

# GoldSim Financial Module



GoldSim 15



Copyright GoldSim Technology Group LLC, 1998-2025. All rights reserved.  
GoldSim is a registered trademark of GoldSim Technology Group LLC.

GoldSim 15 (January 2025)

Document Revision: 0 (January 2, 2025)



GoldSim Technology Group LLC  
255 S. King Street, Suite 800  
Seattle, Washington 98104  
USA

Visit us at [www.goldsim.com](http://www.goldsim.com)  
Email us at [software@goldsim.com](mailto:software@goldsim.com)



---

# Contents

<b>Chapter 1: Introduction</b>	<b>1</b>
<b>Chapter Overview</b>	1
In This Chapter	1
<b>What is GoldSim?</b>	2
<b>What is the Financial Module?</b>	3
<b>Why Use GoldSim Instead of a Spreadsheet for Financial Calculations?</b>	3
<b>Who Should Use the Financial Module?</b>	4
<b>How This Manual is Organized</b>	5
<b>Learning to Use the Financial Module</b>	5
<b>Conventions Used in this Manual</b>	6
<b>Activating the Financial Module</b>	7
<b>The GoldSim Help System</b>	8
<b>Technical Support, User Resources and Software Upgrades</b>	8
GoldSim Maintenance Program	8
Getting Technical Support	9
Other GoldSim Resources	9
<b>Chapter 2: Getting Started with the Financial Module</b>	<b>11</b>
<b>Chapter Overview</b>	11
In This Chapter	11
<b>Basic GoldSim Concepts Necessary to Use the Financial Module</b>	12
Summary of Basic Concepts	12
Understanding Discrete Events and Triggering	13
<b>Overview of the GoldSim Financial Module</b>	19
The Financial Elements	19
The Built-in Financial Functions	27
Using GoldSim's Probabilistic Simulation Engine	31
Viewing and Analyzing Results	32
Documenting Your Financial Model	33
<b>A Simple Financial Module Example</b>	33
Step 1: Setting Up a Dynamic Financial Model	34
Step 2: Adding a Fund Element to Simulate a Bank Account	35
Step 3: Running the Model and Viewing a Simple Result	37
Step 4: Adding Discrete Deposits to the Account	38
Step 5: Running the Model with Deposits	40
Step 6: Removing Discrete Withdrawals from the Account	40
Step 7: Running the Model with Deposits and Withdrawals	43
Step 8: Running Multiple Realizations of a Financial Model	45
Where Do I Go From Here?	48
<b>Chapter 3: Details of the Financial Module Elements</b>	<b>49</b>
<b>Chapter Overview</b>	49
In This Chapter	49
<b>The Fund Element</b>	50

---

Defining How Interest is Computed for a Fund Element .....	51
Specifying Initial Balance, Deposits and Withdrawals for a Fund Element .....	52
Outputs of the Fund Element .....	54
<b>The Cash Flow Element .....</b>	<b>54</b>
Specifying Expenses and Revenues for a Cash Flow Element .....	55
Outputs of the Cash Flow Element .....	57
Computing the NPV of a Cash Flow History .....	57
Computing the IRR of a Cash Flow History .....	58
<b>The Investment Element .....</b>	<b>58</b>
Specifying the Growth of the Investment's Underlying Unit Value .....	59
Specifying Initial Holding, Purchases and Sales for an Investment Element .....	60
Output of the Investment Element .....	61
<b>The Option Element .....</b>	<b>61</b>
Specifying the Type of Option .....	62
Acquiring Options .....	63
Specifying the Growth of the Option's Underlying Unit Value .....	64
Valuing and Exercising Options .....	64
Outputs of the Option Element .....	66
<b>The Insurance Element .....</b>	<b>66</b>
Specifying the Properties of the Insurance Policy .....	67
Generating Insurance Claims .....	68
Outputs of the Insurance Element .....	69
<b>Managing Currencies and Exchange Rates .....</b>	<b>70</b>
Modeling Variable Exchange Rates .....	71
<b>Chapter 4: Running a Financial Simulation and Viewing Results .....</b>	<b>73</b>
<b>Chapter Overview .....</b>	<b>73</b>
In This Chapter .....	73
<b>Understanding Dynamic Financial Modeling in GoldSim .....</b>	<b>74</b>
Setting Up a Dynamic Financial Simulation .....	74
How Discrete Financial Events are Represented in Time .....	75
<b>Using Monte Carlo Simulation in Your Financial Model .....</b>	<b>76</b>
Setting the Monte Carlo Options for a Financial Model .....	76
<b>Saving and Viewing Financial Module Results .....</b>	<b>77</b>
<b>Chapter 5: Example Financial Module Applications .....</b>	<b>81</b>
<b>Chapter Overview .....</b>	<b>81</b>
In This Chapter .....	81
<b>Basic Financial Module Applications .....</b>	<b>82</b>
Example: Using the Fund Element .....	82
Example: Using the Cash Flow Element .....	84
Example: Using the Investment Element .....	86
Example: Using the Option Element .....	88
Example: Using the Insurance Element .....	90
Example: Simulating Variable Exchange Rates .....	91
<b>Advanced Financial Module Applications .....</b>	<b>92</b>
Example: Simulating a Loan Using a Fund Element .....	92
Example: Comparing Alternatives Using a Cash Flow Element - Repair vs. Replace .....	95
Example: Simulating a Stock Portfolio .....	97
Example: Simulating Long and Short Positions on a Security .....	99
Example: Simulating a Reinsurance Policy .....	101

---

**Glossary**

103

**Index**

105





---

# Chapter 1: Introduction

I guess I should warn you, if I turn out to be particularly clear, you've probably misunderstood what I've said.

Alan Greenspan, *Chairman, US Federal Reserve Board (1987-2006)*

## Chapter Overview

GoldSim is a user-friendly, highly-graphical, object-oriented program for carrying out dynamic, probabilistic simulations to support management and decision-making in engineering, science and business.

The GoldSim Financial Module is a program extension to GoldSim that allows you to probabilistically simulate financial systems that include components such as accounts and funds, investments, options, projects or undertakings with specified cash flows, and insurance policies. By combining the specialized financial elements in the Financial Module with GoldSim's underlying probabilistic, dynamic simulation framework, you can quickly simulate and analyze complex financial systems, as well as complex engineering and business systems that have a financial component.

Each of the GoldSim modules has a separate User's Guide describing its capabilities and features. This document provides a complete description of the features and use of the GoldSim Financial Module.

---

**Note:** This document only describes the Financial Module, a program extension to the GoldSim simulation framework. In order to take full advantage of all of the powerful features of the Financial Module, you will also need to become familiar with the various features and capabilities of the underlying GoldSim simulation framework. When necessary, these features and capabilities are mentioned and discussed briefly in the current document. They are described in detail in a separate document, the [GoldSim User's Guide](#).

---

## In This Chapter

This introductory chapter discusses the following topics:

- What is GoldSim?
- What is the Financial Module?
- Why Use GoldSim Instead of a Spreadsheet for Financial Calculations?
- Who Should Use the Financial Module?
- How This Manual is Organized
- Learning to Use the Financial Module
- Conventions Used in this Manual
- Activating the Financial Module

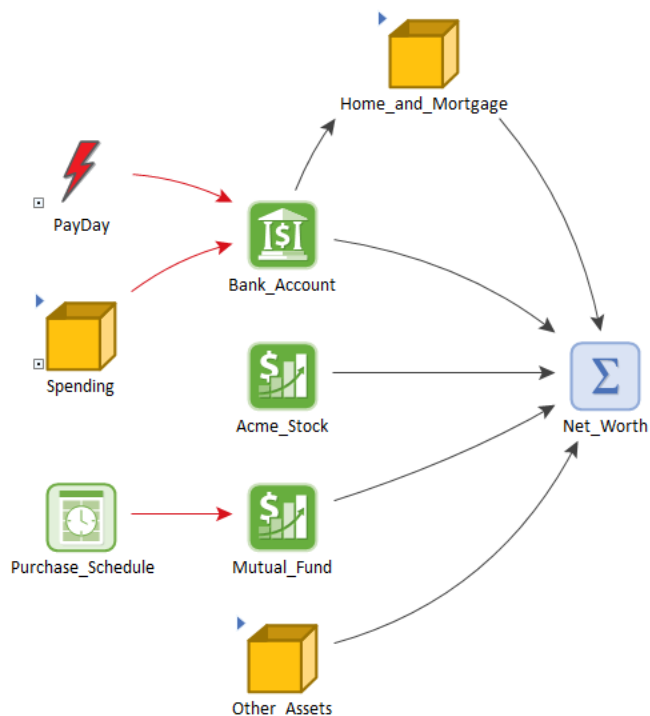
- Using Help
- Technical Support, User Resources and Software Upgrades

## What is GoldSim?

GoldSim is a computer program for carrying out dynamic, probabilistic simulations.

As used here, *simulation* is defined as the process of creating a model (i.e., an abstract representation or facsimile) of an existing or proposed *system* (e.g., a business, a project, a portfolio of investments) in order to identify and understand those factors which control the system and/or to predict (forecast) the future behavior of the system. Almost any system that can be quantitatively described using equations and/or rules can be simulated.

The GoldSim simulation environment is highly-graphical and completely object-oriented. That is, you create, document, and present models by creating and manipulating graphical objects representing the components of your system, data and relationships between the data:



In a sense, GoldSim is like a "visual spreadsheet" allowing you to visually create and manipulate data and equations. As can be seen in the example shown above, based on how the various objects in your model are related, GoldSim automatically indicates their influences and interdependencies by visually connecting them in an appropriate manner. GoldSim also sets up and solves the equations represented by the objects and their interdependencies.

The various objects with which a GoldSim model is constructed are referred to as *elements*. Each element represents a building block of the model, and has a particular symbol or graphical image (which you can subsequently customize) by which it is represented on the screen.

Although the standard elements incorporated within GoldSim can be used to build powerful and complex models, it was realized from the outset of the development of GoldSim that specialized elements and features may be required in order to efficiently model some kinds of systems. As a result, GoldSim was designed to readily facilitate the

incorporation of additional modules (program extensions) to enable it to address specialized problems. The Financial Module is one of these program extensions.

## What is the Financial Module?

The elements provided by the basic GoldSim framework can be (and have been) used to build complex financial models. However, a number of financial calculations are relatively complex, and building them using the standard GoldSim elements can be difficult, and in some cases, require development of custom programming (e.g., in a linked spreadsheet). For example:

- Computing the internal rate of return (IRR) of a series of cash flows requires an iterative calculation.
- Compounding interest in a fund (e.g., a bank account) can be relatively complex since the compounding can be based on different measures (e.g., minimum or average balance), and different compounding periods.
- When simulating investments (e.g., a portfolio of stocks), it is typically necessary to treat these as correlated "random walks" with specified drifts (trends) and volatility.

The Financial Module is a program extension to GoldSim that provides a number of specialized elements that allow these (and many other) kinds of financial calculations to be represented in a straightforward and easy manner. Moreover, the elements of the Financial Module work seamlessly with the rest of the elements in the GoldSim simulation framework, allowing you to access all the power and flexibility of GoldSim.

## Why Use GoldSim Instead of a Spreadsheet for Financial Calculations?

Most financial models today are built using spreadsheets. Spreadsheets are relatively inexpensive and most analysts are comfortable with using them. In addition, numerous macros and add-ons are available to enhance spreadsheets (e.g., tools to enable Monte Carlo simulation within a spreadsheet). So it is natural to ask: "Why can't I just use a spreadsheet for my financial models? What are the advantages of using GoldSim?"

Although spreadsheets are very useful for accounting tasks and managing lists of items, they have weaknesses when used for quantitative modeling tasks, such as analyzing and forecasting the performance of a business or investment portfolio. The major weaknesses of spreadsheets for these types of applications, and how GoldSim addresses these issues, are outlined briefly below:

- **Complex spreadsheets are generally not transparent and are error-prone.** Because of the row and column paradigm used by spreadsheets, the fact that equations are written in terms of cell references, the invisibility of the dependencies between cells, and the lack of a graphical means to document spreadsheets, most spreadsheet models have a low level of transparency. Not only does this lack of transparency make it difficult for others to understand a model, it makes it difficult to check for errors. As a result, numerous studies have shown that complex spreadsheets have a very high incidence of errors. GoldSim's interface, which allows you to build hierarchical, graphical representations of your system (in terms of influence diagrams) was specifically designed to facilitate the construction of transparent, well-documented models.

- **Spreadsheets cannot represent complex dynamics.** In a spreadsheet, you typically deal with dynamics by adding a row (or column) for each timestep (i.e., each day, each quarter, each month) that you want to forecast a value for. In addition to being a very awkward way to represent dynamics, this has a number of serious disadvantages: 1) it is difficult to represent dynamic feedback loops and delays, where a change made to one part of the system has a delayed impact; 2) sudden events (e.g., a deposit or withdrawal, an interest rate change) are difficult to accurately represent; 3) changes in the system's structure with time are hard to represent (e.g., taking out a loan when required), and 4) the length of the timestep cannot be dynamically adjusted during a simulation (e.g., in response to changing conditions). GoldSim is a dynamic simulation program, and representing such complex dynamics is straightforward and easy.
- **Spreadsheets cannot represent uncertain and/or stochastic systems.** All real-world financial systems have uncertain and stochastic components. Spreadsheets cannot deal with these directly. There are third-party add-ons that enable Monte Carlo simulation in a spreadsheet. However, because GoldSim was specifically designed to simulate such systems, it is able to more accurately represent and model stochastic processes and random events than these third-party add-ons.
- **Spreadsheets have no ability to handle dimensions and units.** Because spreadsheets deal only in numbers, and cannot represent units, great care must be taken when building models to handle unit conversions. GoldSim understands dimensions and units, carries out automatic unit conversion, and prevents you from constructing dimensionally-inconsistent models.
- **Some financial components require relatively complex programming in a spreadsheet.** For example, when simulating investments (e.g., a portfolio of stocks), it is typically necessary to treat these as correlated "random walks" with specified drifts (trends) and volatility. Programming this in a spreadsheet can be complex. The GoldSim Financial Module provides a number of specialized elements that are designed to represent common financial components (e.g., funds, investment portfolios, insurance policies, cash flows).
- **GoldSim is a powerful and flexible dynamic simulator that can represent processes (e.g., physical processes, business processes) that cannot accurately be represented in a spreadsheet.** By combining the basic GoldSim framework with the Financial Module, you can simultaneously model both the complex processes associated with a system (e.g., the movement of material through a mine) and the financial components of the system (e.g., the costs associated with transporting the material).

These and other advantages make GoldSim a powerful and flexible improvement over spreadsheets for building quantitative financial models. In addition, because GoldSim can link seamlessly to spreadsheets, you can still use spreadsheets as databases for inputs and outputs, and even as subroutines for specialized calculations.

## Who Should Use the Financial Module?

The GoldSim Financial Module is intended for use by analysts, researchers and students who are interested in understanding and predicting the performance of complex financial systems, and the interaction of such systems with the outside world.

The software itself, although relatively complex, can be mastered by anyone familiar with the basic functions of a personal computer and the Windows operating system. The key requirements for applying the Financial Module are a clear understanding of the financial system being modeled; and a basic understanding of uncertainty analysis and probability theory:

- Because the software was designed to be extremely flexible, it intentionally imposes few constraints on the inputs that you define. Hence, it is your responsibility to ensure that the system being defined is consistent and realistic. As a result, the most important requirement is that you have a clear understanding of the features, processes, and events controlling the behavior of the system to be modeled. This should include a good understanding of the fundamentals of financial analysis and modeling.

- Although GoldSim can be run in a deterministic manner (i.e., with no specified uncertainty in input parameters), one of the key features of GoldSim is its ability to explicitly represent such uncertainty through the use of probability distributions. In order to do so, the user must have at least a basic understanding of the representation and propagation of uncertainty. Appendix A of the [GoldSim User's Guide](#) provides a brief primer on this topic, along with suggestions for further reading.

## How This Manual is Organized

The Financial Module is a specialized extension to the basic GoldSim simulation framework. The document you are reading only describes the GoldSim Financial Module, and assumes that you are somewhat familiar with the basic capabilities of GoldSim. The basic capabilities of GoldSim are described in the [GoldSim User's Guide](#). That document provides a complete description of the features and capabilities of the GoldSim simulation framework.

This document is organized into five chapters.

The five chapters are as follows:

- **Chapter 1: Introduction.** The remainder of this chapter provides the information required for you to get started using the GoldSim Financial Module, including conventions used in the manual, activating the module, using online help, and obtaining technical support.
- **Chapter 2: Getting Started with the Financial Module.** This chapter provides an overview of the features and capabilities of the Financial Module, provides a brief overview of some key concepts of the GoldSim simulation framework, and walks you through a simple example in order to help you get started with the Financial Module.
- **Chapter 3: Details of the Financial Module Elements.** This chapter describes the specialized Financial Module elements in detail.
- **Chapter 4: Running a Financial Simulation and Viewing Results.** This chapter discusses how to set up and run a financial simulation and view results of the simulation.
- **Chapter 5: Example Financial Module Applications.** This chapter describes a number of examples which illustrate the application of the various features and capabilities of the Financial Module.

## Learning to Use the Financial Module

Although GoldSim's intuitive interface will tempt you to simply dive in and start playing with the software, you are strongly discouraged from doing so, even if you are an experienced modeler. Spending some time up front (by following the steps outlined below) is the quickest and most effective way to understand the software's features and capabilities and start building models using the GoldSim Financial Module.

1. **Learn how to use the basic GoldSim framework first.** In order to use the Financial Module, you must first have a basic understanding of the GoldSim framework. You cannot learn the extension without first learning the basic concepts underlying framework. At a minimum, you should take the GoldSim Tutorial. The Tutorial can be accessed from the main GoldSim menu (**Help | Tutorial...**). It is also available directly [here](#).

2. **Take the Online Training Course.** Although the Tutorial is of value to quickly get an overview of GoldSim, it provides a very simple introduction to the most basic GoldSim concepts. The best and most effective way to learn GoldSim is to take a full training course ([Introduction to GoldSim](#)). This free, "hands-on" self-paced course will provide you with a thorough understanding of the key concepts on which GoldSim is based and all of the fundamentals required to build complex models of nearly any kind of system. (Note, however, that the Course only discusses basic GoldSim and does not discuss the Financial Module). Because the Course is quite thorough, it will likely take as long as 40 hours to complete. Of course, if you are already somewhat familiar with simulation (and/or have a strong quantitative background), you may in fact be able to cover the material in considerably less than 40 hours. Note that the Online Training Course always describes that latest version of GoldSim (which will typically be slightly different than older versions).
3. **Read "Getting Started with the Financial Module" (Chapter 2).** This provides an introduction to and a "quick tour" of the GoldSim Financial Module. It presents the basic concepts of how components of a financial system can be simulated in GoldSim, provides an overview of the features and capabilities of the program, and summarizes the specialized elements associated with the module.

**Read More:** *Chapter 2: Getting Started with the Financial Module* on page 11

4. **Open and explore the example files.** When you install GoldSim, a folder labeled "Financial Examples" is installed with the program. (You can quickly access these files by selecting **File|Open Example...** from the main GoldSim menu). This directory contains example Financial Module model files. These examples are introduced and discussed in Chapter 5. These example model files are an excellent way to begin to experiment with the Financial Module.

**Read More:** *Chapter 5: Example Financial Module Applications* on page 81

5. **Download Example Files from the Model Library.** The GoldSim website contains a [Model Library](#) with a number of models illustrating how GoldSim can be used for particular applications. These models tend to be more complex than the simple example files found in the Financial Examples folder, but still relatively simple. Again, while exploring the files, use GoldSim's context-sensitive Help (i.e., the **Help** button in each dialog) to learn more about particular elements or features utilized in the model.
6. **Browse the User's Guides or Help System.** GoldSim has a large number of features, and you will not discover all of them by experimenting with simple example models. To fully utilize GoldSim's powerful features, browse through the User's Guides (both for GoldSim and the Financial Module), using the index and table of contents as your guide. The User's Guides are heavily cross-referenced, so it is easy to just jump around. Note that the Help system contains all of the contents of the User's Guides (with the exception of the technical appendices).
7. **Request your free one hour web-based training session.** When you purchase GoldSim, you are entitled to a free one hour, live web-based training session in which one of our analysts provides an interactive training session via the Internet and telephone. You are strongly encouraged to take advantage of this free training, during which our analysts can provide an introduction to both the basic GoldSim framework and the Financial Module.
8. **Contact us with questions.** When you purchase GoldSim, you are entitled to one year of free support. This does not include assistance in building and debugging your models, but it does include answering questions on how to use GoldSim's features, so feel free to contact us! The best way to do so is through the [GoldSim Help Center](#).

## Conventions Used in this Manual

The following conventions are used in this manual:

Convention	Description
Important Terms	New and important terms are presented in bold italics. These terms all appear in the Glossary of Terms at the end of the document.
File   Open...	Menus and menu selections are separated by a vertical bar. For example, File   Open... means "Access the File menu and select Open".
Ctrl+C	Key combinations are shown using a "+" sign. For example, Ctrl+C means press the Control and C keys simultaneously.
Warning	This means watch out! Warnings typically alert you to potential pitfalls and problems that may occur if you perform (or fail to perform) a certain action.
Note	Notes highlight important information about a particular concept, topic or procedure, such as limitations on how a particular feature can be used, or alternative ways of carrying out an action.

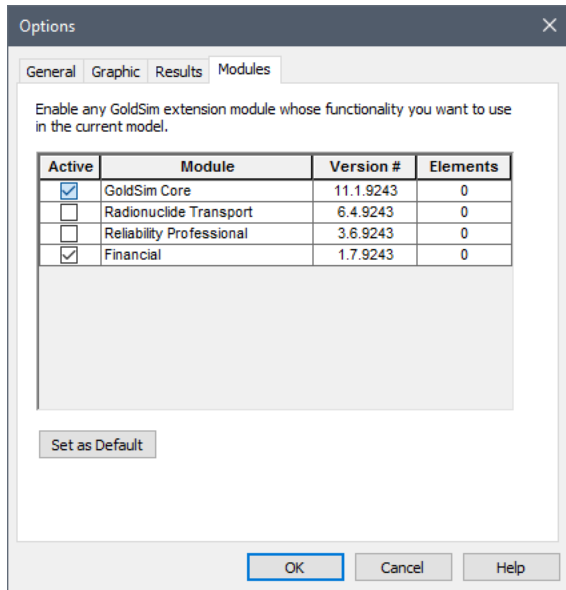
In describing the various mouse actions, the following conventions are used:

Mouse Action	Definition
Click	Press and release the left mouse button once.
Double-click	Press and release the left mouse button twice in rapid succession.
Right-click	Press and release the right mouse button once.
Drag	Press the left mouse button, and while keeping it depressed, move the cursor to another location, then release the button.

## Activating the Financial Module

In order to access the features and capabilities of the Financial (FN) Module, it must be activated on your machine.

You can determine if the Financial Module is activated on your machine by selecting **Model| Options...** from the main menu and selecting the **Modules** tab. The following dialog will be displayed:



All extension modules that you are licensed to use appear in the dialog.

The Financial Module is included with all GoldSim licenses, so it will always be listed. You can activate and deactivate modules that you are licensed to use by clicking the **Active** checkbox. By default, whenever you activate GoldSim, none of the available extension modules allowed by your license will be activated. To use them, you must activate them using this dialog.

If you deactivate a module (such as the Financial Module), any specialized elements associated with that module will be deleted (if any are present) and any menu options will be removed in the current file. If you make a module active, the various options associated with that module are made available again. If you select **Set as Default**, the selected modules will be activated for all new models that are created.

## The GoldSim Help System

GoldSim has an extensive in-product Help facility. The Help system can be accessed by selecting **Help | Help Topics** from the main GoldSim menu or the **Help** button (the question mark) on the standard toolbar. Pressing the **Help** button within any of the dialogs also provides access to Help (in a context-sensitive manner). All of the information in the User's Guides is accessible via GoldSim Help (with the exception of any Technical Appendices).

## Technical Support, User Resources and Software Upgrades

The GoldSim Technology Group is dedicated to providing complete solutions for our customers. We pride ourselves in providing prompt and extensive support and resources to our users, and are committed to ensuring that each installation of our software is successful and adds value to the customer.

## GoldSim Maintenance Program

When you purchase GoldSim software, you receive one year of [Software Maintenance](#), entitling you to the following:



- Free software upgrades so that you always have the latest version of the GoldSim software.
- Basic Technical Support via email (and if necessary, web meeting). Basic support covers installation and licensing questions, as well as questions about GoldSim's features and capabilities.

After the first year, if you wish to continue to have access to new versions and technical support, Software Maintenance can be extended each year with payment of an annual fee.

---

**Note:** Software maintenance is automatically included with GoldSim Leases.

---

## Getting Technical Support

Users with active Software Maintenance can submit questions directly to the GoldSim support team. Evaluation users are also welcome to contact us with questions on GoldSim functionality. The [GoldSim Help Center](#) is the primary portal for technical support. You can submit your questions directly from the Help Center. If you register and log in through the Help Center, you will be able to check the status and view a history of all of your support requests.

The Help Center also includes:

- The [GoldSim Forum](#), where you can post questions to the GoldSim community, or just browse existing messages;
- Articles on licensing questions and modeling tips; and
- An archive of past webinars (which demonstrate GoldSim features and capabilities).

Free Basic Technical Support does not include consulting, model trouble-shooting or detailed assistance with applying GoldSim to a particular problem. Assistance of this nature is defined as [Advanced Technical Support](#). Users may purchase Advanced Technical Support in pre-paid 10 hour blocks.

## Other GoldSim Resources

In addition to the GoldSim Help Center, additional resources are also available. These three resources can be accessed directly from the GoldSim website ([www.goldsim.com](http://www.goldsim.com)):

- Free [Online Training Courses](#) that provide you with a thorough understanding of the key concepts on which GoldSim is based and all of the fundamentals required to build complex models of nearly any kind of system.
- The [GoldSim Model Library](#), which contains a collection of example models to allow you to see how specific features of GoldSim can be used and/or how GoldSim can be used for specific applications.
- The [GoldSim Blog](#), which provides an informal mechanism for GoldSim staff to share their knowledge, point out some of the more advanced (and perhaps overlooked) GoldSim features, share and discuss common mistakes we see in GoldSim applications, discuss interesting applications, and keep you abreast of our plans for further GoldSim developments.

You can stay up to date on the latest GoldSim news by joining the [GoldSim LinkedIn Group](#), which is primarily used for announcements (e.g., new versions, interesting applications).

---

**Note:** When you purchase GoldSim, you are entitled to a free one hour, live web-based training session in which one of our analysts provides an interactive training session via the Internet and telephone. You are strongly encouraged to take advantage of this free training.

---

---

# Chapter 2: Getting Started with the Financial Module

Finance is the art of passing money from hand to hand until it finally disappears.

Robert W. Sarnoff, *Chairman, RCA (1970-1975)*

## Chapter Overview

This chapter provides a brief introduction to some key concepts of the GoldSim simulation framework, provides an overview of the features and capabilities of the Financial Module, and walks you through a simple example in order to help you get started.

If you read nothing else before starting to use the Financial Module, it is strongly recommended that you read this chapter, as it will tell you what the program is capable of doing, provide an overview of how to build a model, and direct you to portions of the manual where you can obtain further information.

---

**Note:** While this chapter provides an introduction to the features of the Financial Module, the technical details are spelled out in Chapters 3 through 5.

---

## In This Chapter

This chapter discusses the following topics:

- Basic GoldSim Concepts Necessary to Use the Financial Module
- Overview of the GoldSim Financial Module
- A Simple Financial Module Example

## Basic GoldSim Concepts Necessary to Use the Financial Module

The Financial Module is a program extension to the GoldSim simulation framework. In order to take full advantage of all of the powerful features of the Financial Module, you will eventually need to become familiar with many of the features and capabilities of the underlying GoldSim simulation framework.

