make it easier to use and more presentable.

✦ This section gives a brief overview of a few ExtendSim user interface features. See the How To: Big Picture chapter of the User Reference for a more extensive summary.

Named connections



The connection lines in this model are manageable, but larger models could have connection lines going all over the place. Named connections use text labels as outputs and inputs, causing data to jump from the output to the input without using connection lines. For example, the Step 1 DR Tutorial model uses named connections to report the amount of yogurt and fruit produced.

Clones and notebooks

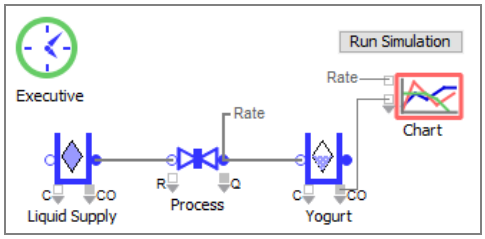
Clones are exact copies of the parameter fields, text, tables, or graphs from a block's dialog. Similar to a shortcut or alias, clones behave identically to the original. Clones can be placed on a model worksheet or on notebooks.

Notebooks are customizable windows for organizing and managing a model's information. You can clone dialog objects, insert text and graphics, and paste pictures onto a notebook, then use it to control model parameters, report simulation results, and document the model. And notebooks can be torn off from the model for better accessibility.

As seen on page 13, the Yogurt Production models have parameter fields, tables, and other objects cloned to two notebooks labeled Inputs and Results.

Interactive dashboard

ExtendSim has many tools and features for creating an interactive dashboard or control panel without programming. For example, the Buttons block (Utilities library) has several pre-built buttons that can be cloned onto the worksheet, such as the Run Simulation button shown here and in the Yogurt Production Final model.



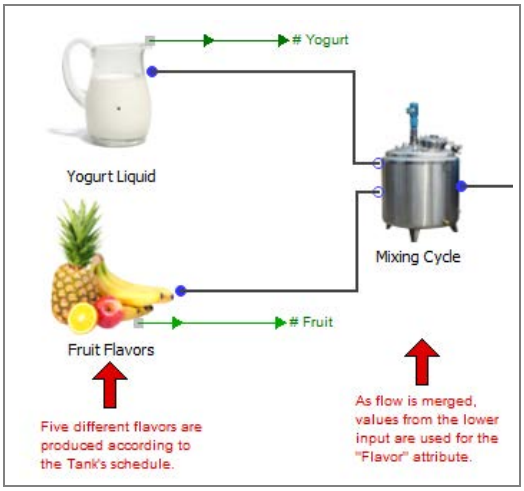
✦ For custom buttons, enter an equation in the Buttons block dialog and give your button a name.

Hierarchy

Hierarchy is a method for representing multiple blocks or an entire submodel as one block within the model. With hierarchy you can:

- Encapsulate sections of a model into one or more blocks for scalability
- Create generic model structures and store in libraries for reusability
- Customize and animate the icons that represent sections of your model

✦ Hierarchical blocks can be created from the top down or from the bottom up, can be copied within a model and from one model to another, and can be saved in libraries for reuse.



Custom animation

As discussed in the previous chapter, the Rate library has a great deal of 2D animated behavior already integrated in the blocks. This provides lot of useful information. For instance, Tanks show different colors and configurations to display information about their levels under different capacity settings as well as the direction of flow within them.

The main method for customizing animation for a Rate library block is by selecting in the block's dialog what you want shown on its icon when the simulation runs. For example, checking the box to show flow units on the icon of the Change Units block, as you did on page 38.

✦ See the User Reference/Discrete Rate/Miscellaneous chapter for a description of the icon animations and what they mean.

Next steps

When you've finished this chapter:

1) Please read the next chapter in this Quick Start Guide. It explains the discrete rate terms and concepts and provides a top level picture of the how the Rate library works.