At a minimum, in order to get started with the Financial Module, you must have an understanding of some fundamental concepts regarding GoldSim. These concepts are provided in the GoldSim Tutorial, which can be accessed from the opening splash screen of GoldSim, or by pressing **Help | Tutorial...** from the main menu. The Tutorial takes approximately one to two hours to complete (and can be completed in phases). If you are new to GoldSim and have not yet completed the Tutorial, you should do so now before proceeding.

### Summary of Basic Concepts

The basic GoldSim concepts that should be understood before you get started with the Financial Module are summarized below. These basic concepts are all discussed in the GoldSim Tutorial, and fall into the following categories:

- Basic Simulation Concepts
- Basic GoldSim Concepts
- Introduction to the GoldSim User Interface

Throughout the remainder of this document, whenever necessary, these features and capabilities are mentioned and discussed briefly. Cross-references are provided to the more detailed descriptions available in the [GoldSim User's Guide](#).

### **Basic Simulation Concepts**

A summary of basic simulation concepts discussed in the GoldSim Tutorial is provided below:

- Simulation is the process of creating a model of an existing or proposed system in order to identify and understand the factors that control the system, or to predict the future behavior of the system.
- In a static simulation, the system model does not change with time.
- In a dynamic simulation, the system model changes and evolves with time.
- Deterministic simulation often represents "the best guess" or "worst case" input values.
- Probabilistic simulation represents uncertainty by specifying some inputs as probability distributions.
- Simulation is a powerful and important tool because it provides a way in which alternative designs, plans and/or policies can be evaluated without having to experiment on a real system.

### **Basic GoldSim Concepts**

A summary of basic GoldSim concepts discussed in the GoldSim Tutorial is provided below:

- GoldSim represents parameters, processes, or events in a system using objects called elements.
- Each element has a symbol or graphical image to represent it, and has a dialog where you specify its properties.
- An element accepts input data and produces output data.

- There are six primary categories of elements: Inputs, Functions, Events, Stocks, Delays, and Results.
- GoldSim represents the links (dependencies) between elements using arrows called influences.
- A special type of element, the Container, can be used to hierarchically organize other elements.
- GoldSim ensures dimensional consistency and carries out all unit conversions for you.

These topics are discussed in detail in Chapter 3 of the [GoldSim User's Guide](#).

## **Introduction to the GoldSim User Interface**

A summary of basic GoldSim user interface concepts discussed in the GoldSim Tutorial is provided below:

- GoldSim has two sections to its window: the *graphics pane* and the *browser*.
- By default, the browser is displayed on the left side of the GoldSim window. You can open and close the browser using the browser button on the toolbar:



- GoldSim commands can be accessed using the menu, toolbars, or context menus.
- Elements are added to the graphics pane by right-clicking on an empty section of the graphics pane, and selecting the appropriate element from the context menu.
- Element inputs, outputs and settings are specified in the element's Properties dialog, which can be displayed by double-clicking on the element's icon.
- The Simulation Settings dialog can be displayed by selecting **Run|Simulation Settings** from the main menu, by pressing **F2**, or by pressing the Simulation Settings button in the toolbar:



- The Simulation Settings dialog allows you to adjust the length of the simulation, the timestep length, and the number of Monte Carlo realizations.
- A model is run by selecting **Run|Run Model** from the main menu, by pressing **F5**, or by pressing the Run button in the toolbar:



- After a model is run, it is in Result Mode. You cannot change the structure of the model or any of the input values while in Result Mode.
- Results can be viewed by right-clicking on an element, or double-clicking on a Result element.
- You can return to Edit Mode from Result Mode by selecting **Run|Return to Edit Mode**, pressing **F4**, pressing the Edit Mode button in the toolbar:



These topics are discussed in detail in Chapter 3 of the [GoldSim User's Guide](#).

## **Understanding Discrete Events and Triggering**

Use of the Financial Module requires a good understanding of one set of advanced features in GoldSim: discrete event modeling. Discrete event modeling is discussed briefly in the Tutorial, but due to its importance within the context of the Financial Module, it is discussed further in the sections below.

The basic description of discrete event modeling and triggering presented here should be sufficient to allow you to understand the features and capabilities of the financial elements discussed in subsequent sections.

However, to fully utilize all the features of GoldSim and the Financial Module, you will eventually want to learn more about the details of discrete event modeling.

Discrete event simulation is discussed in detail in Chapter 5 of the [GoldSim User's Guide](#).

### **What is an Event?**

In GoldSim, a discrete event is something that occurs instantaneously (as opposed to continuously or gradually) in time. It represents a "spike", a discontinuity, a command, or a discrete change of state for the system.

For example, through continuous compounding of interest, the money in a bank account continuously increases, but the account can also increase and decrease instantaneously due to discrete events (i.e., deposits and withdrawals).

Such events are particularly important for the Financial Module because the systems being modeled are controlled primarily by discrete occurrences (such as deposits, withdrawals, purchase, expenses, and the exercising of options).

Of course, "instantaneous" and "gradual" are relative terms. That is, whether something is treated as instantaneous or gradual is a function of the time scale of interest, and hence you must differentiate between the two based on the context of your model. Typically, the distinction will be obvious. For example, if the time scale of interest is 10 years, something happening over the span of a day can be considered to be "instantaneous". If the time scale of interest is several days, however, something happening over the span of a day would in most cases need to be treated in a continuous manner.

GoldSim handles "instantaneous" changes to a model by providing a mechanism for a model to generate and respond to discrete events. This is accomplished by providing the ability to instantaneously trigger an element to take a particular action (e.g., instantaneously change its value) in response to an event. Hence, in GoldSim, an event is specifically defined as *an instantaneous occurrence that subsequently triggers a particular action*.

In GoldSim, an event can be generated in one of four ways:

- The event occurs when a specified condition (e.g.,  $X > Y$ ) becomes true or false;
- The event occurs when a specified output in the model changes;
- The event occurs at a specified calendar or elapsed time; or
- The event occurs based on a specified rate of occurrence, which can be treated as regular or random ("occur exactly once a week" or "occur, *on average*, once a week").

In addition, some elements can respond to an event generated via one of the mechanisms above, and generate a new event.

In some cases, an event will occur (e.g.,  $X$  becoming greater than  $Y$ ) which triggers a particular action in a single element (exercise an option). In such a case, the event is internal to that element, and it does not directly impact other elements. In other cases, however, an event may impact multiple elements, or one element may respond to an event by triggering other elements to take a particular action. In these cases, it is necessary for discrete signals to propagate between elements.

In order to propagate events (and their consequences) between elements in a model, it is necessary to send information between elements intermittently as a "spike" or discrete "packet" of information. To facilitate this, GoldSim allows certain elements to emit and receive (i.e., produce as outputs and/or accept as inputs) a **discrete signal**. Discrete signals are a special category of outputs that emit information discretely, rather than continuously.

Within GoldSim, there are actually two types of discrete signals that can be passed from one element to another: *discrete event signals* and *discrete change signals*.

A discrete event signal is a discrete signal indicating that something (e.g., a purchase or a sale) has occurred. It does not describe the consequence of that occurrence; it simply emits a signal between elements indicating that an event has occurred.

A discrete change signal, on the other hand, emits information regarding the response to an event. In particular, a discrete change signal contains two pieces of information: a value (e.g., 10 dollars) and an instruction (e.g., Add). A discrete change signal can only be generated in response to an event. It can only be received by a few select elements (e.g., a Reservoir, some Financial Module elements) which understand how to process it.

Within the Financial Module, discrete event modeling is a fundamental feature of the elements used to model financial components, and is utilized in the following ways:



- Several elements (which are used to model system components such as funds, accounts, investments, and cash flows) accept as inputs discrete change signals in the form of discrete additions and withdrawals (e.g., deposits, withdrawals, purchases, expenses, revenues).
- Two elements require triggers (by an event) in order to carry out a particular action (i.e., acquire or exercise an option, process an insurance claim).
- Several elements output discrete change signals in response to events (e.g., a discrete output representing the return when an option is exercised). These outputs can then be subsequently used by other elements.







**Note:** Depending on how the event was generated, it may not fall exactly on a "scheduled update" (i.e., a timestep that was defined in the **Time** tab of the Simulation Settings dialog). That is, an event could actually occur between scheduled updates of the model. Such events can trigger an "unscheduled update" of the model.

### **Basic GoldSim Elements that Support Modeling of Events**

In addition to elements within the Financial Module, GoldSim provides a wide variety of other elements that can be triggered by events or can output discrete event signals. These elements are important, as they can be used in conjunction with the financial elements to represent complex financial models. For example, the outputs of these elements can trigger actions within financial elements (e.g., exercising an option).

Some of the event elements that are likely to be most useful to financial modelers are summarized in the following table:

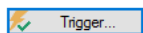
Element	Default Symbol	Function
Timed Event		Generates events based on a specified rate of occurrence, regularly or according to a specified distribution (i.e., randomly).
Triggered Event		Generates events based on one or more specified conditions.

Element	Default Symbol	Function
Discrete Change		Generates a discrete change signal (a value) that can subsequently discretely modify the values of other elements (e.g., add a deposit to an account).
Decision		Generates one of up to three alternative events based on specified conditions.
Random Choice		Generates a user defined event based on specified probabilities.
Milestone		Records the time at which a particular event or specified condition(s) occurs.
Status		Outputs a condition (True/False) in response to particular events or specified conditions.
Event Delay		Delays an event for a specified time. Can also be used to simulate queues.

### Defining Triggers

All discrete event modeling involves *triggering* of one form or another. When you trigger an element, you are telling it that a discrete event has occurred that you want the element to respond to.

Various elements respond to a trigger in different ways. They are, however, all triggered in the same way. That is, these elements all share a common Triggering dialog, which controls how they are triggered. All Triggering dialogs are accessed via a **Trigger...** button that will look similar to this:




---

**Note:** Depending on the element, the label for the trigger button will not always be "Trigger...".

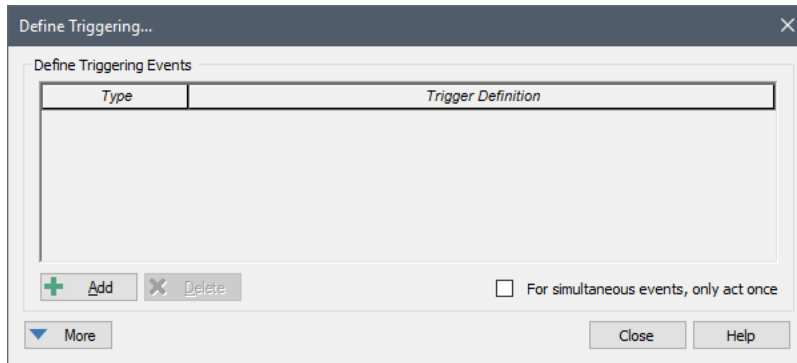
---

**Note:** Holding your cursor over the trigger button displays a tool-tip summarizing the current trigger settings.

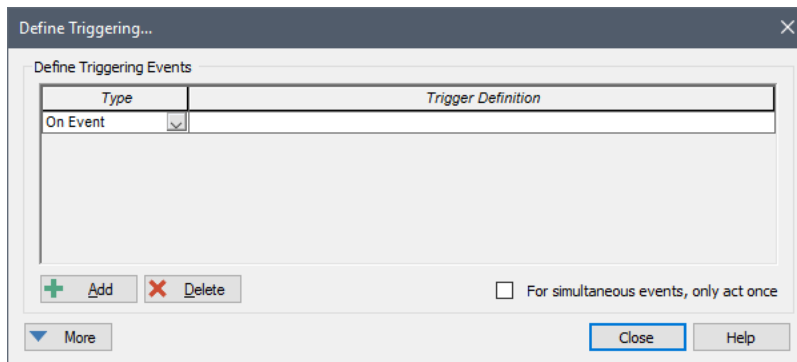
---

The Triggering dialog for every element looks like this:

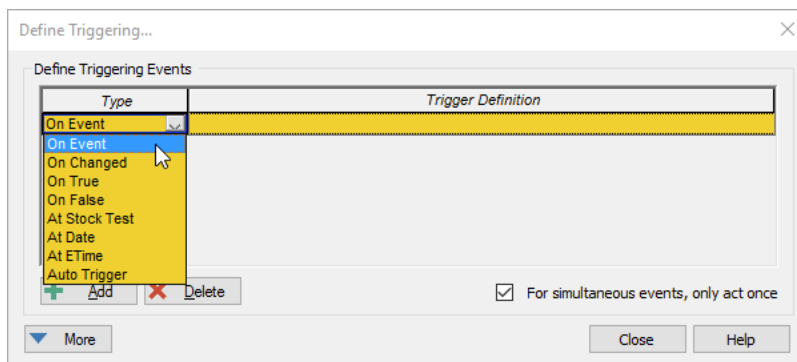




To define a trigger for the element, you simply press the **Add** button to add a row to the list of triggering events:



By default, the **Type** of event added will be "On Event". If you click on the small button in the **Type** column, a drop-list will be presented allowing you to edit the event type:



There are eight types of events that can be added:

**On Event:** The Trigger Definition must be a discrete event signal from another element. The element is triggered whenever the signal is received.

**On Changed:** The Trigger Definition can be any continuous output (it cannot be an expression or a discrete signal). The element is triggered whenever the value of Trigger Definition changes.

**On True:** The Trigger Definition can be any condition output or conditional expression. The element is triggered whenever the Trigger Definition *becomes* True.

**On False:** The Trigger Definition can be any condition output or conditional expression. The element is triggered whenever the Trigger Definition *becomes* False.

**At Stock Test:** The Trigger Definition must be a conditional expression of the form " $A > B$ ", " $A \geq B$ ", " $A < B$ ", " $A \leq B$ ", or " $A = B$ " where A is the primary output of a Reservoir, a Pool or a Fund. The element is triggered whenever the Trigger Definition *becomes* True. As discussed below, this triggering event interrupts the clock and adds an unscheduled update.

**At Date:** The Trigger Definition must be a date or date and time, enclosed in quotations. (Alternatively, the date can also be expressed as the amount of time since 30 December 1899). The element is triggered whenever the simulated date reaches the specified date. As discussed below, this triggering event interrupts the clock and adds an unscheduled update.

**At ETime:** The Trigger Definition must be an elapsed time. The element is triggered whenever the simulated elapsed time reaches the specified elapsed time. As discussed below, this triggering event interrupts the clock and adds an unscheduled update.

**Auto Trigger (only available for some elements):** An Auto Trigger requires no user-defined Trigger Definition and its behavior is defined by its context (i.e., the type of element). Auto Triggers react to the activation or deactivation of their parent Container.

---

**Note:** *At Stock Test*, *At Date* and *At ETime* triggers interrupt the clock and insert an unscheduled update when they occur (whereas *On True*, *On False* and *On Changed* triggers do not create an unscheduled update). To understand the implications of this, consider an example in which your scheduled updates were every 10 days. There are two different ways you could try to trigger an event when the value of Reservoir A became greater than the value B. You could create an *At Stock Test* trigger of  $A > B$ , or you could create an *On True* trigger of  $A > B$ . If we assume that  $A > B$  actually became true at 15 days, these two triggers would behave very differently. The *At Stock Test* trigger would catch this point exactly, and insert an unscheduled update at 15 days. In the absence of any other unscheduled updates, however, the *On True* trigger would not be evaluated and implemented until 20 days. Similarly, if you wished to trigger an event at a specific elapsed time (e.g., 17 days), you could try to do so in two different ways. You could trigger the event using an *At ETime* trigger of 17 days, or you could create an *On True* trigger with  $ETime \geq 17 \text{ days}$ . Again, these two triggers would behave very differently. The *At ETime* trigger would catch this point exactly, and insert an unscheduled update at 17 days. In the absence of any other unscheduled updates, however, the *On True* trigger would not be evaluated and implemented until 20 days.

---

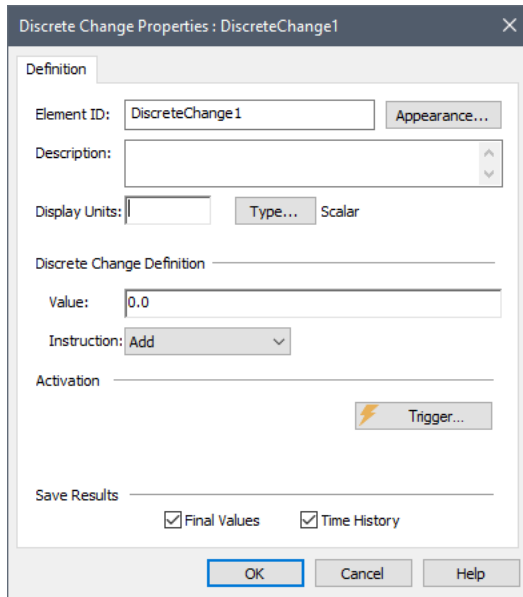
The **More** button provides access to advanced triggering options.

## **Generating Discrete Change Signals in Financial Models**

The Discrete Change element is triggered by an event, and responds by emitting a discrete change signal. The triggering event can be a discrete event signal or another type of event (e.g., a condition, such as X becoming greater than Y).

For financial models, Discrete Changes are used to represent discrete financial transactions (e.g., deposits, withdrawals, purchases, sales).

The dialog for a Discrete Change element looks like this:



You first specify when the element is to be triggered via the **Trigger...** button in the Discrete Change dialog. The **Trigger...** button provides access to a standard Trigger dialog.

Once the element is triggered, it emits a discrete change signal. A discrete change signal contains information regarding the response to an event and consisting of two pieces of information: a *Value* (e.g., 10 dollars) and an *Instruction* (e.g., Add).

For financial models, the **Value** always has dimensions of currency, and can be a number (e.g., 10 \$) or a link. The required order (scalar, vector or matrix) of the Value is specified in a dialog accessed via the **Type...** button. Financial Module elements generally only accept scalar discrete change signals (with the exception of Investment elements, which also accept vectors).

The **Instruction** drop-list contains two choices: "Add" and "Replace". This instructs the element(s) receiving the discrete change signal how to act upon the value being received.

## Overview of the GoldSim Financial Module






This section is intended to provide you with a broad overview of the GoldSim Financial Module. This is done by providing a brief introduction to each of the elements provided by the Financial Module.

This section also briefly illustrates how you can utilize GoldSim's capabilities to run probabilistic simulations (using Monte Carlo simulation), how to view and analyze results, and how financial models can be documented to make them transparent to others.

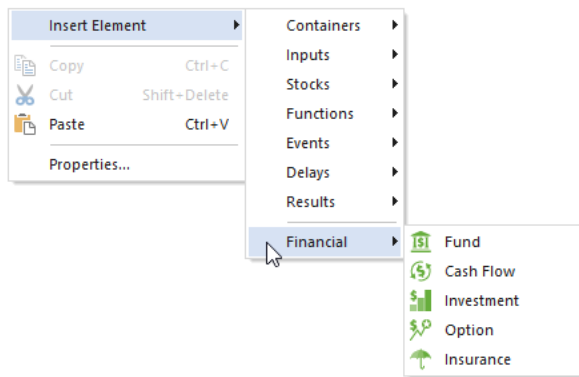
### The Financial Elements

The first step to building a financial model in GoldSim, as it is in any other financial modeling methodology, is to develop a model of the system of interest with all of its components. In GoldSim, the building blocks used to represent the components of the system are the financial elements themselves.

GoldSim currently provides five types of financial elements:

Element	Default Symbol	Function
Fund		Simulates funds and accounts with specified deposits, withdrawals, and interest rates. Outputs include the fund balance and the cumulative interest paid.
Cash Flow		Computes the net present value (NPV) and internal rate of return (IRR) of a cash flow history. Used to model the future return of projects, business ventures, and similar undertakings.
Investment		Simulates an investment such as a security or portfolio of securities. Inputs include purchases and sales, and a history of the underlying security's unit value. Primary output is the investment's value.
Option		Simulates the acquisition and exercise of financial options (puts and calls). Inputs include option type (American, European, Asian), terms and strike price, as well as triggers for acquiring and exercising the option. Primary output is current value (if exercised).
Insurance		Simulates claims against an insurance policy. Inputs include the deductible and cap for the policy, as well as the claims. Outputs include the cumulative covered and uncovered losses on the claims.

Like all modules, these elements are accessed in menus separately from the standard GoldSim elements:



These elements are described further in the following sections.

---

**Note:** In order to make the Financial Module elements available, you must activate the module.

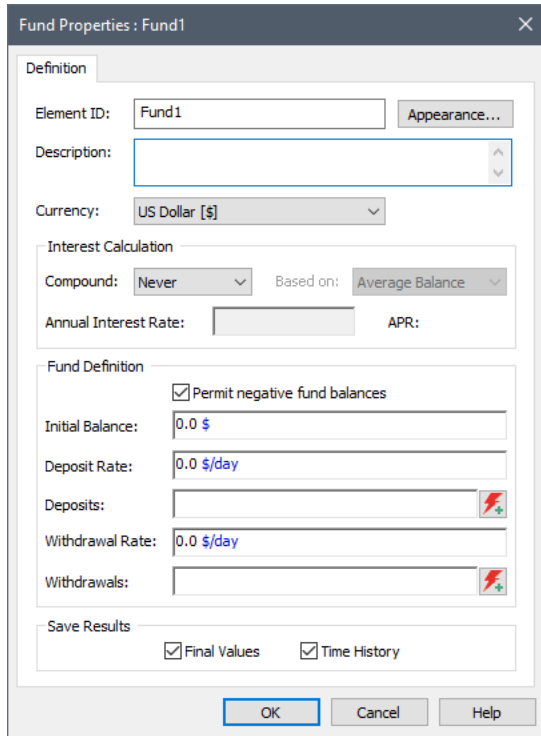
---

**Read More:** *Activating the Financial Module on page 7*

### **Overview of the Fund Element**

The Fund element simulates funds and accounts with specified deposits, withdrawals, and interest rates. This element is used to model bank accounts and loans and other financial components that are governed by an interest rate.

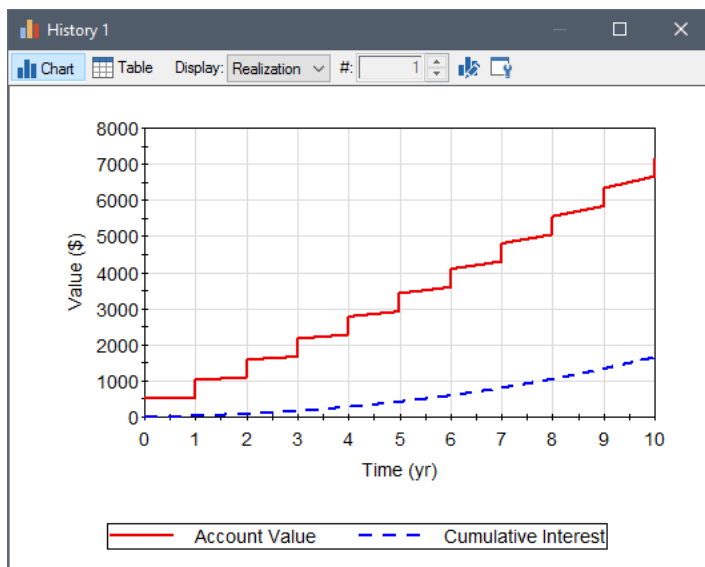
The property dialog looks like this:



After specifying the currency units you wish to use, you must specify how interest is calculated. This includes the interest rate, the type of compounding (e.g., continuous, quarterly, annually) and how the compounding is computed (i.e., based on an average or minimum balance).

You then specify an initial balance, and deposits and withdrawals (which can be entered as continuous rates or discrete items).

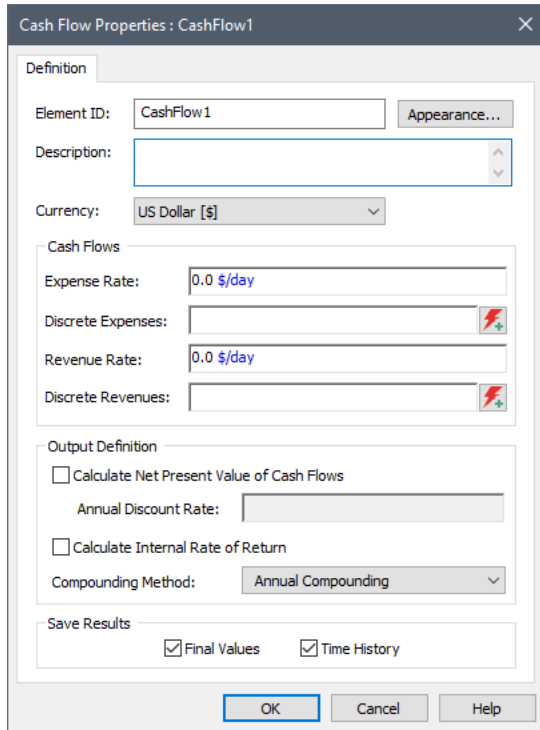
The key outputs are the fund balance and the cumulative interest paid:



## Overview of the Cash Flow Element

The Cash Flow element computes the *net present value* (NPV) and *internal rate of return* (IRR) of a cash flow history. This element is used to model the future return of projects, business ventures, and similar undertakings.

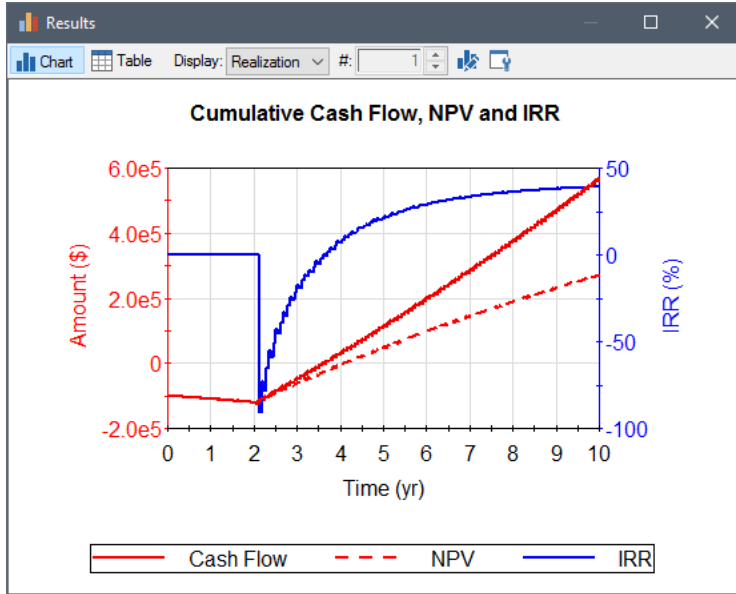
The property dialog looks like this:



After specifying the currency units you wish to use, you must specify expenses and revenues (which can be entered as continuous rates or discrete items).

You then specify whether you want to compute the NPV, the IRR (or both). Calculation of the NPV requires a discount rate.

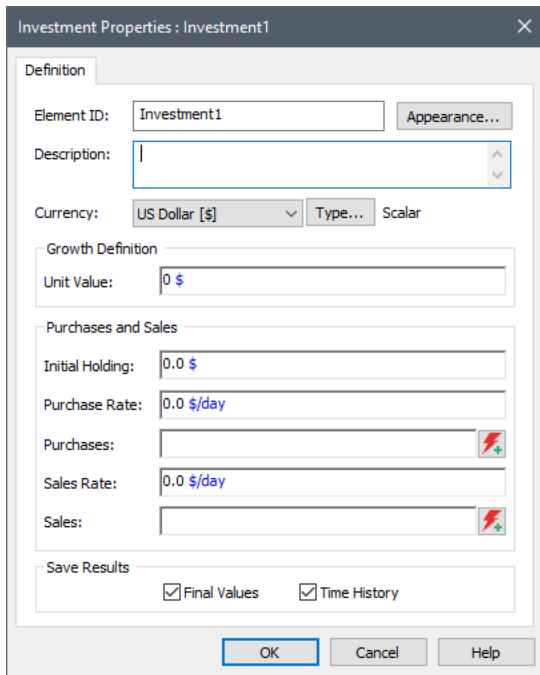
In addition to the NPV and the IRR, the element also outputs the cumulative (undiscounted) cash flow (revenues - expenses):



### Overview of the Investment Element

The Investment element simulates an investment such as a security or portfolio of securities.

The property dialog looks like this:



After specifying the currency units you wish to use, you must specify how the growth of the investment is to be computed. This can be any function, but will often be the output of a History Generator element. The History Generator element (which is part of the basic GoldSim framework), provides a variety of alternatives for generating time histories, including specifying a stochastic growth rate (defined using a mean annual growth rate and volatility, and simulated as a Wiener process).

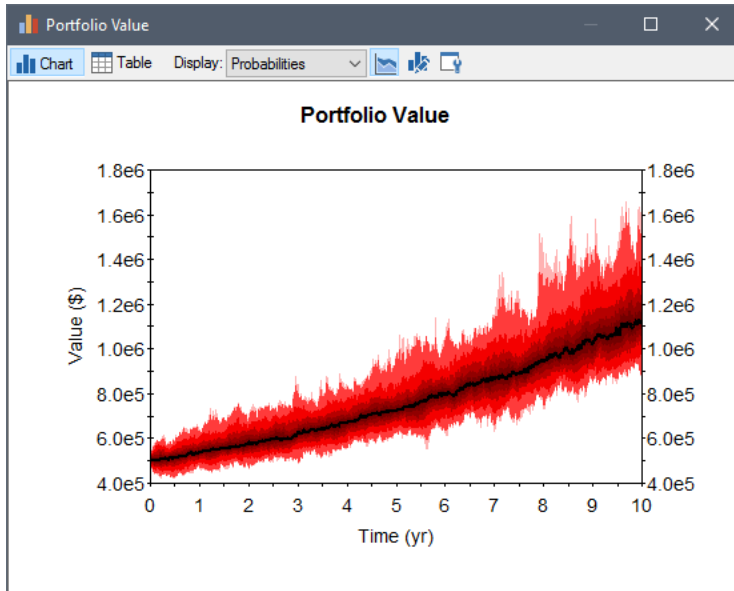
**Note:** The History Generator element is discussed in detail in Chapter 10 of the [GoldSim User's Guide](#).

---

You can use an Investment element to model a collection of investments (by entering vectors of inputs), and you can specify a correlation matrix for the investments (using several different correlation schemes).

Purchases and sales of the investment can be entered as continuous rates or discrete items.

The primary output is the current value of the investment:

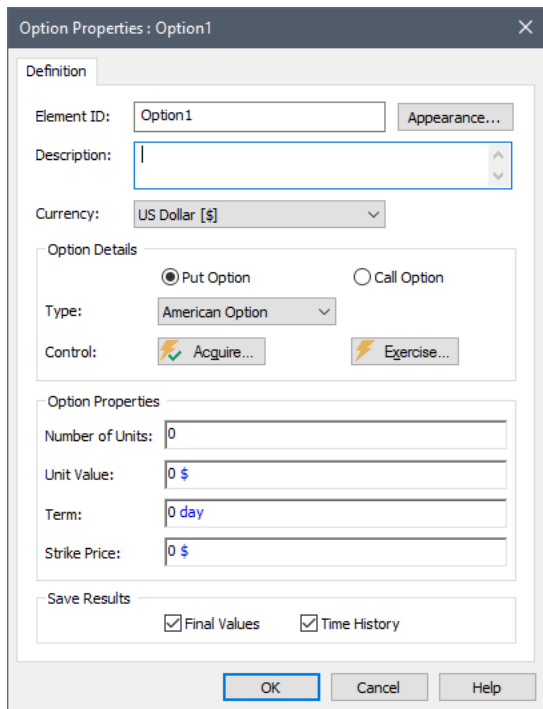


### ***Overview of the Option Element***

The Option element simulates the acquisition and exercise of financial options (puts and calls).

The property dialog looks like this:





After specifying the currency units you wish to use, you must specify whether you are simulating a put or a call, along with the option type (American European, Asian).

The unit value of the underlying security must also be specified. This can be any function, but will often be the output of a History Generator element. This element (which is part of the basic GoldSim framework), provides a variety of alternatives for generating time histories, including simply specifying a constant annual growth rate, and specifying a stochastic growth rate (defined using a mean annual growth rate and volatility, and simulated as a Wiener process)

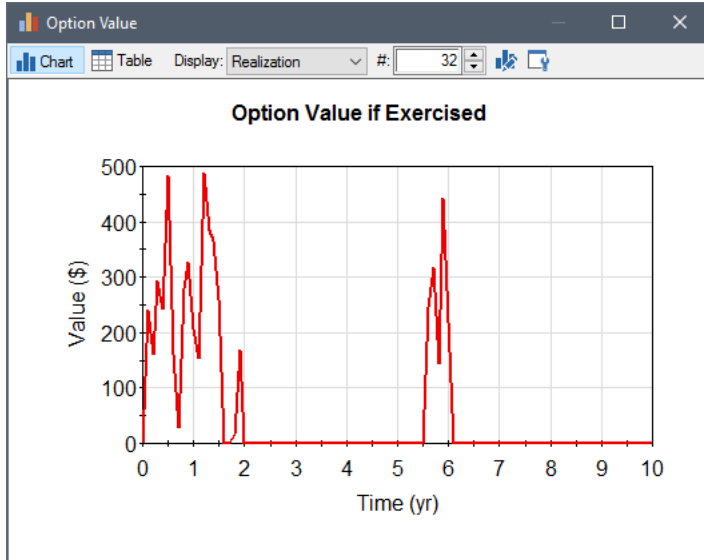
---

**Note:** The History Generator element is discussed in detail in Chapter 10 of the [GoldSim User's Guide](#).

---

Triggers must be specified for acquiring and exercising the option. You must also specify the properties of the option at the time of acquisition (i.e., number of units, term and strike price).

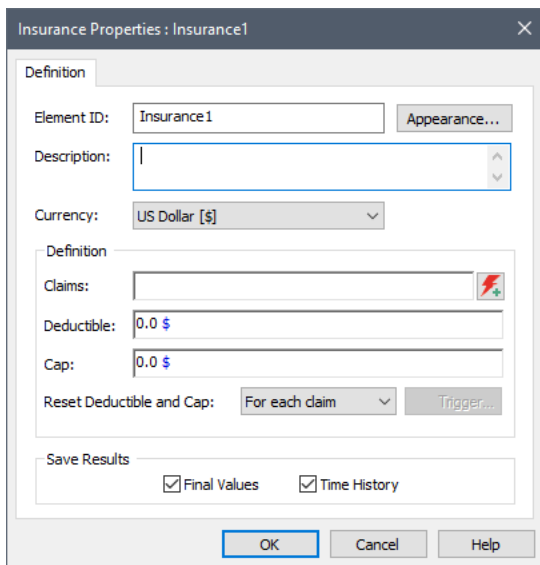
The primary output is current value (if exercised), and a discrete change signal with the value of the exercised options:



### Overview of the Insurance Element

The Insurance element simulates claims against an insurance policy.

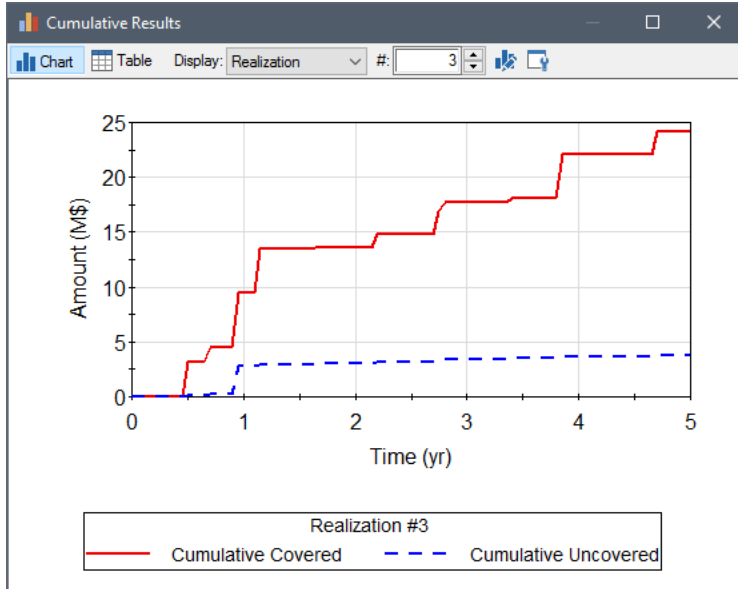
The property dialog looks like this:



After specifying the currency units you wish to use, you must specify the properties of the insurance policy (the deductible, the cap, and when these are reset).

You then specify one or more discrete claims.

The key outputs are the cumulative covered and uncovered losses on the claims, and discrete changes representing the amounts that are covered and uncovered:



## The Built-in Financial Functions

In addition to the elements provided by the Financial Module, you will also likely want to take advantage of a number of built-in financial functions provided by GoldSim.

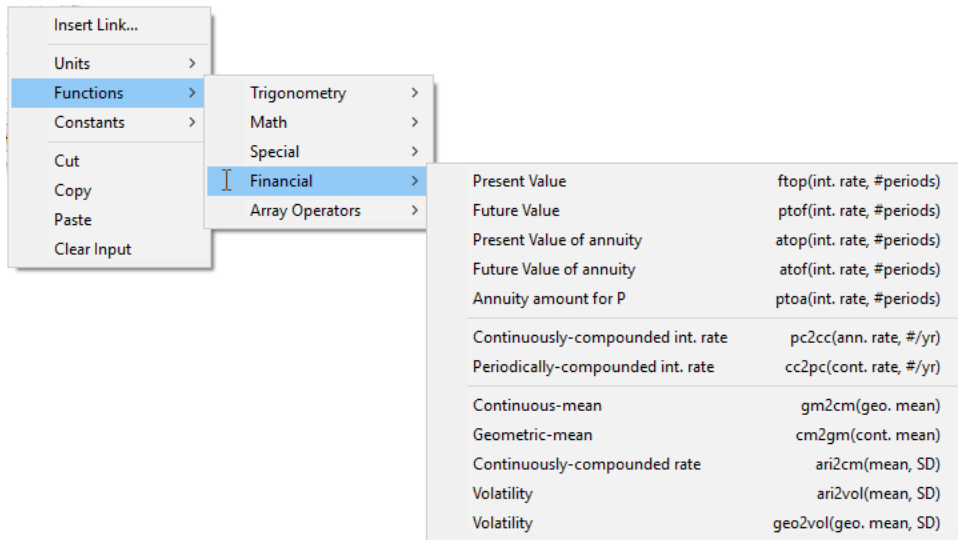
GoldSim provides a wide variety of built-in functions (such as sin, cos, min and max) that can be used in expressions within your models.

---

**Note:** GoldSim's built-in functions are described in detail in Chapter 3 of the [GoldSim User's Guide](#).

---

The financial functions are accessed by right-clicking in an input field and selecting **Functions | Financial:**



### Financial Functions: Simple Compounding

The first five financial functions are simple compounding functions that account for the time value of money and convert between present value, future value and annuities. These are dimensionless functions that act as multiplying factors on a currency amount (e.g., \$) or currency rate (e.g., \$/yr).

In these functions the first argument (int. rate) is the fractional interest per period (expressed as either a fraction or a percentage), and the second argument (#periods) is the number of periods:

Function	Description	Definition
f <sub>top</sub> (int. rate, #periods)	Factor that when multiplied by a future amount returns the present value of that future amount.	$(1 + i)^{-n}$
p <sub>tof</sub> (int. rate, #periods)	Factor that when multiplied by a present amount returns the future value of that present amount.	$(1 + i)^n$
a <sub>top</sub> (int. rate, #periods)	Factor that when multiplied by an annuity amount returns the present value of that annuity.	$\frac{(1 + i)^n - 1}{i(1 + i)^n}$
a <sub>tof</sub> (int. rate, #periods)	Factor that when multiplied by an annuity amount returns the future value of that annuity.	$\frac{(1 + i)^n - 1}{i}$
p <sub>toa</sub> (int. rate, #periods)	Factor that when multiplied by a present amount returns the annuity of that present amount.	$\frac{i(1 + i)^n}{(1 + i)^n - 1}$

*int. rate (i): Must be greater than zero. Can be a scalar or an array*  
*#periods (n): Must be a positive scalar.*

These functions assume discrete compounding. That is, the functions assume compounding once per period.

Since #periods represents the number of periods in these functions, it must be a unitless value. For example, in order to compute the annuity value (the annual payment amount) for a 15 year loan of \$100,000 with an interest rate of 7% per year (compounded annually), you would write the following in an expression:

Equation

```
100000 $ * ptoa(7%,15)
```

### Financial Functions: Continuous/Periodic Compounding Conversions

The next two financial functions are used to convert a continuously compounded interest rate to an equivalent periodically compounded interest rate, and vice-versa.

In these functions the first argument (either ann. rate or cont. rate) is the fractional interest rate (expressed as either a fraction or a percentage), and the second argument (#/yr) is the number of times the equivalent periodically compounded rate is compounded per year:

Function	Description	Definition
pc2cc(ann. rate, #/yr)	Converts an annual interest rate (ann.rate) that is compounded #/yr per year to an equivalent continuously compounded interest rate.	$n * \ln \left( 1 + \frac{Arate}{n} \right)$
cc2pc (cont. rate, #/yr)	Converts a continuously compounded interest rate (cont.rate) to an equivalent periodically compounded annual rate that is compounded #/yr per year.	$n * \left[ \exp \left( \frac{Crate}{n} \right) - 1 \right]$

ann. rate (Arate): Must be greater than or equal to zero. Can be a scalar or an array  
 cont. rate (Crate): Must be greater than or equal to zero. Can be a scalar or an array  
 #/yr (n): Must be a positive scalar.

For example, the following expression computes the continuously compounded interest rate that is equivalent to an annual interest rate of 5%, compounded daily:

```
Current Value = 0.04999657565517335
pc2cc(5%, 365)
```

### Financial Functions: Return Rates and Volatility

The last five financial functions are used to convert between different types of financial return statistics.

In order to understand these functions, it is first necessary to provide some definitions.

Consider a time series (e.g., of the price of a stock):  $P_t, P_{t+1}, P_{t+2}, \dots$ , where  $P_x$  is the price at year  $x$ . The annual return can be defined in several different ways, the most common of which are:

- The **annual return** ( $R_x$ ) between year  $x$  and year  $x+1$  is defined as:

$$R_x = \frac{(P_{x+1} - P_x)}{P_x}$$

- The **log annual return** ( $LR_x$ ) between year  $x$  and year  $x+1$  is defined as:

$$LR_x = \ln \left( \frac{P_{x+1}}{P_x} \right)$$

The log annual return is equivalent to the continuously-compounded rate.

Given multiple years of data, these two annual return definitions represent time series of data. Mean annual returns of such a series can be defined in multiple ways:

- The **geometric mean annual return** ( $GR_x$ ) at time  $x$  given a series of  $n$  annual returns from year  $x-n$  to  $x$  is defined as:

$$GR_x = \left( \prod_{i=x-n+1}^x (1 + R_i) \right)^{1/n} - 1$$

- The **arithmetic mean annual return** ( $AR_x$ ) at time  $x$  given a series of  $n$  annual returns from year  $x-n$  to  $x$  is defined as:

$$AR_x = \frac{\sum_{i=x-n+1}^x R_i}{n}$$

- The **arithmetic mean log annual return** ( $ALR_x$ ) at time  $x$  given a series of  $n$  annual returns from year  $x-n$  to  $x$  is defined as:

$$ALR_x = \frac{\sum_{i=x-n+1}^x LR_i}{n} = \ln(GR_x + 1)$$

The arithmetic mean log annual return is equivalent to the mean continuously-compounded rate.

The geometric mean annual return is the most common statistic used to describe mean annual returns. Hence, when the multiyear return for a mutual fund or stock is quoted, this is typically a geometric mean annual return

In addition to mean returns, the standard deviation of the series of returns can also be computed:

- The **standard deviation of the annual return** ( $SDR_x$ ) at time  $x$  given a series of  $n$  annual returns from year  $x-n$  to  $x$  is defined as:

$$SDR_x = \sqrt{\frac{\sum_{i=x-n+1}^x (R_i - AR_x)^2}{n - 1}}$$

- The **standard deviation of the log annual return** ( $SDLR_x$ ) at time  $x$  given a series of  $n$  annual returns from year  $x-n$  to  $x$  is defined as:

$$SDLR_x = \sqrt{\frac{\sum_{i=x-n+1}^x (LR_i - ALR_x)^2}{n - 1}}$$

The standard deviation of the log annual return is also referred to as the volatility.

The last five financial functions are used to convert between the statistics describing financial time series:

Function	Description
gm2cm (geo. mean)	Converts a geometric mean annual return (GR) to a mean continuously-compounded rate (i.e., an arithmetic mean log annual return (ALR)).
cm2gm (cont. mean)	Converts a mean continuously compounded rate (i.e., an arithmetic mean log annual return (ALR)) to a geometric mean of the annual return (GR).
ari2cm (mean, SD)	Converts an arithmetic mean annual return (AR) and a standard deviation of the annual return (SDR) to an arithmetic mean log annual return (ALR).

Function	Description
ari2vol (mean, SD)	Converts an arithmetic mean annual return (AR) and a standard deviation of the annual return (SDR) to the standard deviation of the log annual return (SDLR), also known as the volatility.
geo2vol (geo. mean, SD)	Converts a geometric mean annual return (GR) and a standard deviation of the annual return (SDR) to the standard deviation of the log annual return (SDLR), also known as the volatility.

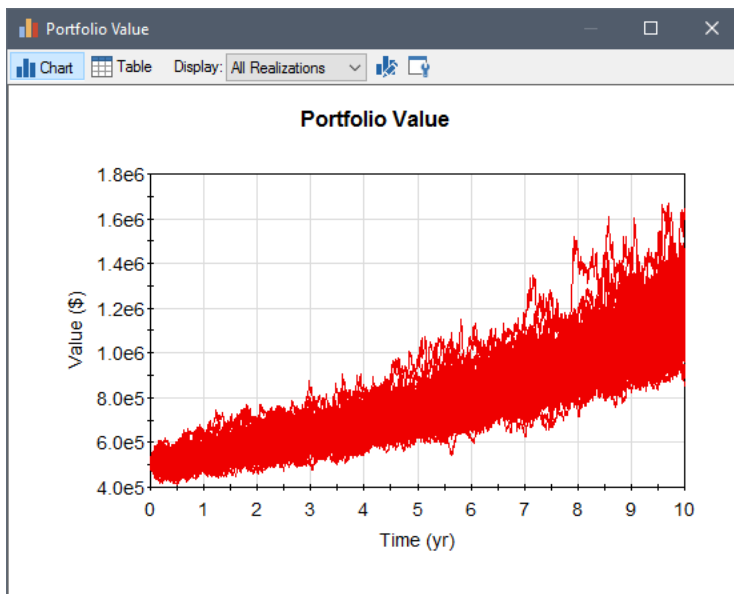
*geo. mean, cont. mean, mean (GR, ALR, AR): Must be greater than zero.  
 SD (SDLR): Must be greater than or equal to zero.  
 Arguments can be scalars or arrays.*

## Using GoldSim's Probabilistic Simulation Engine

Once a model of a financial system has been constructed, we can simulate the system to predict how it will perform through time. By definition, however, the performance of such a system is stochastic (i.e., inherently variable). That is, we can't say exactly how an investment will perform; we can only use historical data to describe the behavior statistically (e.g., based on a mean growth and a volatility).

In addition to this inherent variability, we might also be uncertain about some of the input parameters controlling the model. For example, if a security was new, we would not have a good statistical basis for defining controlling factors like mean growth and volatility. In this case, we could enter these as probability distributions in order to capture this uncertainty.

Variability and uncertainty are represented in GoldSim using Monte Carlo simulation. Monte Carlo simulation consists of calculating a large number of "realizations" (potential futures):



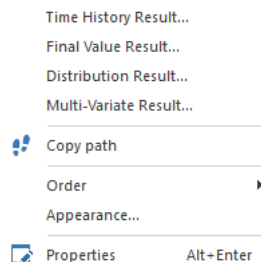
*In this example, 100 equally probable realizations of potential future behavior of an investment are plotted. These realizations were generated using Monte Carlo simulation.*

**Read More:** [Using Monte Carlo Simulation in Your Financial Model](#) on page 76

## Viewing and Analyzing Results

After running a simulation, GoldSim provides a number of ways to view the results.

Right-clicking on an element with a primary output (in the graphics pane or browser) or on a specific output (in an output interface or a browser) which has been saved will provide a context menu for displaying results.

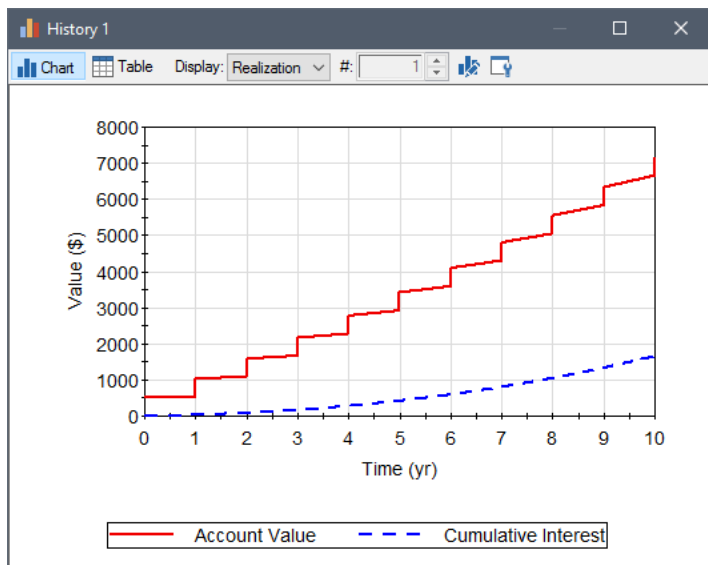


*Depending on the type of output and which results have been saved, different options will be displayed at the top of the menu.*

Clicking on one of the first four items in this menu displays the results for the selected object, either in the form of a chart or a table (depending on user-defined default settings).

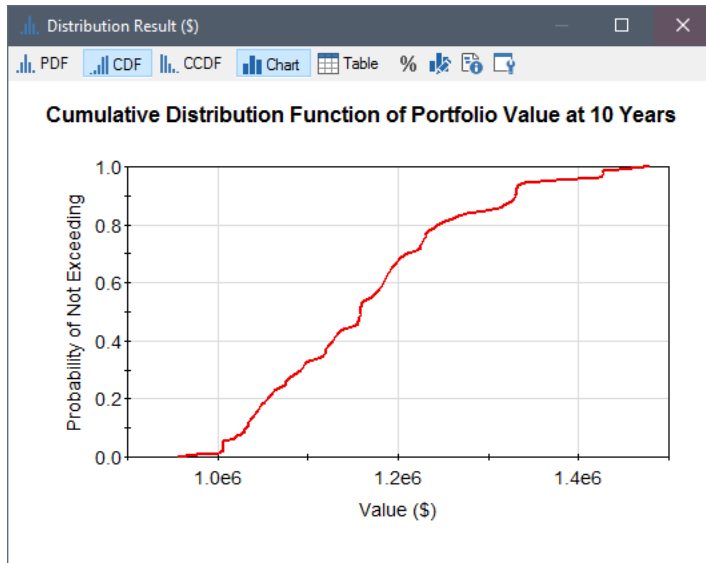
**Note:** You can also add Result elements to you models. Result elements display a particular result that you have specified when you click on them. Result elements are discussed in detail in Chapter 8 of the [GoldSim User's Guide](#).

Time History results show the "history" of a particular output as a function of time, and are probably the most common form of result display you will see:



You can also view probability distributions of a Final Value. This result is only available if you have carried out multiple realizations:





Other result options allow you to view bar, column, pie and scatter plots.

**Read More:** *Chapter 4: Running a Financial Simulation and Viewing Results* on page 73

## Documenting Your Financial Model

GoldSim provides tools that allow you to internally document your model such that the documentation becomes part of the model itself, and hence is immediately available to anyone viewing the model.

GoldSim allows you to add text, images and other graphic objects directly to your model. In addition, you can add hyperlinks to other documents (e.g., a web site or a report). Clicking on the hyperlink then opens that document.

In addition, GoldSim was specifically designed to allow you to organize model elements into a hierarchy (using *Containers*). This facilitates the creation of "top-down" models, in which the level of detail increases as you "push down" into the containment hierarchy. Such a capability is essential if you wish to effectively describe and explain your model at different levels of detail to different audiences. For example, your manager may only want to see the "big picture", while a technical colleague may want to see the low-level details of a particular model.

The ability to create hierarchical, top-down models, in which at any level, details can be "hidden" (inside containers), coupled with GoldSim's powerful documentation features, allows you to design models which can be effectively explained to any audience at the appropriate level of detail.

---

**Note:** Techniques and tools for documenting GoldSim models are discussed in detail in Chapter 9 of the [GoldSim User's Guide](#).

---

## A Simple Financial Module Example

This section provides a hands-on introduction to the Financial Module by walking through the steps necessary to build and edit a simple financial model. Although it is not necessary to do so, it would be helpful if you actually followed along and built and edited the simple model as it is discussed below.

In order to do so, however, you must be able to carry out the basic GoldSim actions discussed in the GoldSim Tutorial (e.g., adding elements, modifying simulation settings, running a model). Therefore, if you wish to build

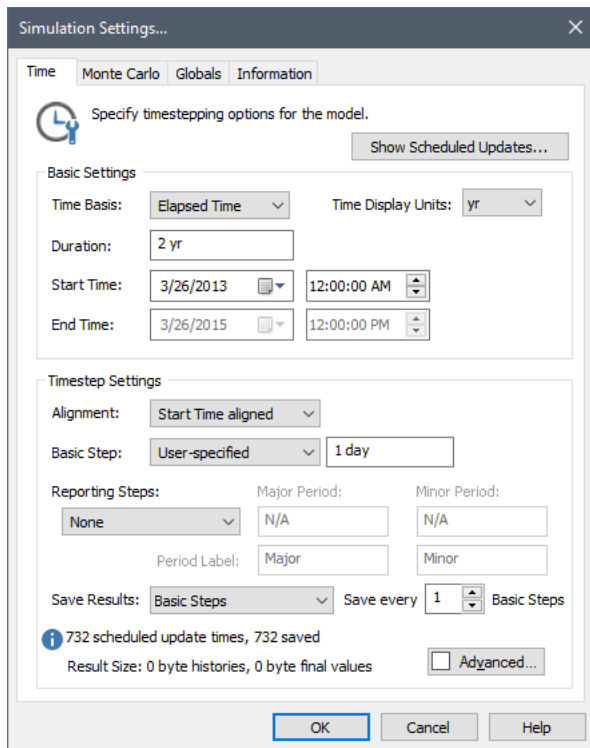
and edit the model described below, and you have not already taken the Tutorial, you should do so now. The GoldSim Tutorial can be accessed by clicking the **Tutorial** link on the GoldSim splash screen, or by selecting **Help|Tutorial** from the GoldSim main menu.

### Step 1: Setting Up a Dynamic Financial Model

We will explore the Financial Module by constructing a simple GoldSim model that utilizes one Financial Module element (the Fund), along with a number of elements from the basic GoldSim framework.

If you wish to follow along, you can start to do so now by opening GoldSim (or if GoldSim is already open, by creating a new model by pressing the new model button or **Ctrl+N**). Although it would be most constructive for you to follow along and build the model yourself, the completed model (named Fund.gsm) can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu).