2) As you build models, refer to the "Discrete Rate"' section of the User Reference and to the numerous example models located at Documents/ExtendSim/Examples/Discrete Rate. Supported by the example models, the Discrete Rate section of the User Reference is devoted to various aspects of building a discrete rate model so that you can to learn how to:

   - Create flow
   - Specify rates, constraints, and movement
   - Merge/diverge and route flow directly and remotely
   - Delay flow using goals, hysteresis, and the Convey Flow block
   - Mix flow and items
   - And get information about advanced topics such as LP technology, upstream supply and downstream demand, and messaging in discrete rate models

3) Also refer to the “How To” section of the User Reference, which points out ExtendSim features that will help you to:

   - Represent the dynamics of the system you’re modeling
   - Create a user interface for the model
   - Get data into the model
   - Manage data for use in the model
   - Perform analysis and assessments
   - Report model results and export data
   - Communicate with other applications and devices

Discrete Rate Quick Start

Concepts & Terminology

Global system

ExtendSim takes a very different approach to discrete rate models compared to continuous process or discrete event models. The Value library blocks in continuous models and the Item library blocks in discrete event models act individually and independently to calculate values or move items. They may send messages and communicate with each other, but there is no overall global connection between the blocks in those types of models.

By comparison, Rate library blocks are dependent on each other and have an effect on one another. Discrete rate models are divided into areas where the included blocks are not independent but instead are part of a global system. The blocks within each area communicate through an internal linear program (LP) that provides the global oversight for that area. Each block in an LP area contributes a part of the LP equation for the area; the result of the LP calculation are the effective rates for that part of the model. And this system is optimized such that, if a particular area does not need to recalculate, it won't.

Overview of discrete rate models

Discrete rate models are concerned with flows, constraints, rates, events, storage capacity, and routing.

- *Flow* is what is stored in and moves through a discrete rate system. Flow can be almost anything: liquids, electronic transactions, and cereal can all flow in a model. Flow is expressed and measured in flow units – either generic units or defined units such as packets, gallons, transactions, boxes, and so forth.
- Flow moves through flow connections in one direction, from one block's outflow connector to another block's inflow connector. It moves at a *rate* that is expressed as a quantity of flow per time unit – the number of packets per second, gallons per minute, boxes per hour, and so forth.
- The discrete rate architecture maximizes the movement of flow. Unless limited in some manner, flow would approach infinity and overwhelm the system. Because the discrete rate architecture maximizes the movement of flow, every model must contain one or more *constraints* (typically a Valve) to limit the rate of flow to something less than infinity. Some examples of constraints include the presence or absence of flow in a Tank, the maximum flow rate defined by a Valve block, and the rule chosen to distribute flow in a Diverge block.
- While constraints determine the maximum rate that flow can move, the effective rate is the actual rate of movement. The effective rate for each section of the model is determined using linear programming (LP), given the set of constraints that has been defined by the model's structure. The model's set of effective rates define how fast flow actually moves from one section of the model to another. As the simulation clock advances from one event to the next, the quantity of flow which has moved is updated.
- The state of a discrete rate model changes only when an *event* occurs. An event might be a Tank that becomes empty or full, a maximum rate that changes during the run, a block that changes its output

proportions, and so forth. Each time an event occurs, ExtendSim makes a calculation to determine, at that moment, what the effective rates are in each part of the model. Any portion of the model that can potentially be impacted by the new event has its effective rates recalculated. This takes into consideration the constraints put on the rates, the location of the flow, and storage *capacity* in the system.

- Each discrete rate model is conceptually divided into unit groups, rate sections, and LP areas. These divisions are handled automatically and internally, and are determined by the type of blocks used in a model, how the blocks are connected, the settings in the blocks, and so forth. See the Discrete Rate section of the User Reference for more information.

A connection between two rate-based blocks can thus be viewed as an infinitely short pipe that is always full of something at a constant pressure – as soon as the effective rate is more than 0, the pipe's contents move at the highest rate possible based on all the constraints given by the system. When the effective rate is 0, the pipe is still full but the flow instantly stops.

✦ While the ExtendSim discrete rate architecture preserves mass balance in the system, no explicit consideration is given to pressure, energy, momentum, or temperature, since these are beyond its scope.

⚠ The ExtendSim discrete rate system attempts to move flow through the model as fast as possible. Without any mechanism to impede its progress, the effective rate would theoretically approach infinity and the flow would move from one part of a model to another instantaneously. In order to avoid this error condition, each LP area of the model must contain one or more constraints (typically a Valve) to restrict the flow to a number that is below infinity. If the required minimum set of critical constraints is not present in a model, ExtendSim stops the simulation and displays an error message.