When specifying the simulation settings (accessed by pressing **F2**), specify that the simulation will be an Elapsed Time simulation with a duration of two years (2 yr), using a 1 day timestep length. Define the **Time Display Units** in years (yr):

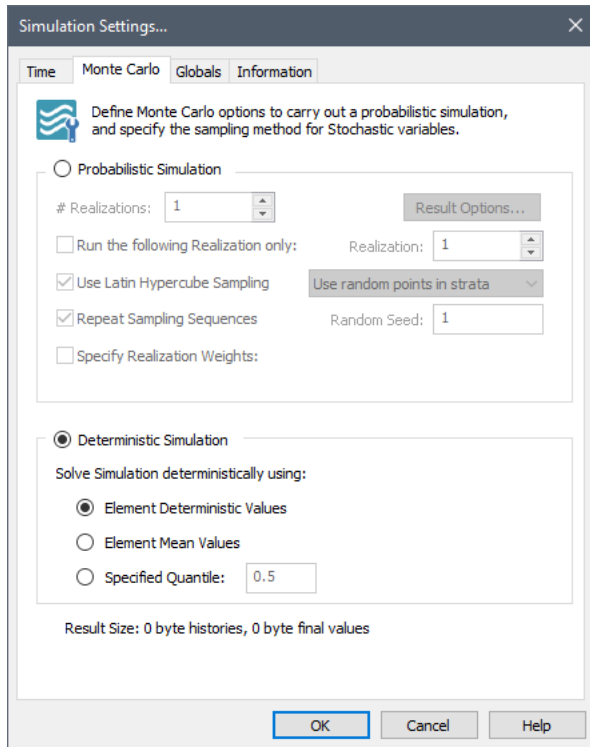


---

**Note:** The Start Time will default to the current date. In this model, it is not used.

---

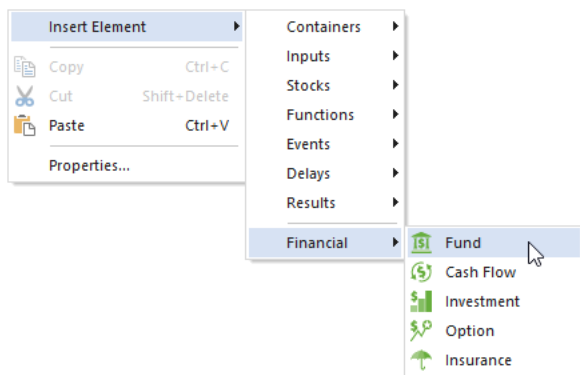
We will start by running a single Deterministic simulation. In the **Monte Carlo** tab of the Simulation Settings dialog, select the Deterministic Simulation option:



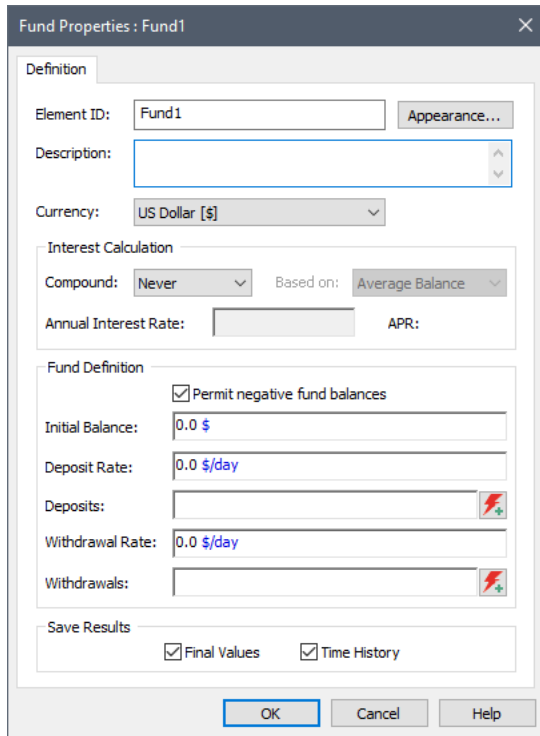
## Step 2: Adding a Fund Element to Simulate a Bank Account

Let's start by inserting a new Fund element into the new model. We will use this element to simulate a bank account.

We can do this by right-clicking in the graphics pane, and selecting **Insert Element|Financial|Fund Element** from the context menu:

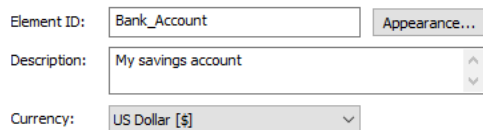


This will insert the element and open the Fund element's property dialog:



If this is your first close look at a financial element, you will notice that it has some common features with the other basic GoldSim elements that you've seen in the GoldSim Tutorial. The **Element ID** and **Description** fields are identical (and the same naming rules apply to reliability elements as to normal GoldSim elements). The **Appearance** button and Save Results checkboxes are also common to all GoldSim elements.

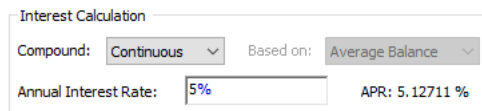
Let's name this element "Bank\_Account" and give it a simple description:



Below the Description field, you specify the Currency units you wish to use. GoldSim will list all of the currencies that are in its database (by default, this is dollars, euros, pounds sterling and yen, and you can easily add any additional currencies by selecting **Model|Currencies** from the main menu). For now, we will choose dollars.

**Read More:** *Managing Currencies and Exchange Rates* on page 70

It is then necessary to specify how interest is calculated. In this example, let's assume a 5% annual interest rate, compounded continuously:




The bottom part of the dialog allows you to specify an Initial Balance, as well as deposits and withdrawals. For now, let's just specify an Initial Balance of 1000\$. We will add some deposits and withdrawals in subsequent steps:

Fund Definition


Permit negative fund balances

Initial Balance: 1000 \$

Deposit Rate: 0.0 \$/yr

Deposits: 

Withdrawal Rate: 0.0 \$/yr

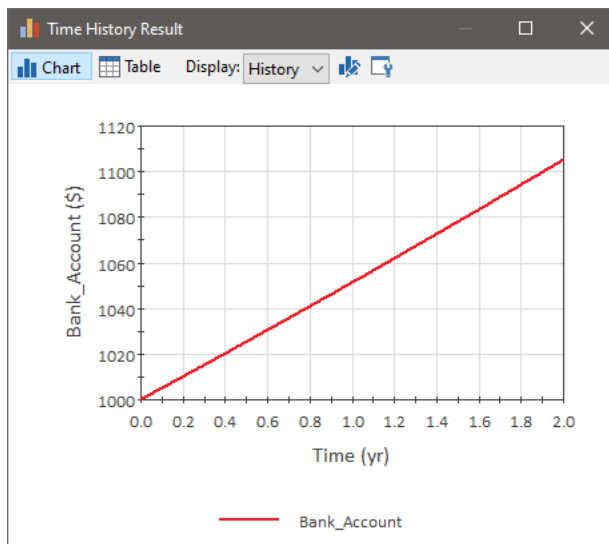
Withdrawals: 

**Note:** In GoldSim, units are added after the value, not before. So in this case, entering \$1000 would be incorrect. Instead, we enter "1000 \$".

After entering the Initial Balance, you can press **OK** to close the dialog.

### Step 3: Running the Model and Viewing a Simple Result

We can now run the model by pressing **F5**. After the simulation completes, right-click on the `Bank_Account` element and then left-click on **Time History Result...** in the context menu to view a time history plot of the result:



What are we looking at here? This is actually a plot of the amount of money in the bank account as a function of time. It starts with an initial value of \$1000, and increases to just over \$1100 after 2 years. The line appears to be linear, but it is actually the early portion of an exponential curve. If the interest rate was significantly higher, or the simulation was longer, the curve would be more obviously exponential.

Before thinking about this result plot further, it's important to first discuss how GoldSim moves the simulation through time. GoldSim "updates" the model as follows:

- The model is updated (all the elements in the model are recalculated) at the timesteps specified by the user. In this case, this means that element values are recalculated at 0 day, 1 day, 2 day, ..., 730 days. Values for time history plots are recorded at these user specified timesteps.

- You can see the actual values used to create the plot by viewing a table of the time history. To do this, press **Table** button at the top of the time history chart display. The following will then be displayed:

Result:	Bank_Account
Unit:	\$
0 yr	1000
0.0027378508	1000.137
0.0054757016	1000.274
0.0082135524	1000.411
0.010951403	1000.548
0.013689254	1000.685
0.016427105	1000.822
0.019164956	1000.959
0.021902806	1001.096
0.024640657	1001.233
0.027378508	1001.37
0.030116359	1001.507
0.032854209	1001.644
0.03559206	1001.781
0.038329911	1001.918
0.041067762	1002.055
0.043805613	1002.193
0.046543463	1002.33
0.049281314	1002.467

- GoldSim also updates the model upon the occurrence of certain kinds of "events". We currently don't have any such events in our model, but will add some later (in particular, some deposits and withdrawals). Events such as these could happen in the middle of a timestep (e.g., at 3.5 days). In this case, GoldSim would actually insert a new timestep at the point the event occurred (although this point would not be visible in the saved result histories).

## Step 4: Adding Discrete Deposits to the Account

Let's make the model more realistic by adding some periodic deposits, in this case, a payday check every 14 days. To do this we will need to add two elements that are part of the basic GoldSim framework to the model: a Time Event, and a Discrete Change.

First, we need to return to Edit Mode (press **F4** and select **OK**). To insert the Timed Event element, right-click in the graphics pane, and select **Insert Element|Events|Timed Event** from the context menu. A Timed Event element will be inserted and its property dialog will be displayed:

Change the element's name to "Pay\_Day". Timed Events produce discrete signals based on a specified rate of occurrence. They can generate random or regular events. Pay\_Day will be a regular event with a rate of occurrence of once every 14 days:

Event Definition

Occurrence Type: Regular time intervals

Rate: 1/14 day

Next we need to specify the consequence of the Pay\_Day: the generation of a deposit of a particular quantity. We do this by using a Discrete Change element, which is inserted by right-clicking in the graphics pane, and selecting **Insert Element|Events|Discrete Change** from the context menu:

Discrete Change Properties : DiscreteChange1

Definition

Element ID: DiscreteChange1 Appearance...

Description:

Display Units: Type... Scalar

Discrete Change Definition

Value: 0.0

Instruction: Add

Activation Trigger...

Save Results  Final Values  Time History

OK Cancel Help

Change the element's name to "Deposit\_Check". Specify Display Units of \$, and enter a value of 1000 \$ (the amount of the pay check). Leave the Instruction as Add. Finally, we need to specify that Deposit\_Check is triggered whenever the Pay\_Day occurs. To do this, press the **Trigger...** button. In the trigger dialog that is displayed, press the **Add** button, and then in the Trigger Definition field, enter "Pay\_Day":

Define Triggering...

Define Triggering Events

Type	Trigger Definition
On Event <input checked="" type="checkbox"/>	Pay_Day

+ Add - Delete  For simultaneous events, only act once

More Close Help

Close this dialog as well as the Discrete Change dialog.

Finally, we need to tell GoldSim where the check is deposited. To do this, double-click on the Bank\_Account, and enter "Deposit\_Check" in the Deposits input field:

Fund Definition

Permit negative fund balances

Initial Balance: 1000 \$

Deposit Rate: 0.0 \$/yr

Deposits: Deposit\_Check

Withdrawal Rate: 0.0 \$/yr

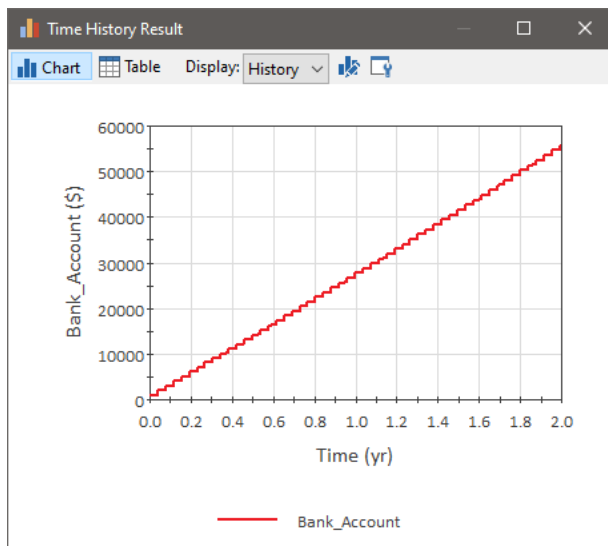
Withdrawals:

After you close the Bank\_Account, your model should look something like this:



## Step 5: Running the Model with Deposits

Let's now run the model again by pressing **F5**. After the simulation completes, right-click on the Bank\_Account element and then left-click on **Time History Result...** in the context menu to view a time history plot of the result:



Notice that our bank account is increasing at a faster rate, as we now have regular discrete deposits (every 2 weeks), as well as accumulation of interest between deposits. After two years, the account contains more than \$55,000.

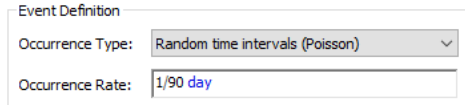
## Step 6: Removing Discrete Withdrawals from the Account

As we all know, our bank accounts don't just have deposits, they also have withdrawals. So let's continue to make the model more realistic by adding some withdrawals, in this case, a purchase (of a random amount) approximately once every three months.

To do this we will again need to add two elements that are part of the basic GoldSim framework to the model: a Timed Event, and a Discrete Change. We will also add a Stochastic element to represent the amount of the purchase.

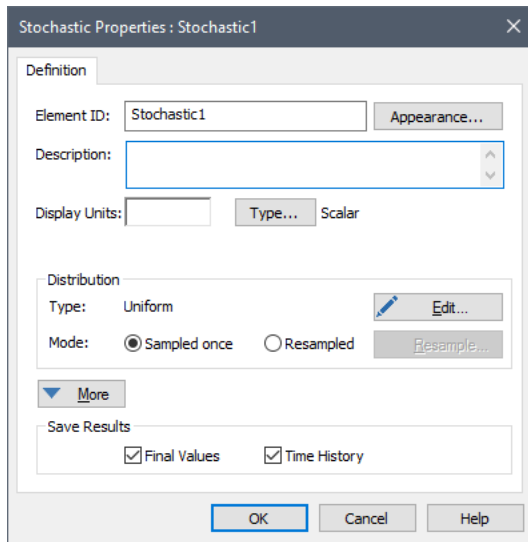


First, we need to return to Edit Mode (press **F4** and select **OK**). Now let's start by inserting another Timed Event element, and changing the element's name to "Purchase". Purchase will be a random (as opposed to a regular) event with an average rate of occurrence of once every 90 days:

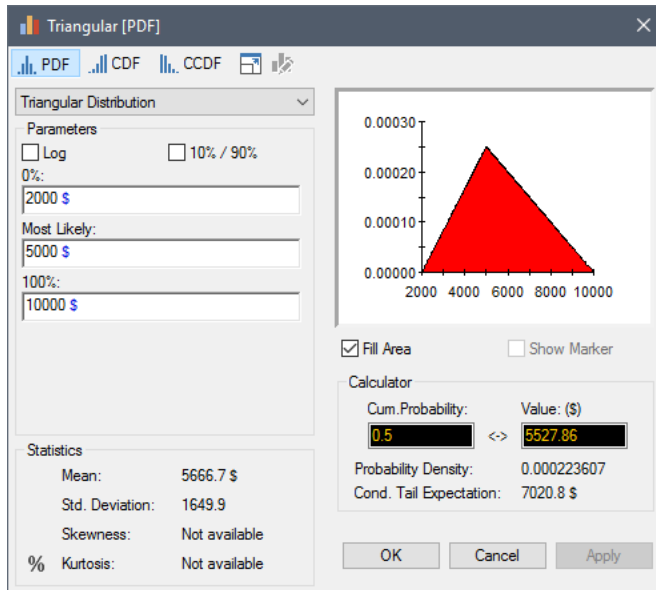


Next we need to specify the consequence of the Purchase: the generation of a withdrawal of a particular quantity. We will do this by using a Discrete Change element. However, before we do that, we need to specify the amount of each purchase. Since unlike a paycheck, purchase amounts are random, we will use a Stochastic element to model this.

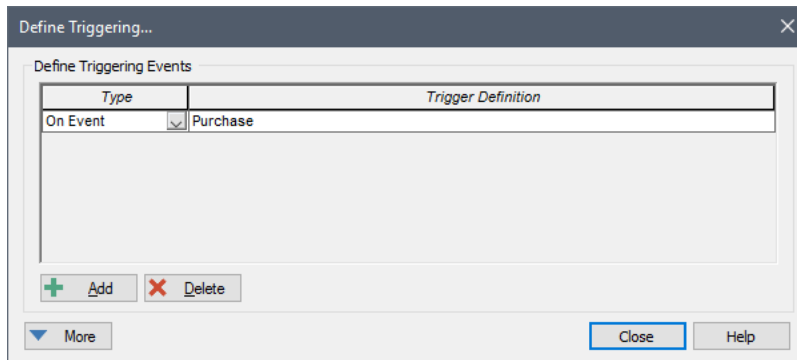
A Stochastic is inserted by right-clicking in the graphics pane, and selecting **Insert Element|Inputs|Stochastic** from the context menu:



Change the element's name to "Purchase\_Amount" and specify Display Units of \$. Next press the **Edit...** button to define the distribution. We will assume a Triangular distribution, with Minimum value of \$2000, a Most Likely value of \$5000, and a Maximum value of \$10000:



Press **OK** to return to the Stochastic dialog. We need to specify that this distribution is resampled whenever a purchase occurs. To do so, select the **Resampled** radio button, and then press the **Resample...** button that becomes active when you do so. In the trigger dialog that is displayed, press the Add button, and then in the Trigger Definition field, enter "Purchase":



Close this dialog as well as the Stochastic dialog.

Now we can create the Discrete Change dialog that represents the actual withdrawal. Insert a Discrete Change element, and name it "Purchase\_Withdrawal". Specify Display Units of \$, and in the Value field, enter "Purchase\_Amount". Leave the Instruction as Add. Next, we need to specify that Purchase\_Withdrawals is triggered whenever the Purchase occurs (just as we did for the Stochastic element). To do this, press the **Trigger...** button. In the trigger dialog that is displayed, press the Add button, and then in the Trigger Definition field, enter "Purchase".

Close this dialog as well as the Discrete Change dialog.

Finally, we need to tell GoldSim where the money is withdrawn from. To do this, double-click on the Bank Account, and enter "Purchase\_Withdrawal" in the Withdrawals input field:

Fund Definition

Permit negative fund balances

Initial Balance: 1000 \$

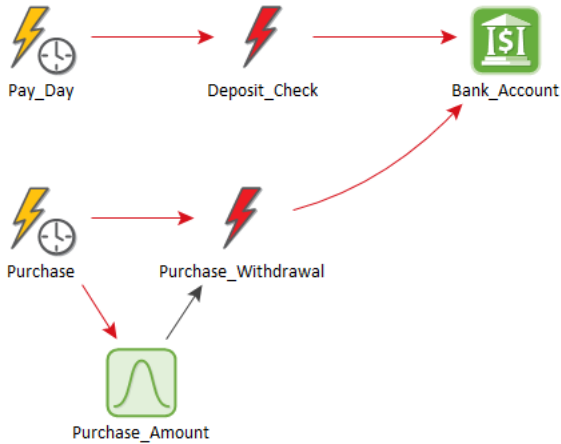
Deposit Rate: 0.0 \$/yr

Deposits: Deposit\_Check

Withdrawal Rate: 0.0 \$/yr

Withdrawals: Purchase\_Withdrawal

After you close the Bank\_Account, your model should look something like this:

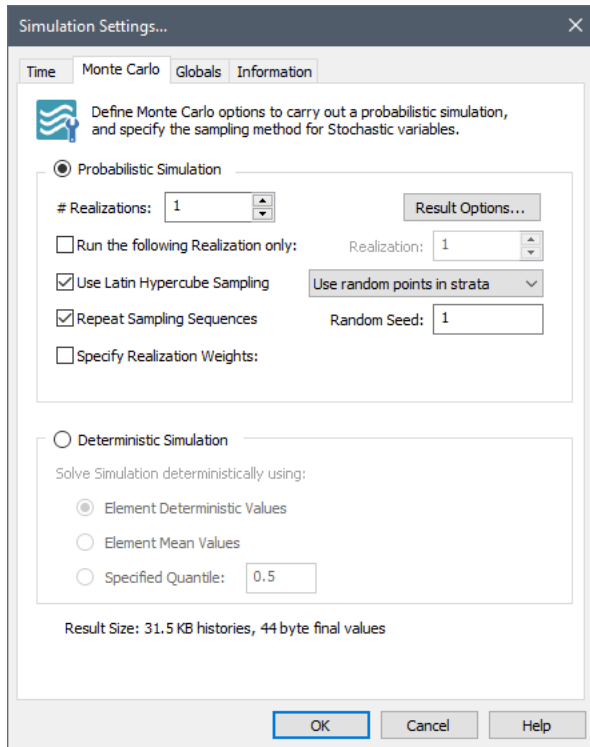


## Step 7: Running the Model with Deposits and Withdrawals

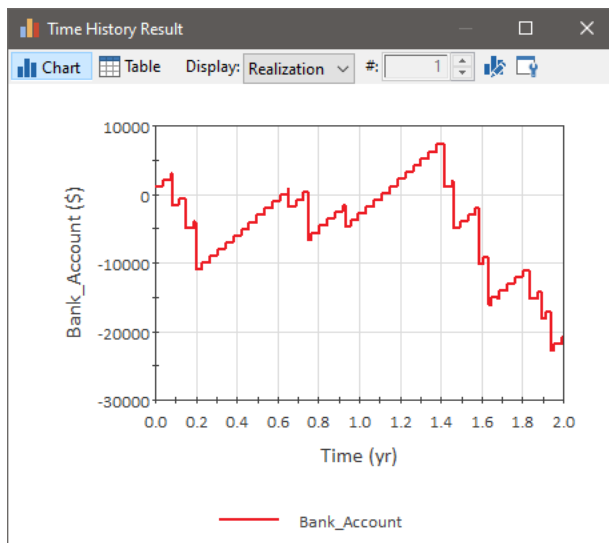
Before we run this model, we want to first make sure that GoldSim will be randomly sampling the purchases and the purchase amounts.

To do this, open the Simulation Settings dialog (press **F2**) and click on the **Monte Carlo** tab. Make sure the **Probabilistic Simulation** radio button is selected, and that the "# Realizations" is set to 1:

## A Simple Financial Module Example



Let's now run the model again by pressing **F5**. After the simulation completes, right-click on the `Bank_Account` element and then left-click on **Time History Result...** in the context menu to view a time history plot of the result:



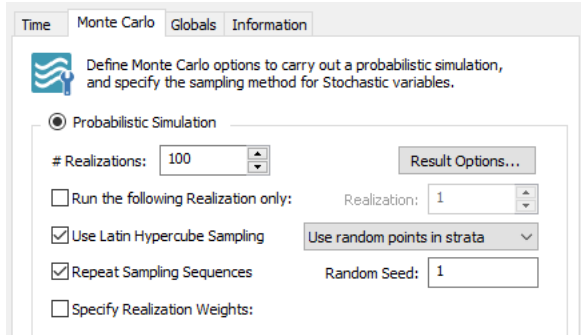
**Note:** Your result plot will be different, since this represents one random realization of the system.

Notice that our bank account now has random withdrawals (in addition to the payday deposits and accumulation of interest).

## Step 8: Running Multiple Realizations of a Financial Model

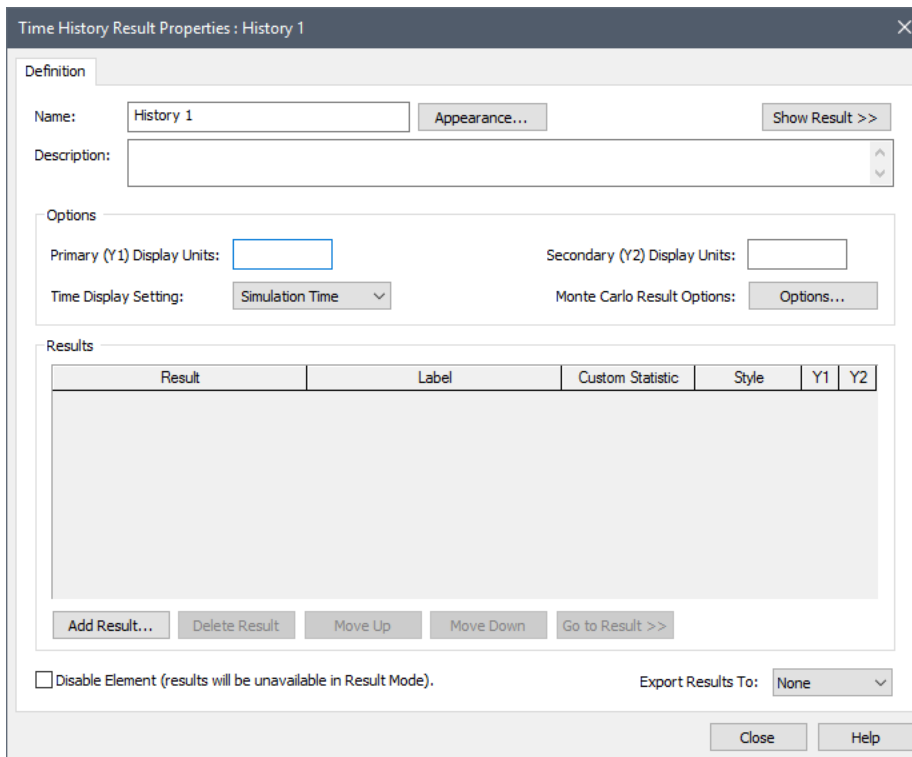
In order to discern the range of possible outcomes, given the salary and spending habits described in the previous steps, we will run multiple realizations of the system using Monte Carlo simulation.

For this simple simulation, 100 realizations will provide a good sample to work with. To change the number of realizations, return to Edit Mode (press **F4**), and open the Simulation Settings dialog (press **F2**). Click on the **Monte Carlo** tab, and change the "# Realizations" to 100:



Close the dialog.

In order to view time histories involving multiple realizations, we need to create a Time History Result element. A Time History Result is inserted by right-clicking in the graphics pane, and selecting **Insert Element|Results|Time History Result** from the context menu:



Press the **Add Result...** button and select Bank\_Account. It will then appear in the list like this:

## A Simple Financial Module Example

Result	Label	Custom Statistic	Style	Y1	Y2
Bank_Account	Bank_Account	Mean (default)		<input checked="" type="checkbox"/>	<input type="checkbox"/>

We can use the Time History Result element to customize the appearance of our plot. We will make just two small changes. Change the name of the element to "Time History Result" (result element names can have spaces) and change the Label for the Result to "Account Value":

Time History Result Properties : Time History Result

Definition

Name:

Description:

Options

Primary (Y1) Display Units:  Secondary (Y2) Display Units:

Time Display Setting:  Monte Carlo Result Options:

Results

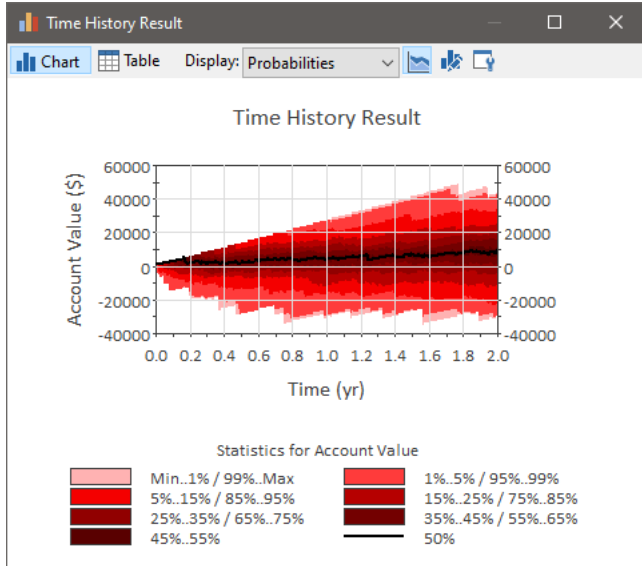
Result	Label	Custom Statistic	Style	Y1	Y2
Bank_Account	Account Value	Mean (default)		<input checked="" type="checkbox"/>	<input type="checkbox"/>

Disable Element (results will be unavailable in Result Mode). Export Results To:

The Name will become the chart header and the Label will be used to label the result on axes and legends.

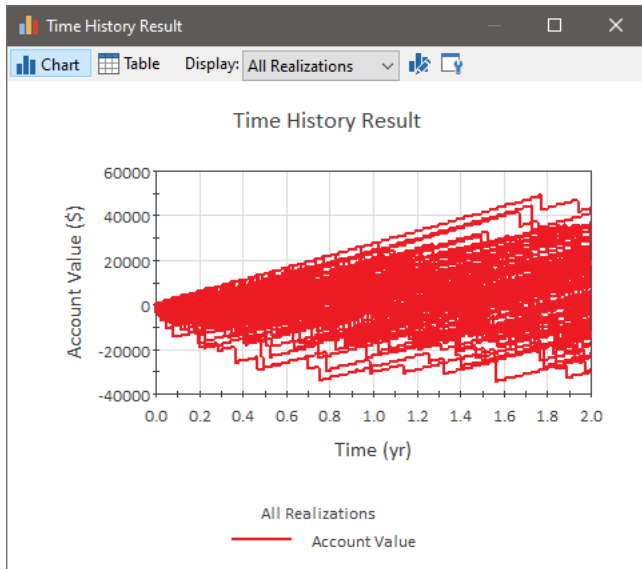
After closing this dialog you can run the model (by pressing **F5**).

After running our simple Monte Carlo simulation, we can view the results in a variety of ways. Let's start by viewing a time history plot. We can do so by double-clicking on the Time History Result element:



When you initially view a time history plot of multiple realizations, GoldSim displays a "Probabilities" view, showing you the median (black line) and percentiles of the output's value over time.

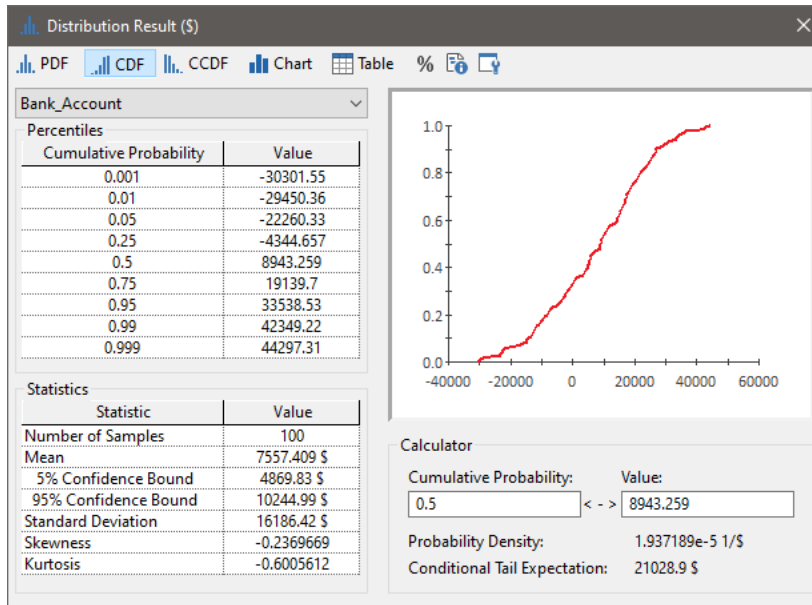
We can select "All Realizations" from the **Display** list at the top of the chart to display all 100 realizations:



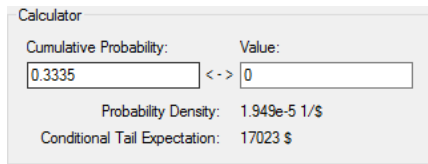
Each line represents a different realization (different, but equally likely, possible future). As can be seen, there is a wide range of possible outcomes, from ending up with a negative balance of nearly \$40,000 to a positive balance of nearly \$50,000.

One question we might want to answer is: "Given my salary and spending habits, what is the probability of having a negative balance after 2 years?". We can easily answer this question by plotting a distribution of the value of the Bank\_Account at the end of the simulation (i.e., after 2 years).

We can do so by right-clicking on the Bank\_Account element and then left-clicking on **Distribution Result...** in the context menu. This will display the following:



This shows the mean, standard deviation, percentiles and other statistics. In the Calculator section, go to the Value field and enter 0:



This displays the probability of the result being less than the specified Value. In this case, it indicates that there is about a 33% chance that the account will have a negative balance after 2 years.

---

**Note:** Your result will be different due to different random number seeds.

---

The file (named Fund.gsm) that we just built can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu).

## Where Do I Go From Here?

This chapter was intended to provide an introduction to the features and capabilities of the Financial Module.

The next two chapters provide detailed explanations of the features introduced here. The final chapter (Chapter 5) discusses a number of example models that illustrate how to represent a variety of systems. These example models are automatically installed with GoldSim.

**Read More:** *Chapter 5: Example Financial Module Applications* on page 81

You may want to start by jumping to the final chapter and experimenting with some of the example models. You will then likely want to refer back to the other chapters describing the details of the Financial Module to fully understand how the examples were constructed.



---

# Chapter 3: Details of the Financial Module Elements

If you have to forecast, forecast often.

Edgar R. Fiedler, *The Three Rs of Economic Forecasting - Irrational, Irrelevant and Irreverent*

## Chapter Overview

This chapter provides the details on the features, capabilities and use of the elements in the Financial Module.

### In This Chapter

This chapter discusses the following topics:

- The Fund Element
- The Cash Flow Element
- The Investment Element
- The Option Element
- The Insurance Element
- Managing Currencies and Exchange Rates

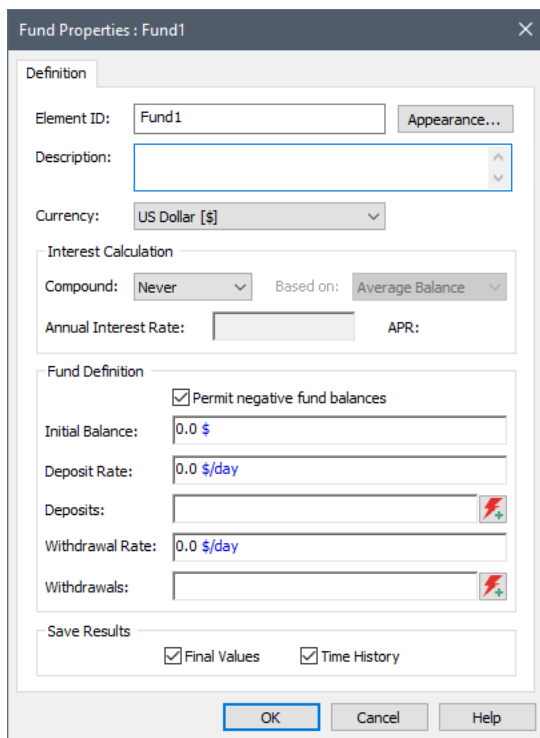
## The Fund Element

The Fund element simulates funds and accounts with a specified initial balance, deposits, withdrawals, and interest rate. This element is used to model bank accounts and loans and other financial components that are governed by an interest rate (e.g., loans). You specify deposits and withdrawals, and the element outputs the current value of the Fund.

The default icon for the Fund element looks like this:



The property dialog looks like this:



Like all GoldSim elements, you first specify an **Element ID** and a **Description**.

Below the Description field, you specify the **Currency** units you wish to use. GoldSim will list all of the currencies that are in its units database (by default, this is dollars, euros, pounds sterling and yen, and you can easily add any additional currencies).

**Read More:** *Managing Currencies and Exchange Rates* on page 70

The details of the Fund element are discussed in the topics below. An example file which illustrates the use of Fund elements (Fund.gsm) can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu).

**Read More:** *Example: Using the Fund Element* on page 82

## Defining How Interest is Computed for a Fund Element

The accounts and funds that you simulate using a Fund element generally will accumulate (or charge) interest that is compounded in a particular manner. The "Interest Calculation" portion of the dialog is used to specify this information:

Interest Calculation

Compound: Quarterly Based on: Minimum Balance

Annual Interest Rate: 5% APR: 5.09453%

Several options are available in the **Compound** drop-list to determine how interest is compounded. In all cases (except when "None" is specified), you must define an **Annual Interest Rate**. GoldSim displays the expected value of the APR (Annual Percentage Rate) to the right of the interest rate.

The actual interest rate that GoldSim uses for the calculation, as well as how the APR is calculated is summarized in the table below (where "Rate" represents the nominal interest rate specified in the **Annual Interest Rate** field). The Annual Interest Rate can be entered as a fraction or a percentage (with the % symbol).

Note that by definition, for all compounding methods except annual, the APR is actually higher than the Annual Interest Rate.

Compounding Option	Interest Rate Used	APR
Continuous	Rate	$e^{\text{Rate}} - 1$
Monthly	Rate / 12 each month	$\left(1 + \frac{\text{Rate}}{12}\right)^{12} - 1$
Quarterly	Rate / 4 each quarter	$\left(1 + \frac{\text{Rate}}{4}\right)^4 - 1$
Annual	Rate each year	Rate

**Note:** Daily compounding can be simulated by selecting Continuous, and specifying an **Annual Interest Rate** (Rate) as follows:

$$\text{Rate} = \text{pc2cc}(\text{DailyRate}, 365)$$

where DailyRate is the nominal annual interest rate (assuming daily compounding) you are trying to compute. The function pc2cc is a built-in financial function that converts a periodic rate to an equivalent continuous rate.

**Read More:** *Financial Functions: Continuous/Periodic Compounding Conversions* on page 28

You must specify what the Interest is based on in the **Based on** field. There are three options: "Minimum Balance", "Average Balance", or "Maximum Balance". This field is grayed out if Continuous compounding is selected, as in this case interest is always based on the current value.

For monthly, quarterly and annual compounding, the interest is applied as a lump sum at the end of each compounding period. For continuous compounding, the interest is applied at every model timestep:

$$F(t) = \frac{[d + \lambda F(t - \Delta t)] e^{\lambda \Delta t} - d}{\lambda}$$

where:

F(t) is the value of the fund at time t;

$\Delta t$  is the timestep length;

F(t- $\Delta t$ ) is the value of the fund at time t - $\Delta t$  (i.e., the previous timestep);

$\lambda$  is the specified nominal annual interest rate (at time t); and

d is the net amount deposited over the timestep.

The following should be noted regarding how interest is calculated in a Fund element:

- For monthly, quarterly, and annual compounding, GoldSim automatically inserts a new timestep at the appropriate time in order to implement the compounding. For elapsed time simulations, GoldSim inserts a timestep once every 365.25/12 days for monthly compounding, once every 365.25/4 days for quarterly compounding, and once every 365.25 days for annual compounding. For date-time simulations, a timestep is added every month, every 3 months or every year (always on the same day of the month as the simulation started on).

---

**Warning:** Inserting the necessary timesteps for monthly or quarterly compounding adds some overhead to the model. If you don't need precision in the timing of interest payments, you can use the "continuous compounding" option and convert your quarterly or monthly rate to an equivalent continuous rate using the `pc2cc` function. For example, Monthly compounding can be simulated by selecting Continuous, and specifying an **Annual Interest Rate** (Rate) as follows:

Rate = `pc2cc(MonthlyRate, 12)`

where `MonthlyRate` is the nominal annual interest rate (assuming monthly compounding) you are trying to compute. The function `pc2cc` is a built-in financial function that converts a periodic rate to an equivalent continuous rate.

- 
- Discrete deposits or withdrawals received at the end of a compounding period are not taken into account for the purposes of calculating the minimum, maximum or average balance for the period.
  - If the Fund value is negative, the interest rate is still applied, but in the negative direction.

## Specifying Initial Balance, Deposits and Withdrawals for a Fund Element


The "Fund Definition" portion of the Fund dialog is used to specify the **Initial Balance**, **Deposits** and **Withdrawals**:

Fund Definition


Permit negative fund balances

Initial Balance: 0.0 \$

Deposit Rate: 0.0 \$/day

Deposits: 

Withdrawal Rate: 0.0 \$/day

Withdrawals: 

Deposits and withdrawals can be specified as continuous rates and/or as discrete changes.

The **Deposit Rate** and **Withdrawal Rate** fields accept constants or links (with dimensions of currency per unit time). These are treated as continuous rates. Hence it would be most appropriate to use these fields only if you had deposits and/or withdrawals that could be approximated as being continuous (e.g., a daily withdrawal). **Deposit Rate** and **Withdrawal Rate** must be non-negative.

The **Deposits** and **Withdrawals** fields accept discrete change signals (typically generated by a Discrete Change element). These fields are used to simulate transactions that must be treated in a discrete (as opposed to continuous) manner. The discrete change signals must have dimensions of currency. Discrete change signals can be positive or negative. A negative Deposit is treated as a Withdrawal, and a negative Withdrawal is treated as a Deposit.

Withdrawals only accept discrete change signals with Add instructions. Deposits accept discrete change signals with Add instructions or Replace instructions. When a Fund element receives a Discrete Change with a Replace instruction, the Fund value is reset to the Value specified by the Discrete Change.

**Read More:** *Generating Discrete Change Signals in Financial Models* on page 18


The buttons to the right of the **Deposits** and **Withdrawal** fields are used to enter multiple discrete changes. These buttons provide access to the following dialog:

Edit multiple deposits

Item1
Item2
Item3

Add... Delete... Close Help

Note that you can also enter multiple discrete change signals into this field by separating them by semi-colons:

Deposits:  

**Note:** If a Fund is inside an inactive conditional Container, all discrete and continuous deposits and withdrawals are ignored. If the deposits or withdrawals originate from an active Container, an warning message is written to the run log.

The **Permit negative fund balances** checkbox controls whether or not the Fund is allowed to have a negative balance. By default, this checkbox is checked on (negative balances are allowed). If negative fund balance are not

allowed (the checkbox is cleared), GoldSim automatically adds two additional outputs to the element to account for situations in which you attempt to withdrawal more money than is available.

**Read More:** *Outputs of the Fund Element* on page 54

---

**Note:** If negative balances are not allowed, GoldSim checks for a negative balance when discrete withdrawals are requested and, if there is a specified withdrawal rate, during every timestep. If a zero balance is encountered partway through a timestep (and interest is continuous), interest is calculated exactly up to the point until the Fund goes to zero.

---

## Outputs of the Fund Element

The key outputs of the Fund element are the fund balance (which has the same name as the element) and the cumulative interest paid (Cum\_Interest). Both of these are scalar values with dimensions of currency.

In addition, if negative balances are not allowed, the element has two additional outputs:

**Actual\_Withdrawal\_Rate.** If negative balances are not allowed and you specify a continuous withdrawal rate input, the actual withdrawal rate could be less than the requested withdrawal rate (the input). This could occur if the Fund value goes to zero. Hence, the Actual\_Withdrawal\_Rate is always less than or equal to the requested withdrawal rate. The Actual\_Withdrawal\_Rate output is a scalar value with dimensions of currency per time.

---

**Note:** If negative balances are not allowed, and you specify a withdrawal rate such that the Fund hits zero in the middle of a timestep, the Actual\_Withdrawal\_Rate output reports the withdrawal rate for that timestep as the average withdrawal rate over the timestep (rather than the withdrawal rate at the end of the timestep, which would be zero)

---

**Actual\_Withdrawals.** If negative balances are not allowed and you specify a discrete withdrawal rate input, GoldSim immediately outputs the actual withdrawal. The Actual\_Withdrawals output is a discrete change signal with an Add instruction (identical to the type of output produced by a Discrete Change element). The value is scalar with dimensions of currency. The value of the Actual\_Withdrawals signal could be less than the requested discrete withdrawal (the input). This could occur if the Fund value goes to zero. Hence, the Actual\_Withdrawals output value is always less than or equal to the requested discrete withdrawal.

---

**Read More:** *Generating Discrete Change Signals in Financial Models* on page 18

## The Cash Flow Element

The Cash Flow element computes the net cash flow, the net present value (NPV) and internal rate of return (IRR) of a cash flow history. This element is used to model the future return of projects, business ventures, and similar undertakings.

The default icon for the Cash Flow element looks like this:



The property dialog looks like this:

---

Like all GoldSim elements, you first specify an **Element ID** and a **Description**.

Below the Description field, you specify the **Currency** units you wish to use. GoldSim will list all of the currencies that are in its units database (by default, this is dollars, euros, pounds sterling and yen, and you can easily add any additional currencies).

**Read More:** *Managing Currencies and Exchange Rates* on page 70

The details of the Cash Flow element are discussed in the topics below. An example file which illustrates the use of Cash Flow elements (Cashflow.gsm) can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu).

**Read More:** *Example: Using the Cash Flow Element* on page 84

## Specifying Expenses and Revenues for a Cash Flow Element

The key inputs to a Cash Flow element are the expenses and revenues that define the cash flow history.

For example, if you were simulating a business venture that required an upfront start-up cost of \$1,000,000, operating costs of \$10,000/month, and generated returns at a rate of \$70,000/month, you would enter the start-up cost and the operating cost as expenses, and the returns as revenues.

The "Cash Flows" portion of the Cash Flow dialog is used to specify the **Expenses** and **Revenues**:

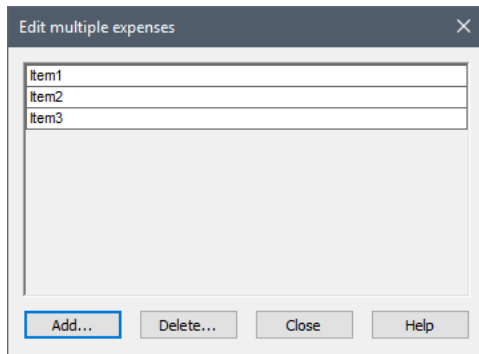
Expenses and revenues can be specified as continuous rates and/or as discrete changes.

The **Expense Rate** and **Revenue Rate** fields accept constants or links (with dimensions of currency per unit time). These are treated as continuous rates. Hence it would be most appropriate to use these fields only if you had expenses and/or revenues that could be approximated as being continuous (e.g., a daily operating cost). **Expense Rate** and **Revenue Rate** must be non-negative.

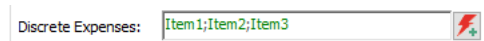
The **Discrete Expenses** and **Discrete Revenues** fields accept discrete change signals (typically generated by a Discrete Change element). These fields are used to simulate transactions that must be treated in a discrete (as opposed to continuous) manner. The discrete change signals must have dimensions of currency and Add instructions. The discrete change signals can be positive or negative. A negative Discrete Expense is treated as a Discrete Revenue, and a negative Discrete Revenue is treated as a Discrete Expense.

**Read More:** *Generating Discrete Change Signals in Financial Models* on page 18

The buttons to the right of the **Discrete Expenses** and **Discrete Revenues** fields are used to enter multiple discrete changes. These buttons provide access to the following dialog:



Note that you can also enter multiple discrete change signals into this field by separating them by semi-colons:



Note that when specifying cash flows into the future, it will often be appropriate to adjust the values for inflation. The manner in which this can be done is discussed below.

### ***Adjusting Cash Flow Expenses and Revenues for Inflation***

When specifying cash flows into the future, it will often be appropriate to adjust the values for inflation. This can be done in a straightforward way by using GoldSim's built-in financial compounding functions.

**Read More:** *Financial Functions: Simple Compounding* on page 28

In particular, the "ptof" function converts a present value to a future value given a rate per period and the number of periods. For example, if the current cost for a particular service was 1000 \$/yr, and you assumed an inflation rate of 3%/yr, then the inflated cost at any time in the future could be computed as:

$$1000 \text{ \$/yr} * \text{ptof}(3\%, \text{ETime|yr})$$

The output of this expression is the inflated cost of 1000 \$/yr at Etime years into the future.

This would be entered into the Cash flow element as follows:



Cash Flows

Expense Rate:

If you wanted to apply inflation to a discrete expense or revenue, you would need to use the "ptof" function in the Value field of the Discrete Change element.

**Read More:** *Generating Discrete Change Signals in Financial Models* on page 18

## Outputs of the Cash Flow Element

The default output of the Cash Flow element (which has the same name as the element) is the cumulative net cash flow. This is the accumulated difference between all revenues and all expenses (such that a positive value indicates that revenues exceed expenses). This output is a scalar value with dimensions of currency.

GoldSim also provides two additional optional outputs in the "Output Definition" portion of the Cash Flow dialog:

Output Definition

Calculate Net Present Value of Cash Flows

Annual Discount Rate:

Calculate Internal Rate of Return

Compounding Method:

If **Calculated Net Present Value of Cash Flows** is checked, GoldSim adds an output called NPV. This is a scalar value with dimensions of currency. If this box is checked, the **Annual Discount Rate** field is activated (and must be entered as a percentage).

If **Calculate Internal Rate of Return** is checked, GoldSim adds an output called IRR. This is a scalar, dimensionless value representing the annual return rate.

The **Compounding Method** applies to how the NPV and IRR are calculated.

These two optional outputs are discussed in detail in the sections below.

## Computing the NPV of a Cash Flow History

If **Calculate Net Present Value of Cash Flows** is checked in the "Output Definition" portion of the Cash Flow dialog, GoldSim adds an output called NPV. This is a scalar value with dimensions of currency.

The net present value calculation (NPV) recognizes the time value of money, and discounts future cashflows (both expenses and revenues) based on a specified discount rate. Hence, an expense incurred 5 years in the future has a present value that is less than the same expense incurred today. Likewise, a revenue generated 5 years in the future is worth less than the same revenue incurred today.

If the **Calculate Net Present Value of Cash Flows** is checked, the **Annual Discount Rate** field is activated. The Annual Discount Rate can be entered as a fraction or a percentage (with the % symbol).

The net present value of an expense or revenue generated at some time  $t$  into the future is a function of the **Compounding Method** (Continuous or Annual). The default is Annual.

Generally, the discount rate will be entered as a constant (with riskier projects being assigned a higher discount rate). However, you can also specify the discount rate to be time variable. In this case, GoldSim treats it as a "zero coupon rate", such that the future cash flows are discounted using the value of the discount rate at the time the cash flow is incurred.

## Computing the IRR of a Cash Flow History

If **Calculate Internal Rate of Return** is checked in the "Output Definition" portion of the Cash Flow dialog, GoldSim adds an output called IRR. This is a dimensionless scalar value (representing the annual return rate).

The internal rate of return (IRR) is the discount rate that would yield an NPV for the cash flows equal to zero. That is, it is the discount rate that makes the present value of returns equal to the present value of expenses (investments). As such, the IRR is a measure of the profitability of the investments.

Like the NPV, the IRR calculation is a function of the **Compounding Method** (Continuous or Annual). The default is Annual.

Calculation of the IRR requires an iterative solution. Convergence to a unique solution may not initially (or ever) be possible. In general, the IRR exists and is unique if one or more years of net investment (negative cash flow) are followed by years of net revenues. This has several important implications for interpreting this output:

- Prior to the existence of a unique IRR or until the first valid result (between -0.9999 and 99.999), GoldSim outputs an IRR of zero.
- After computing a valid result, if the annual IRR is less than -0.9999 (-99.99%), it is treated as "unsolvable", and GoldSim outputs an IRR value of -99.999.
- After computing a valid result, if the annual IRR is greater than 99.999 (9999.9%), it is treated as "unsolvable", and GoldSim outputs an IRR value of 99.999.

Because IRR calculations can be time-consuming, GoldSim simplifies the calculation in several ways:

- Continuous inflow and outflows are treated as a single lump sum applied at the mid-point of each timestep.
- All flows that occur within 2.5% of the current time of each other are lumped together into a single item at the weighted average time of the transactions.
- The IRR result is updated at a maximum every 1/200th of the simulation duration (as well as at the first and last time points). For histories with a large number of timesteps, this can produce a slightly stair-stepped time history plot.

## The Investment Element

The Investment element simulates the growth of an investment or group of investments such as a security or portfolio of securities. You input purchases, sales and the unit value of the investment, and the element outputs the investments current value.

The default icon for the Investment element looks like this:



The property dialog looks like this:

Like all GoldSim elements, you first specify an **Element ID** and a **Description**.

Below the Description field, you specify the **Currency** units you wish to use. GoldSim will list all of the currencies that are in its units database (by default, this is dollars, euros, pounds sterling and yen, and you can easily add any additional currencies).

**Read More:** *Managing Currencies and Exchange Rates* on page 70

To the right of the Currency, you can specify the **Type**. By default, an Investment provides a scalar output (and requires scalar inputs). However, via this button, you can also specify it to have a vector (1-D array) output (and require vector inputs).

The details of the Investment element are discussed in the topics below. An example file which illustrates the use of Investment elements (Investment.gsm) can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu).

**Read More:** *Example: Using the Investment Element* on page 86

## Specifying the Growth of the Investment's Underlying Unit Value

An Investment element tracks the value of an investment. To do so, it keeps track of specified purchases and sales. However, the key input to an Investment element is a description of the growth of the investment's underlying unit value (e.g., the price of a stock).

The "Growth Definition" portion of the Investment dialog is used to specify the **Unit Value** of the investment:

For example, if the investment represented ownership of shares of a stock, this would be the share price.

This value must be a scalar with dimensions of currency. The **Unit Value** must always be a positive number.

Typically, the Unit Value will be a time series function. This can be any function, but will often be the output of a History Generator element. This element (which is part of the basic GoldSim framework), provides a variety of alternatives for generating time histories, including simply specifying a constant annual growth rate, and specifying a stochastic growth rate (defined using a mean annual growth rate and volatility, and simulated as a Wiener process).

The History Generator also supports specification of correlation matrices, which allows you to specify the correlations amongst a portfolio of investments.

---

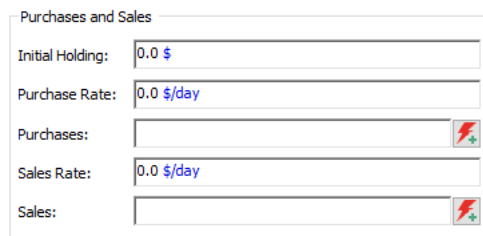
**Note:** The History Generator element is discussed in detail in Chapter 10 of the [GoldSim User's Guide](#).

---

## Specifying Initial Holding, Purchases and Sales for an Investment Element

After specifying the growth of the Investment's underlying unit value (e.g., the share price of a stock), you must then specify your Initial Holding in the Investment, as well as any Purchases and Sales.

The "Purchases and Sales" portion of the Investment dialog is used to specify this information:



Note that these inputs are entered in terms of the value (in currency), as opposed to the number of units (e.g., number of shares). These inputs must all have the same order as the output (specified via the **Type** button). By default, they are scalar, but they can also be specified as vectors (e.g., to simulate a portfolio of investments).

The **Initial Holding** has units of currency and represents the value of the Investment at the beginning of the simulation. This must be greater than or equal to zero.

Purchases and sales can be specified as continuous rates and/or as discrete changes.

The **Purchase Rate** and **Sales Rate** fields accept constants or links (with dimensions of currency per unit time). These are treated as continuous rates. Hence it would be most appropriate to use these fields only if you had purchases and/or sales that could be approximated as being continuous (e.g., a daily purchase or sale). **Purchase Rate** and **Sales Rate** must be non-negative.

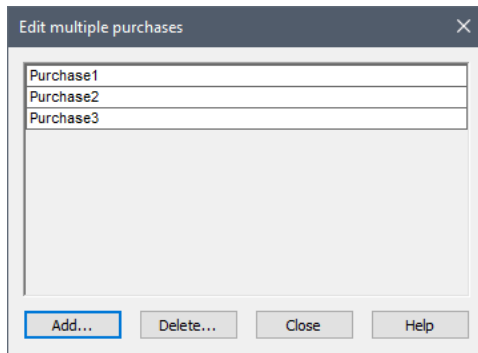
The **Purchases** and **Sales** fields accept discrete change signals (typically generated by a Discrete Change element). These fields are used to simulate transactions that must be treated in a discrete (as opposed to continuous) manner. The discrete change signals must have dimensions of currency.