Terminology and architecture

When building a discrete rate model, it is helpful to understand the terminology and to have an overview of ExtendSim discrete rate architecture.

LP technology

To provide global oversight to calculate the effective rates in a discrete rate model, ExtendSim uses linear programming (LP) technology. The purpose of the LP calculation is to determine the maximum effective flow rates in the system given the constraints defined by block settings and the structure of the model. After all the rules for storage capacity and movement have been declared in the model, ExtendSim uses the LP calculation to cause as much flow as possible to move through the system. This calculation is handled automatically and internally. For more information, see the advanced topic "LP technology" in the Discrete Rate section of the User Reference.

Layout of a discrete rate model

A discrete rate model can combine continuous blocks (such as those in the Value library), and discrete event blocks (typically from the Item library), with discrete rate blocks from the Rate library. Other than the Executive block (discussed below), you can place the blocks in a model anywhere you want, remembering that ExtendSim evaluates discrete rate blocks along the path of the connections.

Executive block

If you use any discrete rate blocks in a model, the model will require the Executive block (Item library). The Executive does event scheduling and makes the LP calculation for rate-based models. It must be present in every discrete rate model and it must be located on the left of all the other blocks in the model.



The Executive plays a special role in discrete rate simulations. The block's Discrete Rate tab allows you to set global options for discrete rate models, manage quantity units, and select advanced options for specific Merge and Diverge modes. Its LP Solver tab has information about the linear program (LP) that provides global oversight for discrete rate models.

The settings in the Discrete Rate tab are explained fully in the User Reference.

✦ Most of the Executive's options are for advanced users. Unless you use flow layer attributes, it is rare that you would need to make any changes in the Executive's dialog.

Connectors and connections

The Rate library provides blocks for simulating rate-based flows. Most of the blocks in the Rate library have flow connectors and value connectors; the Interchange block also has item connectors:

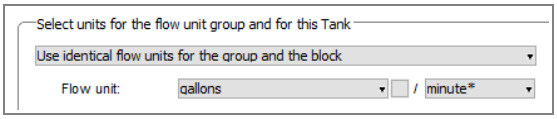
Connector type	Connectors (In & Out)	Line Type
Flow		
Value		
Item		

- In a discrete rate model, flow connectors report the effective rate of the flow at each event. The flow moves in one direction, from one block's flow output ("outflow") connector to another block's flow input ("inflow") connector.
- Value connectors provide information about the quantity of flow and a block's capacity, as well as information about the effects that the flow has in the model.
- The item connectors on the Interchange block provide an interface between portions of the model that are discrete rate and portions that are discrete event.

When combining discrete rate blocks with blocks from other libraries, you will only be able to connect compatible connectors. To represent the flow from one block to another, an outflow connector has to be connected to an inflow connector. Each flow connector can have one and only one flow connection. However, it is possible to connect an outflow connector to both an inflow and a value input connector. In this case the value connector reads the effective rate from the connection. A flow connection cannot be made with an item connector.

Units

- *Flow units* indicate what is flowing from one flow connector to another. For instance, gallons, bottles, and transactions are all types of flow units.



Each Rate library block in a discrete rate model has a flow unit. The flow units are identical for all the blocks within a unit group.

- *Time units* define how time is measured as the model run progresses. Like other ExtendSim blocks, blocks in the Rate library can use the default global time unit or a local time unit. Hours, minutes, and seconds are all examples of time units.


Calendar dates are not available if months or years have been selected as the specific global time unit for a discrete rate model. Furthermore, if Calendar dates has been selected, Rate library blocks will not be able to select Months or Years as their local time unit.

Rates

One of the most important aspects of a discrete rate model is the rate of flow. A rate is the ratio of the flow units to the model's time units. This is displayed in block dialogs as units/time, gallons/minute, transactions/second, boxes/hour, and so forth.

Several different types of rates are considered during the model building process:

- Maximum rate – the upper limit of the rate of flow.
- Effective rate – the actual rate of flow.
- Upstream supply and downstream demand – potential rates.
- Infinite rate – any value equal to or greater than a large specified number.

The User Reference and example models

The discrete rate portion of the ExtendSim User Reference has chapters that discuss specific discrete rate modeling concepts and techniques. It also has advanced topics such as LP technology, upstream supply and downstream demand, and messaging in discrete rate models.