Discrete change signals can be positive or negative. A negative Purchase is treated as a Sale, and a negative Sale is treated as a Purchase.

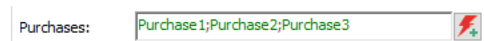
**Sales** only accept discrete change signals with Add instructions. **Purchases** accept discrete change signals with Add instructions or Replace instructions. When an Investment element receives a Discrete Change with a Replace instruction, the Investment value is reset to the Value specified by the Discrete Change.

**Read More:** *Generating Discrete Change Signals in Financial Models* on page 18

The buttons to the right of the **Purchases** and **Sales** fields are used to enter multiple discrete changes. These buttons provide access to the following dialog:



Note that you can also enter multiple discrete change signals into this field by separating them by semi-colons:



**Note:** If an Investment is inside an inactive conditional Container, all discrete and continuous purchases and sales are ignored. If the purchases or sales originate from an active Container, a warning message is written to the run log.

**Note** Regular reinvestment of dividends would be simulated by adding these reinvestments as discrete purchases.

## Output of the Investment Element

The one output of the Investment element (which has the same name as the element) is the current value of the Investment. This output is a scalar value with dimensions of currency.

Note that if you wanted to compute the number of units of the Investment at any given time, you would simply divide the Investment output by the Unit Value.

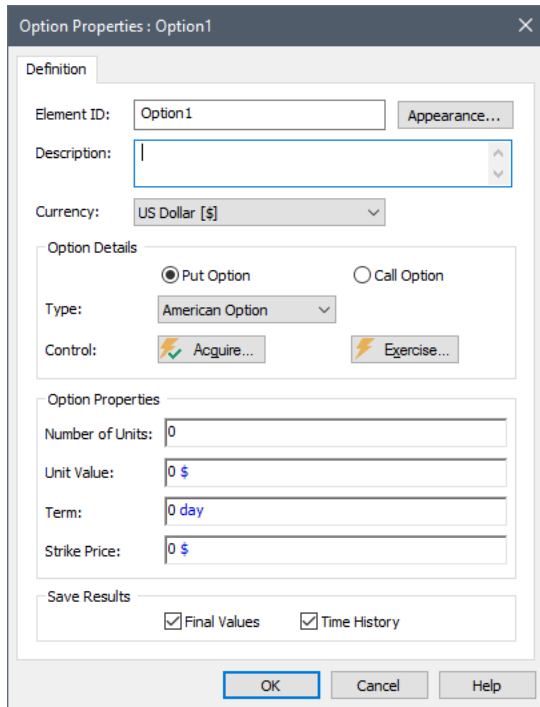
## The Option Element

The Option element simulates the acquisition and exercise of financial options (puts and calls). You specify the type of option, option properties (e.g., number of units, strike price, term), the unit value of the underlying stock, and when the option is acquired and exercised, and the element outputs the value realized upon exercise of the option.

The default icon for the Investment element looks like this:



The property dialog looks like this:



Like all GoldSim elements, you first specify an **Element ID** and a **Description**.

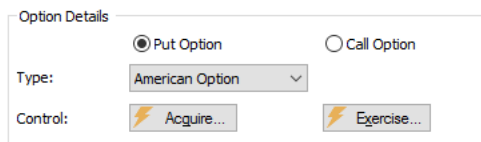
Below the Description field, you specify the **Currency** units you wish to use. GoldSim will list all of the currencies that are in its units database (by default, this is dollars, euros, pounds sterling and yen, and you can easily add any additional currencies).

**Read More:** *Managing Currencies and Exchange Rates* on page 70

The details of the Option element are discussed in the topics below. An example file which illustrates the use of Option elements (Option.gsm) can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu).

## Specifying the Type of Option

The first step in defining an Option element is to define the type of option you wish to simulate. The "Option Details" portion of the dialog is used to specify this information:



A pair of radio buttons are used to specify whether you wish to simulate a **Put Option** or a **Call Option**. A put option gives the buyer the right to *sell* stock at a specified price (the exercise price) within a given period of time. A call option gives the buyer the right to *buy* stock at a specified price (the exercise price) within a given period of time.

GoldSim allows you to specify (via the **Type** drop-list) one of three types of options:

- An "American Option" may be exercised at any time prior to the expiration date.
- A "European Option" may be exercised only at the expiration date.
- An "Asian Option" may be exercised only at the expiration date, and is valued based on the average price of the underlying stock over a given period of time. In GoldSim, this is always the time since the option was acquired.

## Acquiring Options

During a simulation, you can acquire an option by specifying an Acquisition trigger. When you acquire an option, you must specify the option properties (number of units acquired, current unit value, term and strike price).

The **Acquire...** trigger button is used to trigger the acquisition. Pressing this button brings up the standard GoldSim triggering dialog:



This dialog allows you to specify conditions or events that will result in an acquisition to be triggered.

**Read More:** *Defining Triggers* on page 16

The "Option Properties" portion of the dialog is used to specify the properties of the option being acquired:

Option Properties	
Number of Units:	0.0
Unit Value:	0.0 \$
Term:	0.0 day
Strike Price:	0.0 \$

The **Number of Units** must be a positive value. Although it generally will be an integer, it does not have to be. That is, GoldSim will allow you to acquire a option consisting of a fractional number of units.

The **Unit Value** is the current unit price of the underlying security. After an option is acquired, GoldSim computes the value of the option at any given time based on the current Unit Value. This value must be a scalar with dimensions of currency. The **Unit Value** must always be a positive number.

**Read More:** *Specifying the Growth of the Option's Underlying Unit Value* on page 64

The **Term** is the duration of the option. The Option expires when the simulation time is greater than the acquisition time plus the term. Depending on the type of Option, it can be exercised prior to or at the expiration time of the Option. The **Term** must be a positive value with dimensions of time.

**Note:** GoldSim automatically interrupts the clock (i.e., adds a timestep) at the expiration time of an Option. This ensures that the Options exercised at their expiration (European and Asian options) are accurately simulated regardless of the model timestep.

---

The **Strike Price** is the price that the stock can be bought or sold for when the Option is exercised. It must be a positive value with dimensions of currency.

**Read More:** *Valuing and Exercising Options* on page 64

The **Number of Units**, **Term** and **Strike Price** may be functions of time, but their value is only used at the time the Option is acquired. An Option element can only "hold" one option at a time. That is, once the Acquire trigger has been triggered, it cannot be triggered again until the Option has been exercised or has expired.

## Specifying the Growth of the Option's Underlying Unit Value

In order to value an option (and determine what it is worth at the time it is exercised), it is necessary to simulate and track the Option's underlying unit value (i.e., the price of a stock).

The Unit Value is specified in the "Option Properties" portion of the dialog:

Option Properties	
Number of Units:	0.0
Unit Value:	0.0 \$
Term:	0.0 day
Strike Price:	0.0 \$

The **Unit Value** must always be a positive number with dimensions of currency.

Typically, the Unit Value will be a time series function. This can be any function, but will often be the output of a History Generator element. This element (which is part of the basic GoldSim framework), provides a variety of alternatives for generating time histories, including simply specifying a constant annual growth rate, and specifying a stochastic growth rate (defined using a mean annual growth rate and volatility, and simulated as a Wiener process).

---

**Note:** The History Generator element is discussed in detail in Chapter 10 of the [GoldSim User's Guide](#).

---

## Valuing and Exercising Options

Once an option is acquired, its value is computed by GoldSim as follows:

- For Put options:  
Option Value = (Strike Price - Unit Value)\*Number of Units
- For Call options:  
Option Value = (Unit Value - Strike Price)\*Number of Units

In both cases, if the Option Value evaluates to a negative number, it is set to zero. Options with positive Option Values are said to be *in the money*. Hence, Put options are in the money if the Strike Price is greater than the current Unit Price. Call options are in the money if the Strike Price is less than the current Unit Price.

---



**Note:** When valuing an option, GoldSim does not account for the price that you paid to purchase the option. It only accounts for the value of the option at the time it is exercised.

It is important to note that when computing the Option Value, GoldSim uses the values for the **Strike Price** and the **Number of Units** that existed at the time the Option was acquired, while it uses the **Unit Value** that exists at the current time (i.e., whenever the Option Value is computed).

After you acquire an Option, it can be exercised in one of two ways:

- For all option types, the Option is automatically exercised at its time of expiration if it is *in the money*.
- For American options, the Option can be exercised any time after acquisition via the **Exercise...** button. Pressing this button brings up the standard GoldSim triggering dialog:

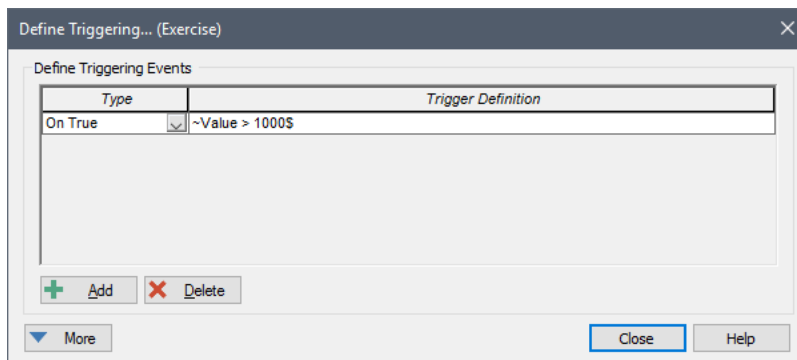


This dialog allows you to specify conditions or events that will result in an exercise to be triggered.

**Read More:** *Defining Triggers* on page 16

**Warning:** If you exercise an American option (via the **Exercise...** trigger) that is not in the money (has a value of zero), GoldSim will immediately cancel the Option.

Under some circumstances, you may want to define the trigger in terms of the current value of the options (e.g., "exercise the option if the value exceeds X") GoldSim facilitates this by providing a locally available property called "Value". This represents the current value of the option (as defined above). You would reference this value as "~Value":



**Note:** Value is an example of a locally available property. As such, it only has meaning inside the **Trigger Definition** field, and cannot be referenced anywhere else. Locally available properties are discussed in Chapter 10 of the [GoldSim User's Guide](#).

---

## Outputs of the Option Element

The primary output of the Option element (which has the same name as the element) is the current Option Value. Prior to acquisition of an Option, and after exercising an Option, this value is displayed as zero. After acquiring an Option, and up until the point it is exercised, the value is computed based on the Option properties (Strike Price, Number of Units) and the current Unit Value.

**Read More:** *Valuing and Exercising Options* on page 64

In addition, the Option element has a secondary output:

**Exercise.** When an Option is exercised, GoldSim outputs a discrete change signal with an Add instruction named Exercise (identical to the type of output produced by a Discrete Change element). The value is scalar with dimensions of currency. The value represents the Option Value at the time the Option was exercised. This discrete change output is only emitted if the option's value is positive.

**Read More:** *Generating Discrete Change Signals in Financial Models* on page 18

## The Insurance Element

The Insurance element simulates claims against an insurance policy. You must specify the properties of the insurance policy (the deductible, the cap, and when these are reset), and then you specify one or more discrete claims. The key outputs are the cumulative covered and uncovered losses for the claims.

The default icon for the Insurance element looks like this:



The property dialog looks like this:

Like all GoldSim elements, you first specify an **Element ID** and a **Description**.

Below the Description field, you specify the **Currency** units you wish to use. GoldSim will list all of the currencies that are in its units database (by default, this is dollars, euros, pounds sterling and yen, and you can easily add any additional currencies).

**Read More:** *Managing Currencies and Exchange Rates* on page 70

The details of the Insurance element are discussed in the topics below. An example file which illustrates the use of Insurance elements (Insurance.gsm) can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu).

**Read More:** *Example: Using the Insurance Element* on page 90

## Specifying the Properties of the Insurance Policy

The "Definition" portion of the Insurance dialog is used to specify the properties of the insurance policy:

The key properties of the policy are the **Deductible** and the **Cap**.

The **Deductible** is the cumulative amount of claims that the insured is responsible for before coverage starts (from the insurer's point of view, the lower loss limit). The **Cap** is the maximum amount payable to the insured from the insurer until the value is reset (from the insurer's point of view, the upper loss limit). Both of these inputs have units of currency. The **Deductible** must be a non-negative values. The **Cap**, however, can be also be negative. If it is negative, the **Cap** is ignored (there is no cap on coverage). If it is zero, all claims are rejected.

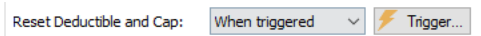
---

**Note:** To simulate an expiring policy, set the **Cap** to zero.

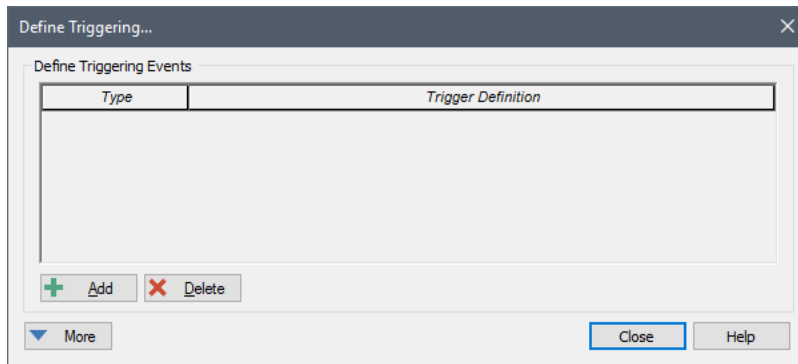
---

The **Reset Deductible and Cap** drop-list determines when these values are reset. There are four options:

- "For each claim". The Deductible and Cap are reset prior to processing a claim. That is, this option is used for policies in which each event has the full deductible and cap applied.
- "Annually". The Deductible and Cap are reset once per year (with the first reset occurring at one year after the Insurance element is activated, which, by default, is the start of the simulation). GoldSim automatically interrupts the clock (i.e., adds a timestep) once per year. This ensures that Caps and Deductibles are reset annually regardless of the model timestep.
- "Never". The Deductible and Cap are never reset.
- "When triggered". The Deductible and Cap are reset by a user-specified trigger. If this option is selected, the Trigger... button to the right of the input field becomes available:



Pressing this button brings up the standard GoldSim triggering dialog:



This dialog allows you to specify conditions or events that will result in Deductible and Cap being reset.

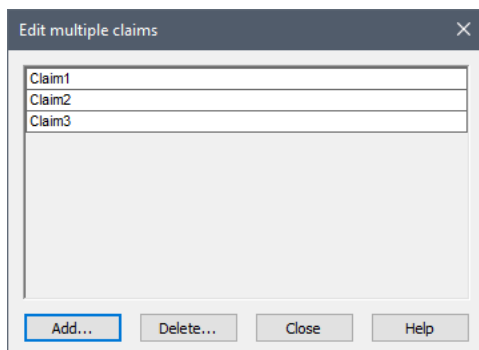
**Read More:** *Defining Triggers* on page 16

The **Deductible** and **Cap** can be specified as functions of time, but are only applied at the start of each realization, and when these limits are reset (as described above).

## Generating Insurance Claims

Insurance claims are simulated as discrete change signals (typically generated by Discrete Change elements). Hence, the **Claims** field accepts only discrete change signals.

The button to the right of the field is used to enter multiple claims:



Note that you can also enter multiple discrete change signals into this field by separating them by semi-colons:



The discrete change signals representing the claims must have dimensions of currency, must have non-negative values and can have either Add or Replace instructions (the two instruction types are treated identically).

**Read More:** *Generating Discrete Change Signals in Financial Models* on page 18

When an Insurance element receives a Claim, it applies the Deductible and Cap, and determines what portion of the claim is covered, and what portion is not covered. The outputs of the Insurance element summarize the current state of the policy (e.g., cumulative amount covered, cumulative uncovered). The element also generates discrete change signals representing the covered and uncovered claim amounts.

---

**Note:** If the Deductible and Cap are scheduled to be reset at the same time at which a claim is received, the claim is always processed before resetting the Deductible and Cap.

---



---

**Note:** If an Insurance element is inside an inactive conditional Container, all claims it receives are ignored.

---

## Outputs of the Insurance Element

The Insurance element has six outputs:

**Covered\_Loss.** This is a discrete change signal that is generated if and only if the deductible is satisfied and the cap is not exceeded. It is identical to the type of output produced by a Discrete Change element. It represents the part of the claim that is covered by the insurer. It is a discrete change signal with an Add instruction, and is a scalar value with dimensions of currency.

**Uncovered\_Loss.** This is a discrete change signal that is generated if and only if some of the claim is not covered (either to satisfy the deductible, or if the cap is exceeded). It is identical to the type of output produced by a Discrete Change element. It represents the part of the claim that is not covered by the insurer. It is a discrete change signal with Add instruction, and is a scalar value with dimensions of currency.

**Read More:** *Generating Discrete Change Signals in Financial Models* on page 18

**Cum\_Covered.** This is the cumulative value of covered claims over the simulation. That is, it is the cumulative amount of claims that is covered by the insurer. It is a scalar value with dimensions of currency.

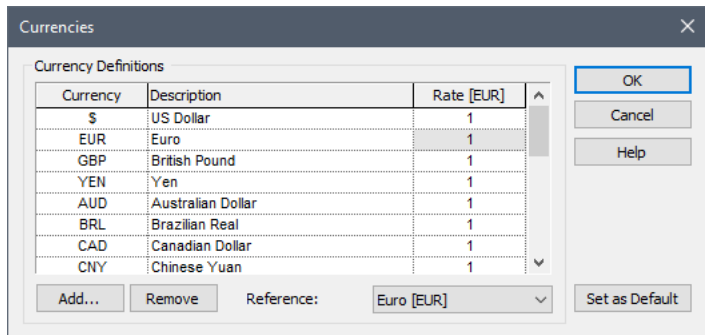
**Cum\_Uncovered.** This is the cumulative value of uncovered claims over the simulation. That is, it is the cumulative amount of claims that is not covered by the insurer (cumulative losses to the insured). It is a scalar value with dimensions of currency.

**Ded\_Balance.** This is the current amount of deductible that is unused. Hence, prior to the first claim (and whenever the deductible is reset), it is equal to the Deductible amount. It is a scalar value with dimensions of currency.

**Cap\_Balance.** This is the current amount of cap that is unused. Hence, prior to the first claim (and whenever the cap is reset), it is equal to the Cap amount. It is a scalar value with dimensions of currency.

## Managing Currencies and Exchange Rates

All of the elements in the Financial Module require that you specify the Currency units to use for the element. GoldSim provides a wide variety of built-in currency units. You can access a dialog for defining conversion rates (and adding new units) by selecting **Model | Currencies...** from the main menu. The following dialog will be displayed:



By default, all currencies are assigned a conversion (exchange) rate of 1 per unit of the **Reference** currency. To change the rate, click into the **Rate** field, and enter a new value (which must be a positive number).

If you press the **Set as Default** button and then press OK, the current exchange rates will be used as defaults for any new models that are created.

The **Reference** currency determines how the **Rate** is defined (rates are defined relative to the Reference currency). In addition, the Reference currency is displayed as the default in new Financial Module elements.

---

**Note:** The exchange rates must be entered as constant values. If you want to model transient exchange rates, you should define the Rate as 1, and model the exchange rate directly in your model as a variable.

---

**Read More:** *Modeling Variable Exchange Rates* on page 71

---

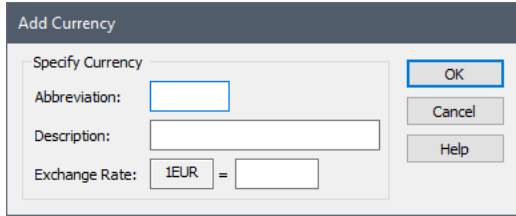
**Note:** Although currency units are listed in the Units Manager dialog (accessed by selecting **Model|Units...** from the main menu), you cannot edit the exchange rates in the Units Manager. This can only be done via the Currencies dialog.

---

**Note:** For all built-in currency units, there are corresponding units for a thousand and a million units. Hence, k\$ is thousands of US Dollars, kEUR is thousands of Euros, M\$ is millions of US Dollars, and MEUR is millions of Euros.

---

Pressing the **Add...** button in the Currencies dialog allows you to create new currencies. When you do so, the following dialog is displayed:



To add a new currency, provide an abbreviation, description and exchange rate.

The **Remove** button can only be used to remove a currency that you have added (you cannot remove any of the built-in currencies).

To make it easier to enter currency units in you models, GoldSim provides four shortcut keys that can be used within a Currency or Units input field:

Shortcut Keys	Unit
Ctrl+d	\$
Ctrl+e	EUR
Ctrl+p	GBP
Ctrl+y	YEN

**Note:** When displaying currencies in results displays, you can control whether or not scientific notation is used (by default, it is not except for values above 1E10). The options for doing so are found on the **Results** tab of the Options dialog (accessed via **Model|Options...** from the main menu).

## Modeling Variable Exchange Rates

Like all other conversion factors, the conversion factors between currencies (i.e., the exchange rates) are treated as constants in GoldSim. Unlike other conversion factors, however, in the real world, exchange rates vary with time.

In many cases, it will be most appropriate for you to assume a constant exchange rate in you simulation models. In some cases, however, the time variable nature of exchange rates may be a critical part of your model, and must be included explicitly.

In order to represent variable exchange rates, you must model the exchange rate directly in your model as a variable. This can be done as follows:

1. Create a model for the exchange rate. The exchange rate will typically be the output of a Time Series element or a History Generator element. The exchange rate is dimensionless (e.g., it will have units of EUR/\$).
2. In the Currencies dialog, define the conversion factor as 1.
3. In your model, compute you answers in one currency (e.g., \$) and then manually convert them to the other currency (e.g., EUR) by multiplying them by the output representing the exchange rate.

A simple example illustrating this procedure (VariableExchange.gsm) can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu).

**Read More:** *Example: Simulating Variable Exchange Rates* on page 91



---

# Chapter 4: Running a Financial Simulation and Viewing Results

It is time that financial types developed a greater tolerance for imprecision, because that's the way the world is.

John C. Burton

## Chapter Overview

This chapter describes how to set up and run a financial model in GoldSim. For the most part, the information presented here is discussed in detail in Chapter 7 of the [GoldSim User's Guide](#)). This chapter summarizes that information, and adds some comments that are specific to running financial models.

## In This Chapter

This chapter discusses:

- Understanding Dynamic Financial Modeling in GoldSim
- Using Monte Carlo Simulation in Your Financial Model
- Saving and Viewing Financial Module Results

## Understanding Dynamic Financial Modeling in GoldSim

Dynamic simulation allows you to develop a representation of a financial system, and then observe that system's evolution and performance over a specified period of time.

The primary advantages of dynamic simulation are:

- The system can evolve into any feasible state and its properties can change suddenly or gradually as the simulation progresses; and
- The system can be affected by random (stochastic) processes and events, which may be either internal (e.g., purchases, sales) or external (interest rate changes).

In order to run a dynamic simulation, you must specify the duration of the simulation (e.g., 1 month, 1 year) and the length of the timesteps that you will use (i.e., the degree to which time will be discretized).

The specified number of timesteps is the minimum number of times that GoldSim will recalculate and update all of the elements in the model. The number of timesteps required to accurately model a system depends to a large extent on how you've built your model.

GoldSim will automatically interrupt the simulation and force it to update (i.e., insert a timestep) under certain circumstances (e.g., when an option expires and is automatically exercised).

**Read More:** *How Discrete Financial Events are Represented in Time* on page 75

---

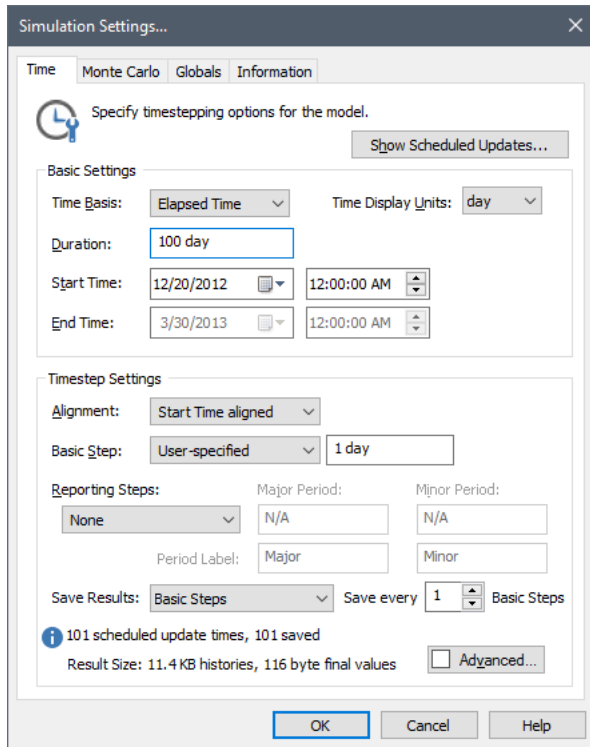
**Note:** Details GoldSim's dynamic timestepping algorithm, including a discussion selecting the proper timestep for a model, are presented in Appendix F of the [GoldSim User's Guide](#).

---

## Setting Up a Dynamic Financial Simulation

The basic time options in GoldSim are defined in the **Time** tab of the Simulation Settings dialog. The Simulation Settings dialog is accessed by pressing **F2** or by selecting **Run | Simulation Settings...** from the main menu.

The **Time** tab of the Simulation Settings dialog is shown below:



The **Duration** of the simulation can be specified as a time period (e.g., 5 years), or in terms of a start and end date in the Basic Time Settings section of the dialog. The number of timesteps (**Basic Step**) is specified in the Timestep Settings portion of the dialog.

GoldSim also provides a variety of advanced timestepping options, which are accessed via the **Advanced...** button in the **Time** tab.

---

**Note:** The timestepping options in GoldSim are discussed in detail in Chapter 7 of the [GoldSim User's Guide](#).

---

## How Discrete Financial Events are Represented in Time

When an event (such as a regular payday) occurs that triggers a deposit into a Fund, when an option expires in an Option element (and triggers the option to be exercised), or when the Deductibles and Cap are reset on an annual basis, the events may not fall exactly on a "scheduled" timestep (i.e., a timestep that was defined in the **Time** tab of the Simulation Settings dialog). That is, events such as these may actually occur between scheduled timesteps.

These trigger an "unscheduled update" of the model. Unscheduled updates are timesteps that are dynamically inserted by GoldSim during the simulation in order to more accurately simulate the system. That is, they are not specified directly prior to running the model. GoldSim inserts them automatically (and, generally, without you needing to be aware of it).

For example, if you had specified a one day timestep, and discrete deposit occurs at 33.65 days (i.e., between the scheduled one-day updates), GoldSim would insert an unscheduled update at 33.65 days.

---

**Note:** The manner in which GoldSim inserts timesteps in response to events is discussed in detail in Chapter 7 of the [GoldSim User's Guide](#).

---

A key and important difference between scheduled updates and unscheduled updates is that scheduled updates are included in time history plots and tables (unless you choose to exclude them). Unscheduled updates, however, do not appear in time history plots and tables. That is, although these timesteps may affect the results (e.g., by making them more accurate at the scheduled timesteps), unscheduled updates of the model are not saved and plotted. Only the scheduled updates are actually saved and plotted.

---

**Note:** In some cases, it may be of interest to see the values of selected outputs that were computed at unscheduled updates. To facilitate this, Time History Result elements provide an option to do so.

---

## Using Monte Carlo Simulation in Your Financial Model

Once a model of a system has been constructed, you can simulate the system to predict how it will evolve through time. By definition, however, the performance of most financial systems is stochastic (i.e., inherently variable), since many inputs (e.g., unit values for investments, interest rates) are generally described stochastically. That is, we can't say exactly how an investment will perform in the future; we can only describe its possible future behavior statistically (e.g., using a mean growth rate and a volatility).

In addition to this inherent variability, we might also be uncertain about some of the input parameters controlling the model. For example, if we had limited cost data regarding a particular component of a project we were simulating, the parameters describing the costs for that component would be uncertain, and we could enter these as probability distributions in order to capture this uncertainty.

Variability and uncertainty are represented in GoldSim using Monte Carlo simulation. Monte Carlo simulation consists of calculating a large number of "realizations" (potential futures). Each realization simulates the same system with the same initial conditions, but with different sampled stochastic values, both at the beginning of the simulation and as the system evolves through time. This results in a large number of separate and independent results, each of which is considered equally likely. These realizations can then be combined to provide statistical information on possible outcomes.

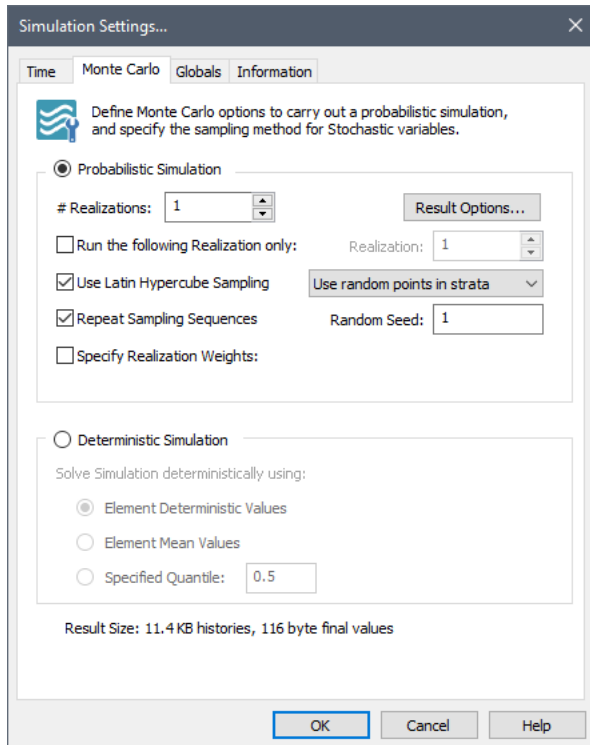
The number of realizations that are required in order to accurately capture the behavior of a system is a complex issue that can be influenced by the computational requirements of running a realization, and the frequency of the behaviors you wish to capture.

A rule of thumb for determining the number of realizations is that the number of realizations should be large enough that at least 10 simulations will have an occurrence of the most infrequent behavior you want to capture. For example, if you wanted to observe two consequences, one which occurred once in every 10 realizations, and another that occurred once in every 500 realizations, an adequate number of realizations for the simulation would be 5000.

## Setting the Monte Carlo Options for a Financial Model

Monte Carlo options in GoldSim are defined in the **Monte Carlo** tab of the Simulation Settings dialog. The Simulation Settings dialog is accessed by pressing F2 or by selecting **Run | Simulation Settings...** from the main menu.

The **Monte Carlo** tab of the Simulation Settings dialog is shown below:

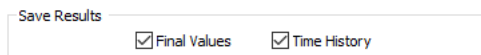


To change the number of realizations, simply edit the number in the **# Realizations** field.

**Note:** The Monte Carlo options in GoldSim are discussed in detail in Chapter 7 of the [GoldSim User's Guide](#).

## Saving and Viewing Financial Module Results

All Financial Module elements have multiple outputs. Like all other elements in GoldSim, Financial element property dialogs have check boxes (at the bottom of the dialog) to specify whether outputs of the element are to be saved. You can save the **Final Values** (the values at the end of each realization in the simulation) and/or **Time History** (the value at selected timesteps throughout the simulation).



By default, when you create a new scalar element, the Save Results checkboxes will be checked (on). For vector elements (Investments can be defined as vectors), the **Time History** checkbox defaults off.

The **Final Values** checkbox always controls whether Final Value results are saved. However, the **Time History** checkbox can be overridden. In particular, it is always applied for single realization runs, but is overridden for multiple realization runs. In particular, Time History results for multiple realization runs are only saved for outputs that are connected to Result elements.

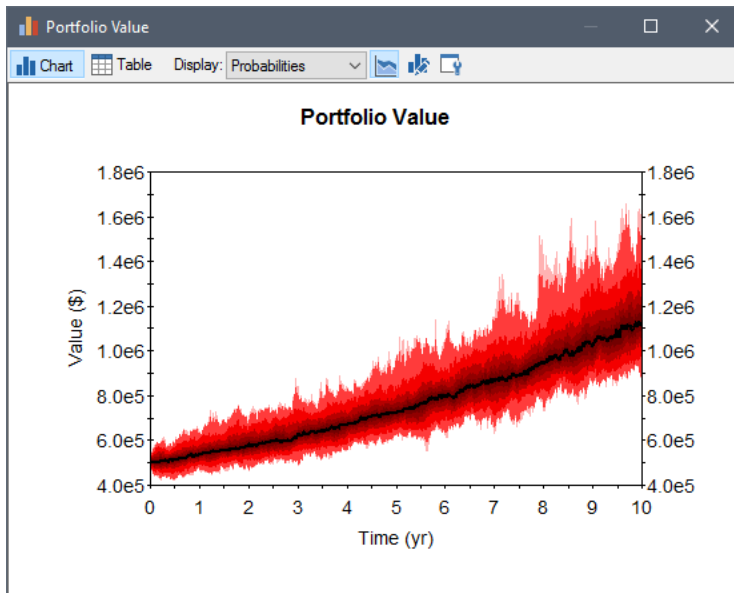
When Time History results are saved, by default GoldSim saves the values of outputs at every Basic Step (and Reporting Period). You can instruct GoldSim to only save results at selected timesteps (e.g., every tenth Basic Step) when defining timestepping options in the Simulation Settings dialog.

## Saving and Viewing Financial Module Results

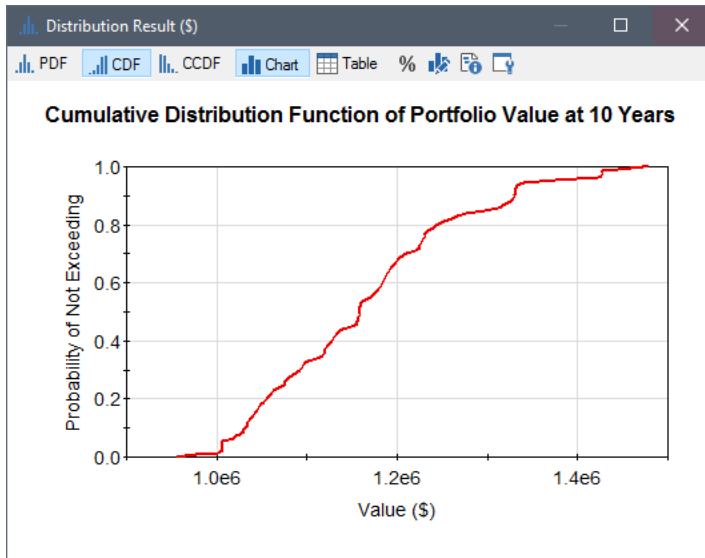
Like all element outputs in GoldSim, after running a model, these outputs can be accessed by right-clicking on the output port of the element.

**Note:** Time history results may not be available in this manner, depending on the nature of the simulation. In particular, for runs with multiple realizations, extra steps must be taken to save and view time history results.

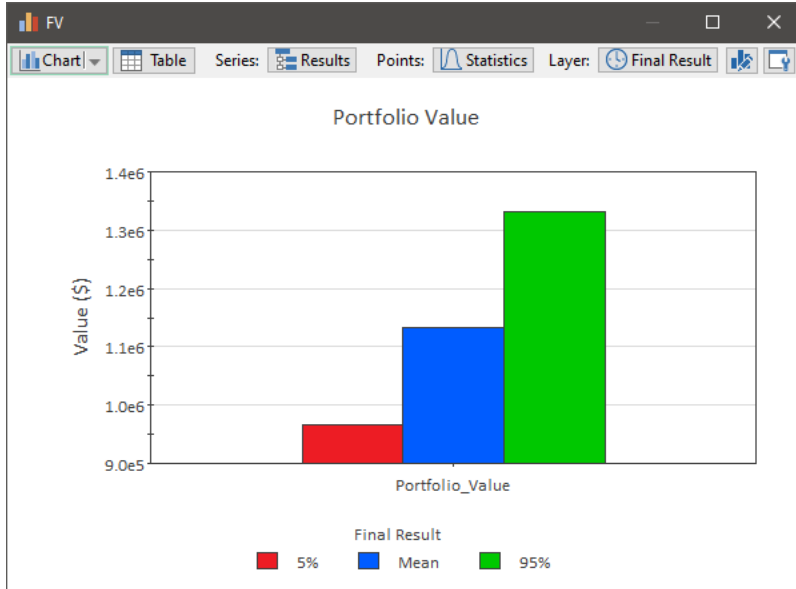
Common results that can be plotted include time histories:



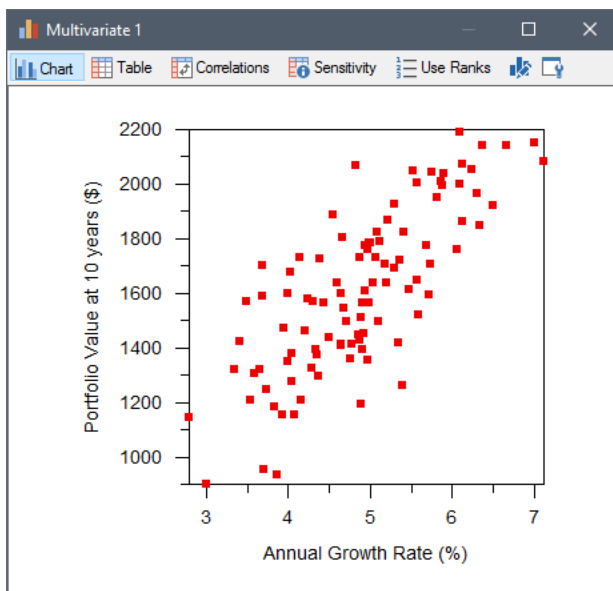
distributions:



column charts:



and scatter plots:



**Note:** Basic concepts associated with saving and viewing results in GoldSim are discussed in Chapter 3 of the [GoldSim User's Guide](#).





---

# Chapter 5: Example Financial Module Applications

High finance isn't burglary or obtaining money by false pretenses, but rather a judicious selection from the best features of those fine arts.

Finley Peter Dunne

## Chapter Overview

This chapter describes a number of example models that are installed with the Financial Module.

---

**Note:** All of the models described in this chapter can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu).

---

## In This Chapter

This chapter discusses the following topics:

- Basic Financial Module Applications
- Advanced Financial Module Applications

## Basic Financial Module Applications

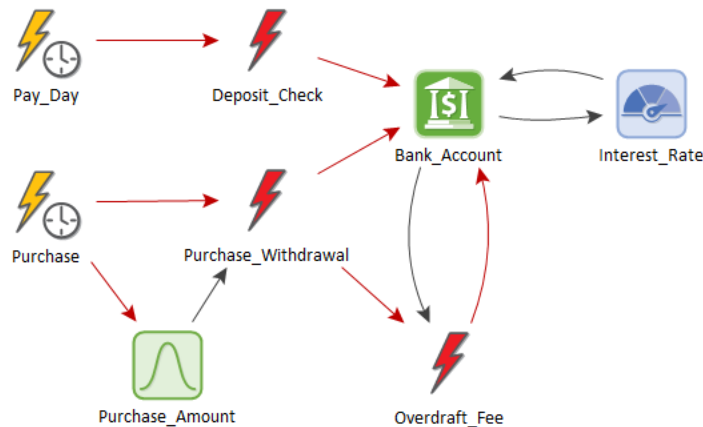
The examples described below illustrate the use of each of the five Financial Module elements. A sixth example illustrates how to simulate variable exchange rates in a financial model.

### Example: Using the Fund Element

The Fund element simulates funds and accounts with a specified initial balance, deposits, withdrawals, specified nominal rate and compounding frequency. This element is used to model bank accounts and loans and other financial components that are governed by an interest rate (e.g., loans). You specify deposits and withdrawals, and the element outputs the current value of the Fund.

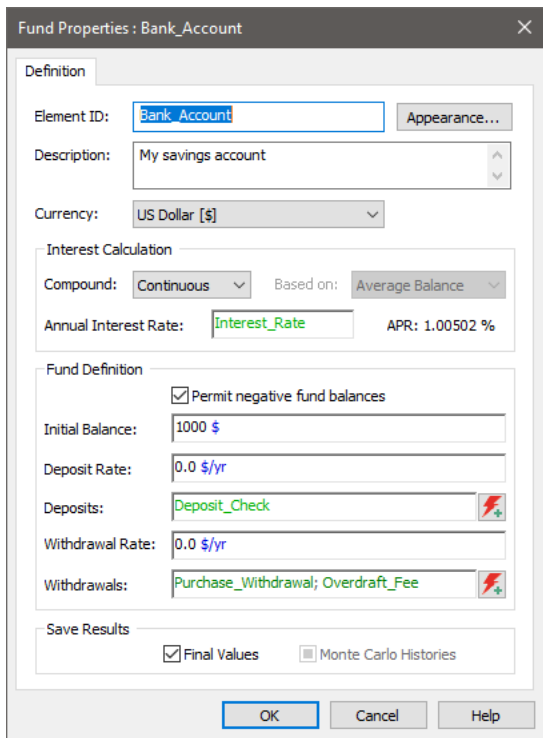
**Read More:** *The Fund Element* on page 50

The Fund.gsm example model can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu). It demonstrates how a Fund element could be used to model a bank account:



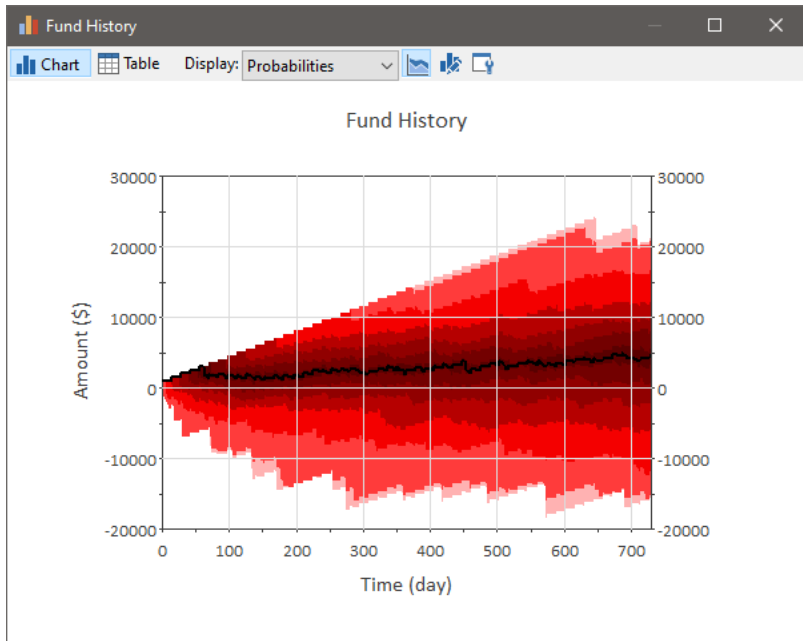
In this example, the Fund holder is paid \$500 every 14 days, which is deposited into the fund. On average, once every 90 days, the account holder makes a purchase (with the amount sampled from a Triangular distribution), which is then deducted from the fund.

The Fund element is set up as follows:



The Fund element tracks the net value of deposits and purchases, and applies interest to the average balance at a nominal rate of 1% per year (if it has a positive amount). The Fund has an initial balance of \$1000, and can go negative. If a purchase occurs while the account has a negative amount, an overdraft fee of \$50 is charged. Moreover, interest is charged on the total overdraft at a rate of 10% per year.

The Fund's output is as follows:



The monthly additions can be seen along with the spending over the course of the month. Note that the amount in the account is actually negative during some realizations, and as a result, the Fund element would have automatically applied interest charges instead of paying interest (as well as an overdraft fee for every purchase).

**Read More:** *Example: Simulating a Loan Using a Fund Element* on page 92

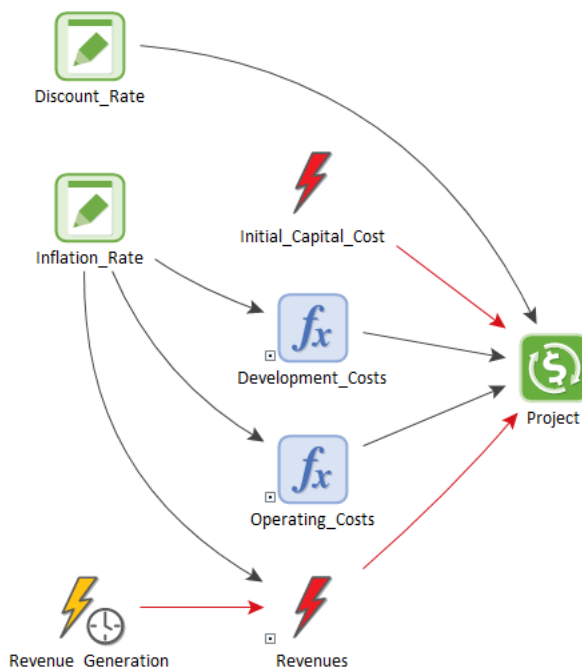
- *The Fund Element* on page 50
- *Example: Simulating a Loan Using a Fund Element* on page 92

## Example: Using the Cash Flow Element

The Cash Flow element computes the net cash flow, the net present value (NPV) and internal rate of return (IRR) of a cash flow history. This element is used to model the future return of projects, business ventures, and similar undertakings.

**Read More:** *The Cash Flow Element* on page 54

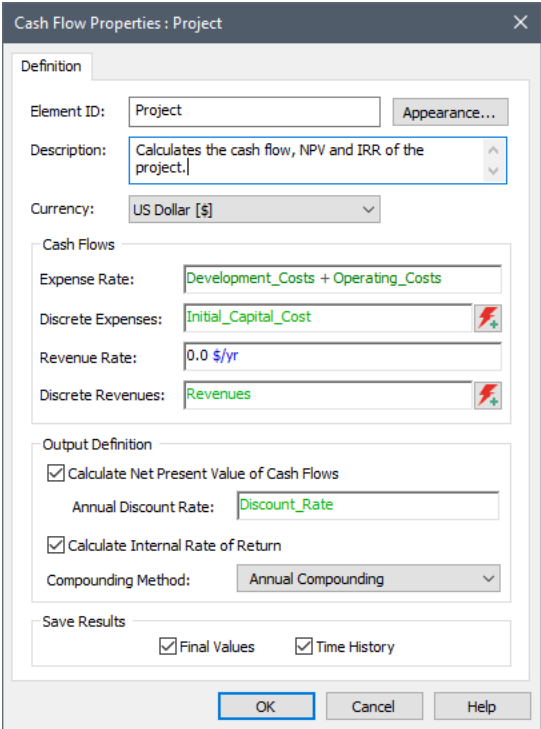
The CashFlow.gsm example model can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu). It demonstrates the use of the Cash Flow element to simulate a project:



In this case, a capital expenditure (represented by the Initial\_Capital\_Cost Discrete Change element) is made at the start of the simulation. Development costs occur continuously over the first two years of the simulation and are adjusted upward by the Inflation\_Rate over time. Operating costs occur continuously over the remainder of the simulation and also increase by the Inflation\_Rate each year.

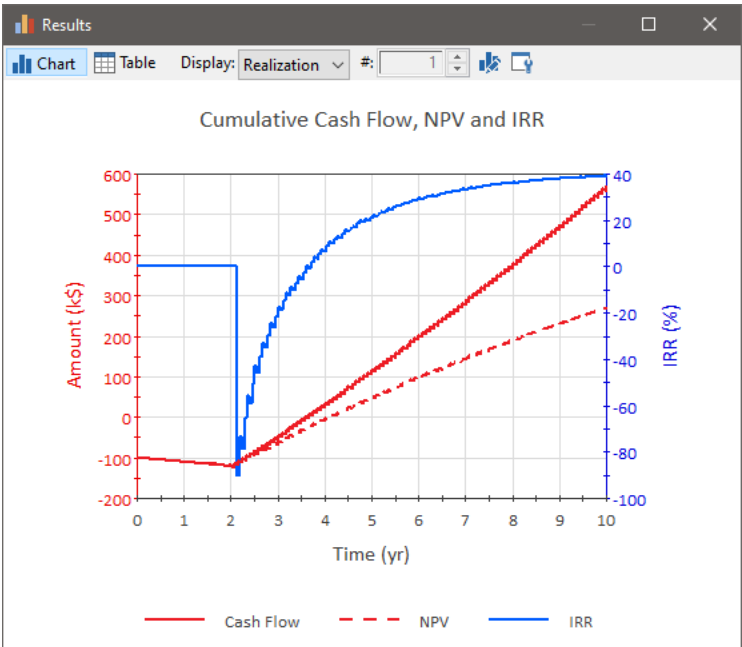
Revenues occur on a monthly basis, but only after two years. To model this, the Revenue\_Generation Timed Event emits an event on a regular basis once per month. The Revenues Discrete Change uses the Revenue\_Generation event as a trigger, but ignores events that arrive prior to two years of simulated time. Like costs, Revenues are also adjusted upwards by the Inflation\_Rate.

To set up the Cash Flow element, you simply connect continuous and discrete inflows and outflows to the appropriate fields in the element:



Once the cash flows have been specified, NPV and IRR can be calculated by checking the boxes in the Output Definition section of the property dialog. In the case of the NPV, an annual discount rate also needs to be specified.

In this model, Cash Flow (the main output of the element), the NPV and the IRR are plotted together in one Time History element:



The IRR graph for the first two years may seem counter intuitive given that the project generates no revenue during that period. However, this is due to the fact that the IRR output displays zero until the first point in the simulation where the calculation actually converges (revenues must occur before the solution will converge).

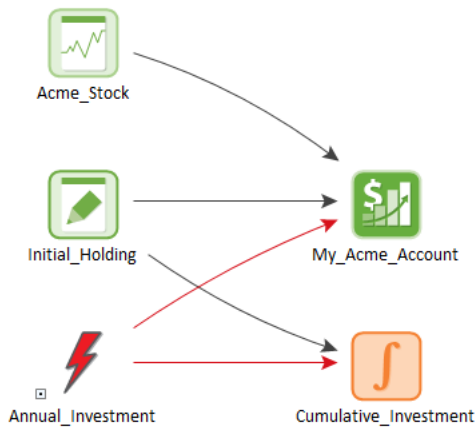
**Read More:** *Example: Comparing Alternatives Using a Cash Flow Element - Repair vs. Replace* on page 95

## Example: Using the Investment Element

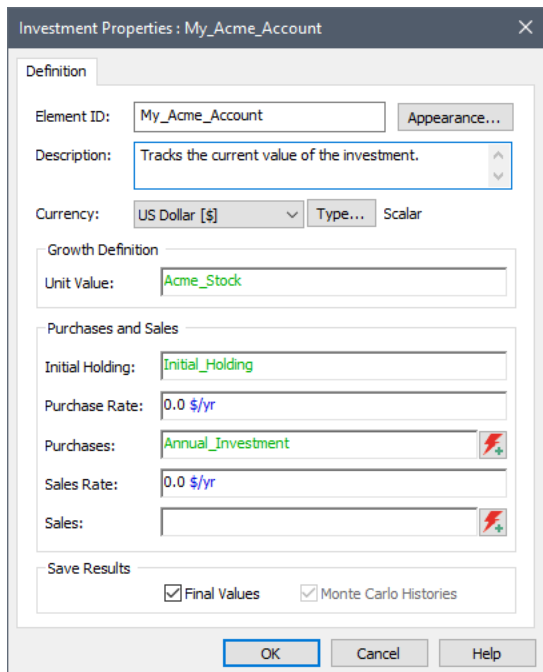
The Investment element simulates the growth of an investment or group of investments such as a security or portfolio of securities. You input purchases, sales and the unit value of the investment, and the element outputs the investments current value.

**Read More:** *The Investment Element* on page 58

The Investment.gsm example model can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu). It demonstrates how an Investment element could be used to model the performance of an investment in a stock:



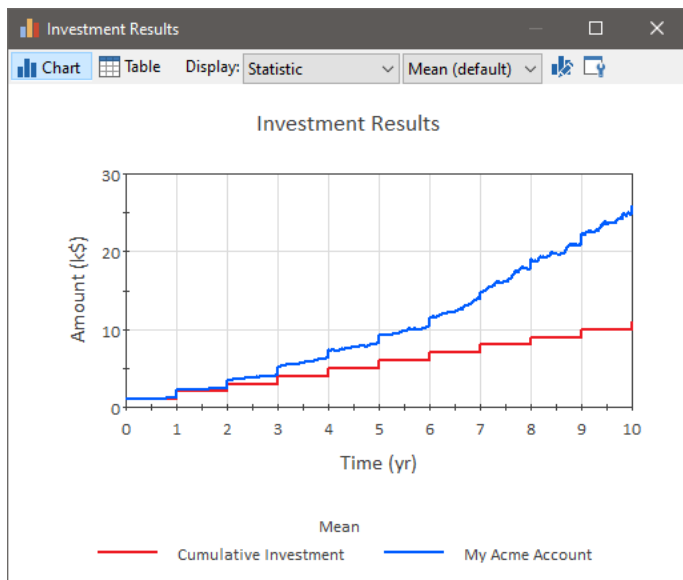
In particular, the model simulates the performance of an investment into Acme stock over a ten year period. In this case, the initial holding is \$1000, and an additional \$1000 is invested at the end of each simulated year:



The price of the stock is modeled using a History Generator element, which is then used by the Investment element (My\_Acme\_Account) to determine the current value of investment holdings.

**Note:** The History Generator element is discussed in detail in Chapter 10 of the [GoldSim User's Guide](#).

The result of the model is a graph which tracks the current value and the total amount invested over the duration of the simulation:



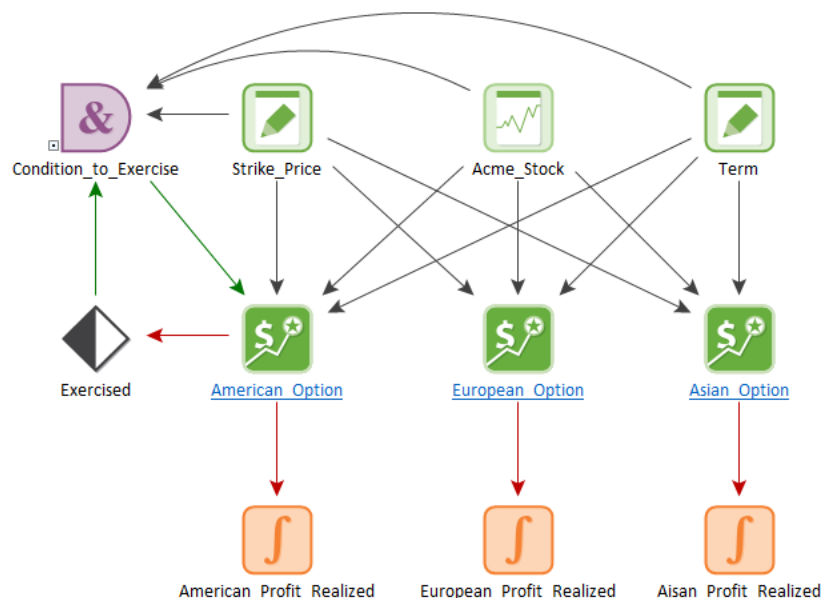
**Read More:** *Example: Simulating a Stock Portfolio* on page 97; *Example: Simulating Long and Short Positions on a Security* on page 99

## Example: Using the Option Element

The Option element simulates the acquisition and exercise of financial options (puts and calls). You specify the type of option, option properties (e.g., number of units, strike price, term), the unit value of the underlying stock, and when the option is acquired and exercised, and the element outputs the value realized upon exercise of the option.

**Read More:** *The Option Element* on page 61

The Option.gsm example model can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu). It demonstrates how an Option element can be used to model American, European and Asian calls on an underlying stock:



In the model, a History Generator element is used to model the price movement of the underlying stock, which is the input into three different Option elements.