ExtendSim ships with numerous example models such as the models used in the Quick Start tutorial and examples that show how to merge and diverge flow, set rates and constraints, and more.

Index

A-C

application areas
  discrete rate modeling 5
application window 11
arrays of connectors 17
asterisk (*) 25
attributes of flow 40
backup 25
block
  connecting 28
  connectors 16
  dialogs 15
  Help button 16
  icons 14
  inserting 38
  labels 14, 16
  names 14
  view 16
Bump Connect technique 18
buttons
  Run Simulation 12
Change Units block 38
changeover period 40
Chart library 20
charts 13
clones 41
Close All Library Windows command 18
communication between ExtendSim models and Python 20
connecting blocks 28
connection lines 17
  Smart Block technique 18
connections
  definition 16
  flow type 47
  lines 17
  methods 17
  named 18, 41
  throw/catch 18
connectors
  arrays of connectors 17
  definition 16
  Flow 47
  flow 16
  variable connectors 17
control key 27
Ctrl key 27

D-F

default view for block 16
definition of discrete rate modeling 3
dialogs
  opening 15
Discrete Rate chapters 9
discrete rate modeling
  application areas 5
  building a model 35
  compared to other methodologies 8
  declaring flow units 29
  definition 3
  event scheduling 47
  heterogeneous flows 5
  homogeneous flows 5
  rates (introduction) 48
  units 47
  why do it? 4
double-headed arrow
  for expanding a variable connector 17
dragging cursor
  definition 17
End time 24
example models 10
Executive block (Item library) 25
  in discrete rate model 47
flow
  attributes 40
  connectors 47
  definition 45
  layers 40
  maximized 45
  rate 45
  units 47
flow connectors 16
flow units 47
  declaring 29

G-I

Getting Started model 11
global object ID 15
global time units
  setting 24
graphs 13
Help button 16
heterogeneous things 5
hierarchy 21, 42

homogeneous stuff 5
How To chapters 9
industries 5
infinite check box in Tank block 29
inserting a block 38
interactive simulation 15

J-L

labels for blocks 14, 16
layers of flow 40
layout
  of a discrete rate model 46
libraries
  library windows 18
library
  Rate library 19
  Utilities 41
library window 18
Line Chart block (Chart library) 25
LP technology
  introduction 46

M-N

Merge block 37
model
  backup file 25
  block definition 14
  connection lines 17
  connectors 16
  example 10
  icons 14
  interactive 15
  layout of a discrete rate model 46
  opening 12
  running 12
models
  Quick Start 11
  Step 1 DR Tutorial 12
  Step 2 DR Tutorial 38
  Step 3 DR Tutorial 39
  Yogurt Production Final 40
multi-threaded run mode 31
multi-threading capability 31
named connections 18, 41
names of blocks 14, 16
notebooks 41

O-P

object ID 15
Open All Library Windows command 19
opening a model 12
plotters 13
Point and Click method 18
Point and Click technique 26
Python Bridge block 20

Q-S

Quick Start model 11
Rate library
  overview 19
rates
  flow 45
  in discrete rate models 48
Report library 20
results (displaying on charts) 13
run modes
  Fastest 31
  multi-threaded 31
  single-threaded 31
run parameters 24
Run Simulation button 12
Run Simulation command 12
running a model 12
Setup tab 24
simulation
  interactive 15
  parameters 24
  results on graphs 13
  running 12
  runs 24
Simulation Setup command 24
single-threaded run mode 31
Smart Block technique 18, 26
Smart Connection technique 28
Step 1 DR Tutorial model 12
Step 2 DR Tutorial model 38
Step 3 DR Tutorial model 39
stuff
  definition 5

T-V

Tank block 25
  intermediate storage 27
  sink 27
  source 27
Tank block (Rate library)
  infinite check box 29
things
  definition 5
Throw/Catch connection 18
time units
  global 24
tooltip 14

tutorial
simple model 23
User Reference
Discrete Rate chapters 9
How To chapters 9
Utilities library 20, 41
Valve block (Rate library) 25
variable connectors 17
views 16
volumetric flow rate 5

W-Z

ways to make connections 17
Yogurt Production Final model 40
Yogurt Production model 23