---

**Note:** The History Generator element is discussed in detail in Chapter 10 of the [GoldSim User's Guide](#).

---

The strike price, option type, term, and conditions for exercise prior to expiry (for American options) are all specified in the option element:



Option Properties : American\_Option

Definition

Element ID:  Appearance...

Description: A call option: can buy at the Strike Price

Currency: US Dollar [\$]

Option Details

Put Option  Call Option

Type: American Option

Control:  Acquire...  Exercise...

Option Properties

Number of Units:

Unit Value:

Term:

Strike Price:

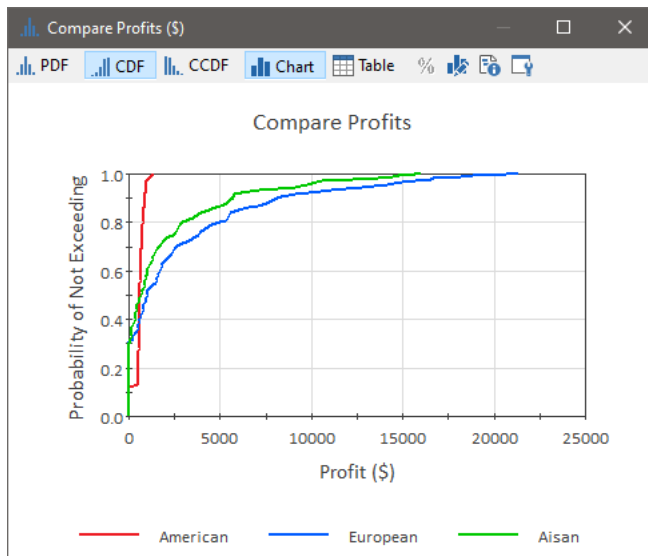
Save Results

Final Values  Monte Carlo Histories

OK Cancel Help

If the condition for exercise occurs, or the value of the option is greater than zero at the end of the option's term, the option is exercised and a discrete change representing profit from the option is placed in the Profit\_Realized element.

Given the stochastic nature of the underlying stock (as modeled by the History Generator), multiple Monte Carlo realizations are carried out. The primary output of the model is a Distribution Result comparing the profits of the three different types of options:



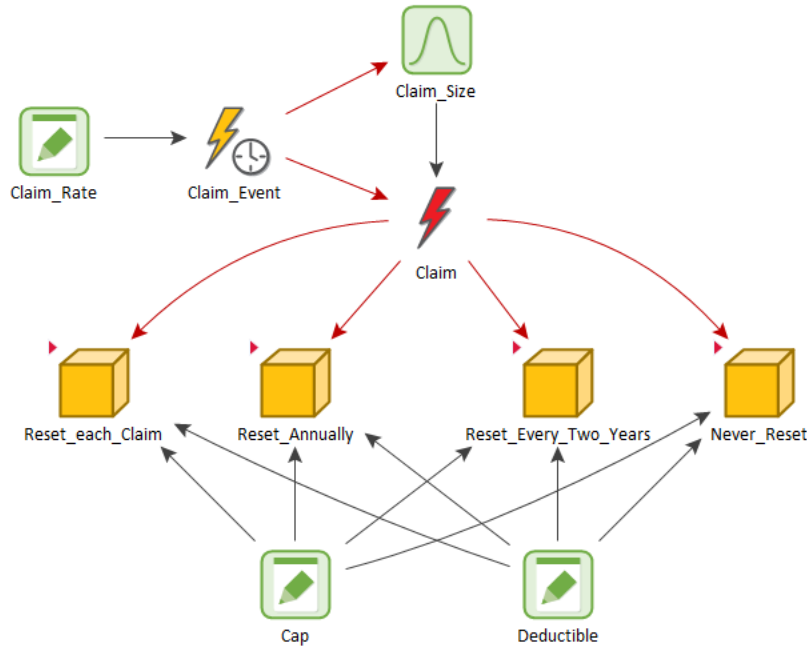
**Read More:** *Example: Simulating Long and Short Positions on a Security* on page 99

## Example: Using the Insurance Element

The Insurance element simulates claims against an insurance policy. You must specify the properties of the insurance policy (the deductible, the cap, and when these are reset), and then you specify one or more discrete claims. The key outputs are the cumulative covered and uncovered losses for the claims.

**Read More:** *The Insurance Element* on page 66

The Insurance.gsm example model can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu). It demonstrates the use of the Insurance element:



In the file, each Container holds an Insurance element that is subject to the same claims. Each Insurance element has the same deductible and cap, but the deductible and cap is reset at different times (after each claim, annually, after two years, and never):

Insurance Properties : Insurance\_Policy

Definition

Element ID: Insurance\_Policy Appearance...

Description: Represents the insurance policy (deductible and cap reset after each claim).

Currency: US Dollar [\$]

Definition

Claims: Claim

Deductible: Deductible

Cap: Cap

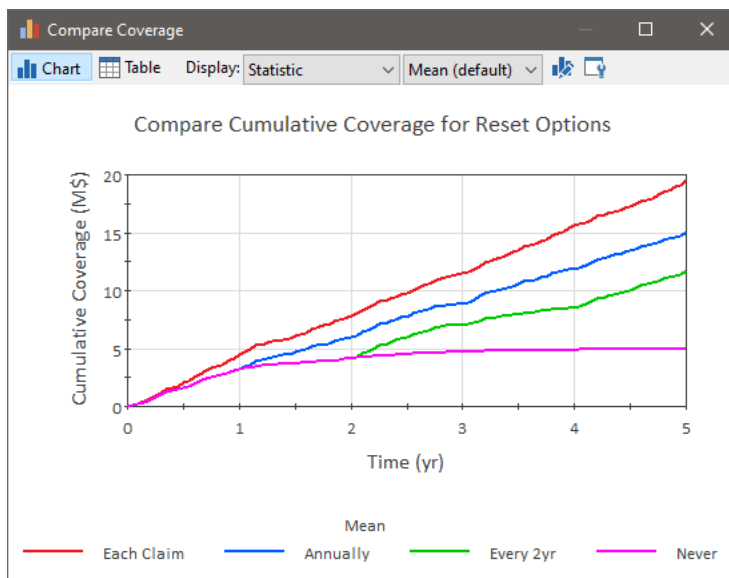
Reset Deductible and Cap: For each claim Trigger...

Save Results

Final Values  Monte Carlo Histories

OK Cancel Help

The results of the model (for each Container) provide information on covered and uncovered claims, along with changes in deductible and cap balances. The four different policies are then compared (in terms of the cumulative amount of coverage):



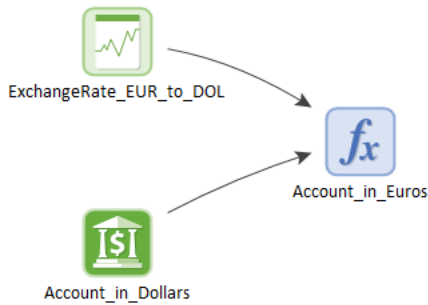
**Read More:** *Example: Simulating a Reinsurance Policy* on page 101

## Example: Simulating Variable Exchange Rates

Like all other conversion factors, the conversion factors between currencies (i.e., the exchange rates) are treated as constants in GoldSim. Unlike other conversion factors, however, in the real world, exchange rates vary with time.

**Read More:** *Modeling Variable Exchange Rates* on page 71

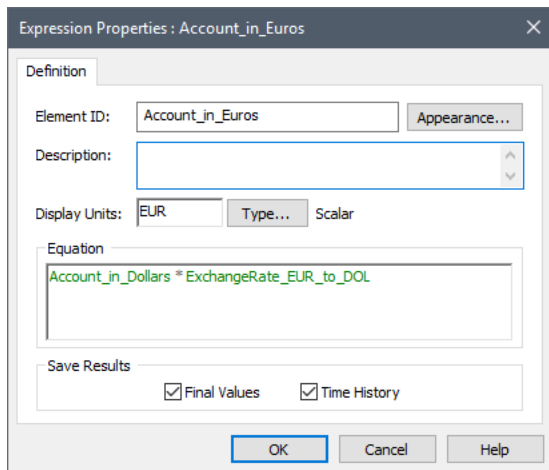
The VariableExchange.gsm model can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu). It demonstrates how you can use dynamic exchange rates in your financial models:



In this model, instead of using the currency conversion features in GoldSim (which assume static exchange rates), a History Generator element is used to model the exchange rate.

**Note:** The History Generator element is discussed in detail in Chapter 10 of the [GoldSim User's Guide](#).

This exchange rate is used to manually calculate the value in the new currency within an Expression element:



Note that when variable exchange rates are used and modeled in this manner, it is critical that all conversion rates in the Currencies dialog be set to 1.

## Advanced Financial Module Applications

The examples described below illustrate several more advanced applications and techniques for the Financial Module elements.

### Example: Simulating a Loan Using a Fund Element

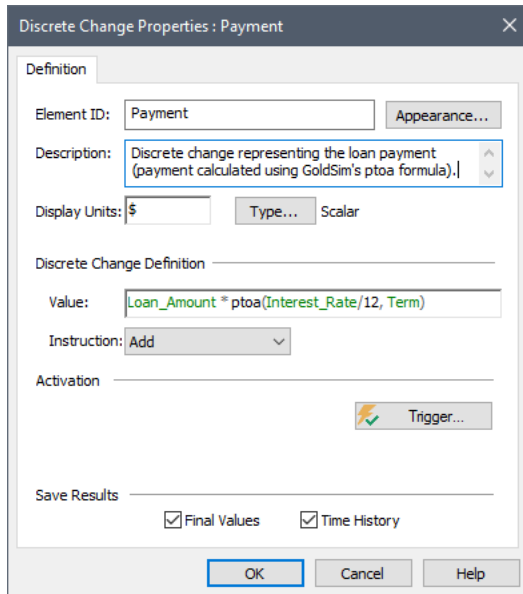
In addition to modeling interest-bearing securities and accounts, Fund elements can be used to model loans. This can be done in one of two ways: with the loan amount as a positive fund balance or with the loan amount as a negative fund balance.

**Read More:** *The Fund Element* on page 50

GoldSim automatically applies the specified interest rate as a credit to a positive balance and as a debit to a negative balance. In both cases, the nominal interest rate is specified as a positive value.

The Loan.gsm example model can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu). It demonstrates how a loan can be simulated. The file illustrates both methods for representing a loan (modeling the loan amount as a positive fund balance and modeling the loan amount as a negative fund balance).

In this model, payments are applied as discrete monthly events (using a Discrete Change element). The value of each payment is calculated using GoldSim's built in "ptoa" function, which converts a present value (in this case a loan amount) to an annuity with equal monthly payments over the duration of the loan:

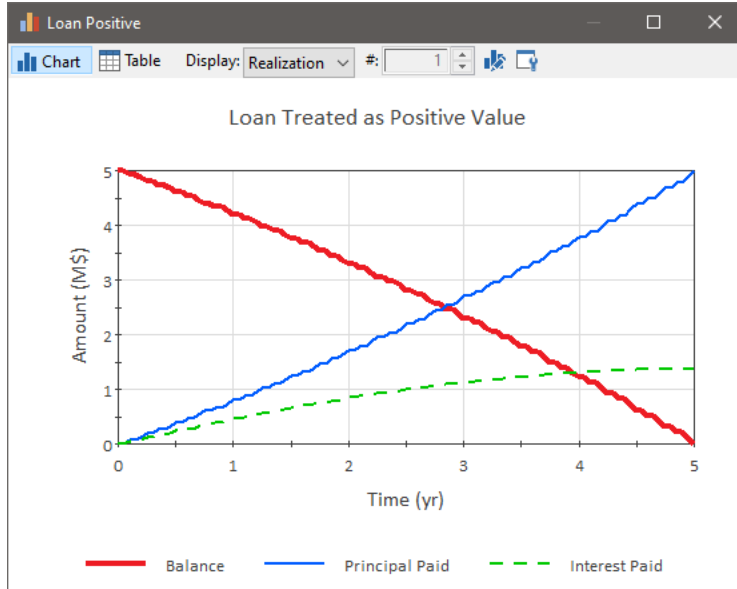


**Read More:** *The Built-in Financial Functions* on page 27

The payment is applied as a withdrawal if the loan is modeled as a positive balance:

and a deposit if the loan is modeled as a negative balance:

The cumulative amount of payments is tracked in the Cumulative\_Payments integrator. Cumulative interest payments are equal to the interest applied to the loan. The principal payment is equal to the payments made less the interest paid. The value of the loan along with cumulative interest and principal payments can then be plotted over the course of the simulation:



## Example: Comparing Alternatives Using a Cash Flow Element - Repair vs. Replace

The Cash Flow element is used to model the future return of projects, business ventures, and similar undertakings. As such, it can be used to compare and contrast alternatives.

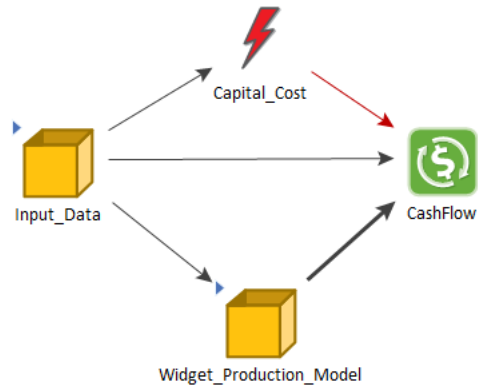
**Read More:** *The Cash Flow Element* on page 54

The CashFlowAlternatives.gsm example model can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu).

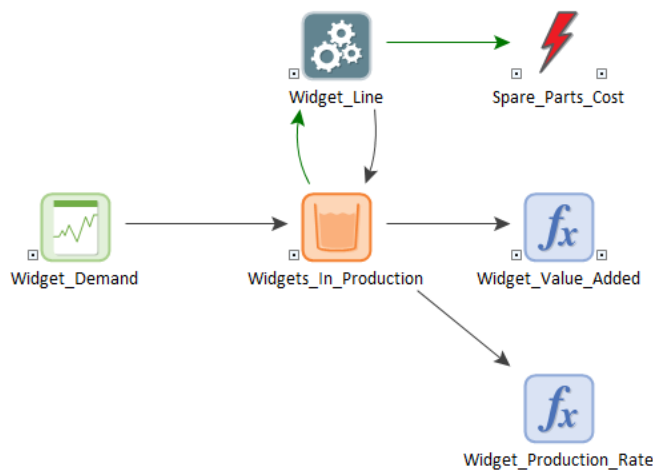
It illustrates how the Cash Flow element can be used to compare and contrast alternatives by comparing refurbishment and replacement options for an aging production line. The model is set up to use GoldSim's scenario feature.

The line manufactures widgets, and each widget can be sold for \$300 more than the raw materials used to produce it. Demand for widgets varies over time. The company wants to determine whether it is better to refurbish or replace the line with a higher capacity unit. Note that replacing the line is more expensive, but a refurbished line would be more prone to breakdown and more expensive to repair.

The model considers three scenarios: the base line (do nothing), refurbish, or replace. The Input Containers contain the basic inputs (a number of which vary depending on the scenario):



The Widget\_Production\_Model Container contains a Reliability Function element (from the Reliability Module), a Reservoir to keep track of the widgets being produced, as well as some additional support elements.



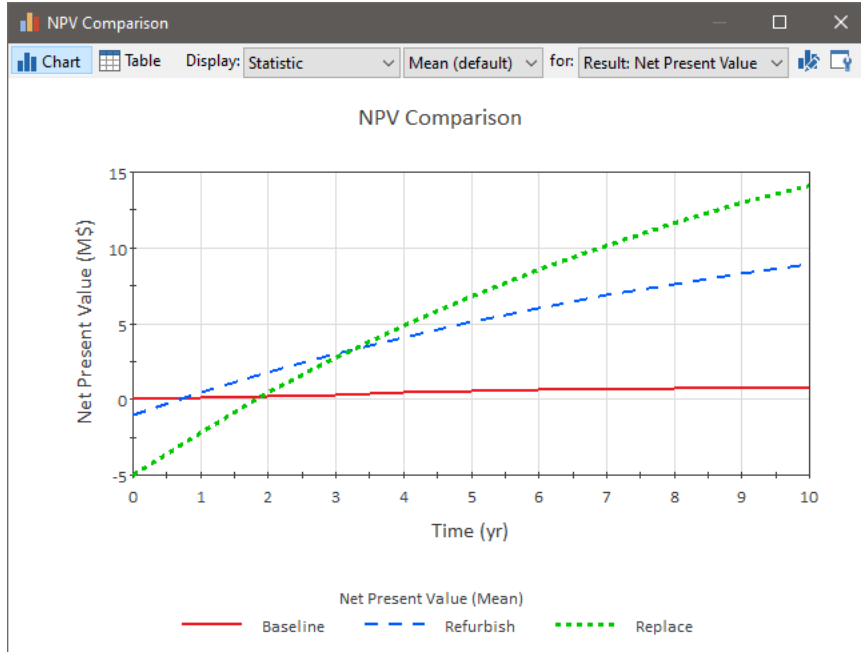
**Note:** The History Generator element is discussed in Chapter 10 and Scenarios are discussed in Chapter 7 of the [GoldSim User's Guide](#). The Reliability Module (and the Function element) are discussed in detail in the [Reliability Module User's Guide](#).

The Reliability element (in this case, a Function element) is used to model breakdown and repair of the Widget line, with failure of the line triggering a discrete change with a value equal to the cost of spare parts required for repair. When the Widget line is operating, it adds widgets to the Widget\_Production Reservoir at the line's capacity until the Reservoir holds 250 Widgets (representing a full warehouse).

Widgets are removed from the Reservoir element in each Container according to the simulated demand output by a History Generator. This is multiplied by the value added per widget and passed through to the Revenues input of the Cash Flow element.

By comparing the cash flows of the three scenarios it is possible to compare the options on a net cash flow and an NPV basis. The NPV comparison is shown below:





## Example: Simulating a Stock Portfolio

The Investment element simulates the growth of an investment or group of investments such as a security or portfolio of securities. You input purchases, sales and the unit value of the investment, and the element outputs the investments current value.

**Read More:** *The Investment Element* on page 58

The Portfolio.gsm example model can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu). It demonstrates how the GoldSim financial elements can be used to model a portfolio of stocks

A History Generator element is used to generate vectors of stochastic data which is correlated (in this case, using a Gaussian copula). In this model, the History Generator element outputs data for ten different stocks:



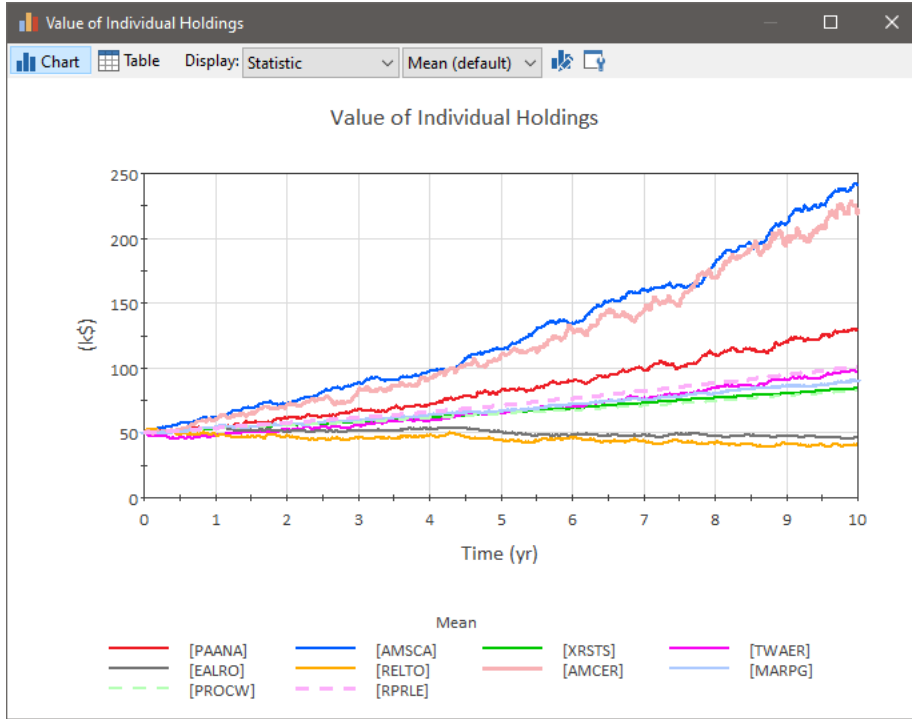
For each stock, a growth rate, volatility, reversion, initial price and median price have been specified (by entering these as vectors). A sparse correlation matrix and a Gaussian copula have also been specified to further reflect the behavior of the ten stocks.

---

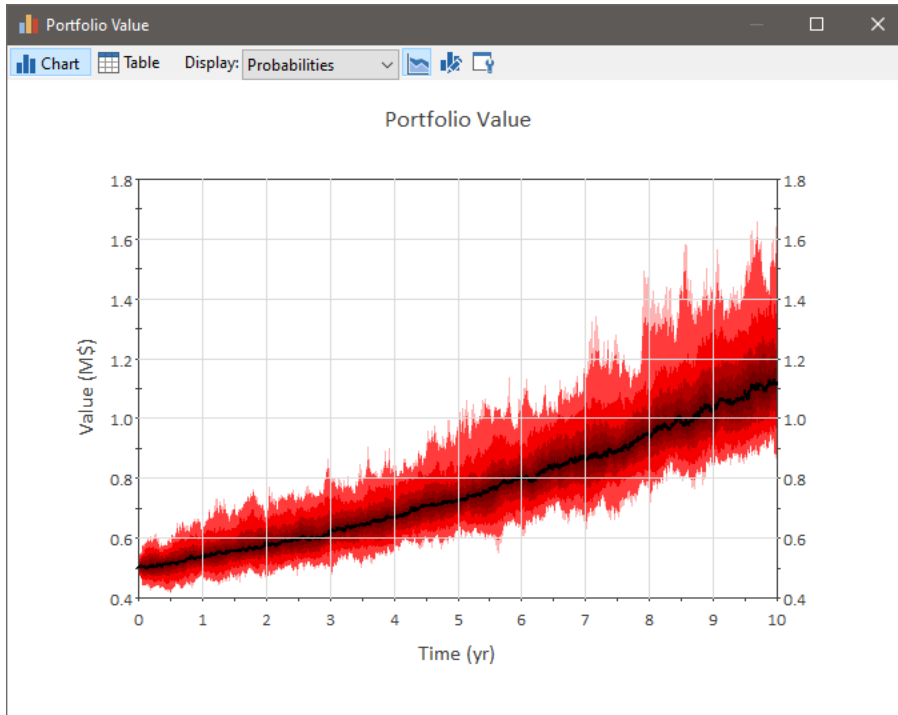
**Note:** The History Generator element is discussed in detail in Chapter 10 of the [GoldSim User's Guide](#).

---

The History Generator element serves as an input to an Investment element (defined as a vector) that tracks movement in a portfolio with an initial investment of \$50,000 in each of the ten stocks:



The ten items of the Investment element's vector output can then be summed to obtain the portfolio value:



### Example: Simulating Long and Short Positions on a Security

The Positions.gsm example model can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu). It demonstrates how

different positions in a single stock might be modeled. This relatively complex model uses Fund, Investment and Option elements.

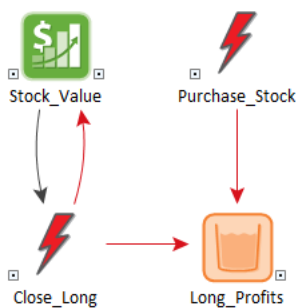
**Read More:** *The Fund Element* on page 50; *The Investment Element* on page 58; *The Option Element* on page 61

A History Generator is used to simulate movement in the stock price, while four Containers simulate the current value and profits from Long, Short, Put and Call positions.

**Note:** The History Generator element is discussed in detail in Chapter 10 of the [GoldSim User's Guide](#).

### Long Position

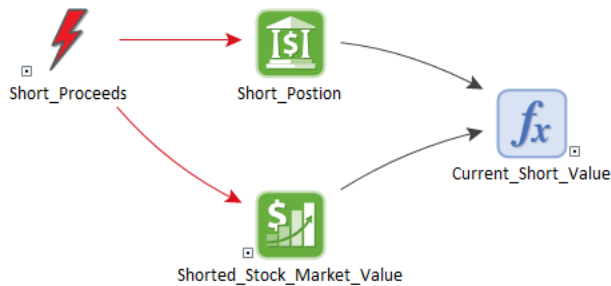
In this model a long position is simulated using an Investment element and a Reservoir:



The Investment element simulates the value of the stock as it changes over time, while the Reservoir element has the cost of the stock subtracted from it at the start of the simulation, and the value of the position added to it when it is closed. In this model, the Close\_Long Discrete Change is triggered when the long position's value is \$2000 greater than its starting value or at the end of the simulation, but any valid triggering condition could be used.

### Short Position

A short position is somewhat more complex to simulate as profits or losses need to be accrued and are not applied until the position is closed. In this example, a conditional Container is used. Elements inside the conditional Container track the difference between the current value of the investment and the proceeds of the short (which are arbitrarily invested in an interest bearing security modeled by a Fund element):

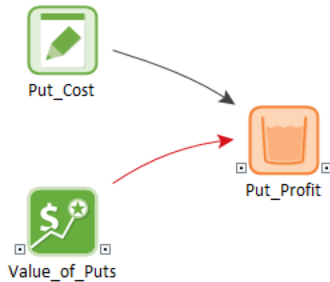


When the conditional Container is activated at the start of the simulation, it immediately triggers a Discrete Change element representing the short sale and the proceeds are placed into the Short\_Position fund where they accrue interest.

They are also placed into an Investment element that tracks the value of a long position in the stock. The Short\_Position less the Shorted\_Stock\_Market\_Value is equal to the current value of the short. Once the conditional Container is deactivated (in this model, deactivation is triggered when profits are greater than \$1000, losses are greater than \$500, or the end of the simulation is reached) the Current\_Short\_Value is frozen and can then be used to calculate profits.

### Put and Call Positions

In this model, the Put and Call conditional Containers are identical except for the option type. The Put model is as follows:



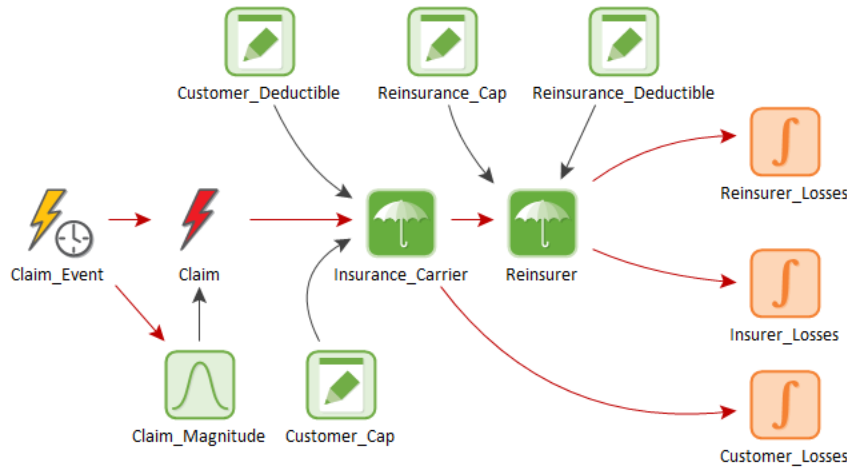
The initial value of the Profit Reservoir is set to a negative value equal to the cost of the puts. The Option element is set up to acquire the option at the start of the simulation and its main output provides the current value of the options. When the option is exercised (either because it is worth \$1000 or more, or because it has a positive value at the end of the option's term) the holding's current value is added to the Profit reservoir by the Exercise Discrete Change output of the Option element.

## Example: Simulating a Reinsurance Policy

The Insurance element simulates claims against an insurance policy. You must specify the properties of the insurance policy (the deductible, the cap, and when these are reset), and then you specify one or more discrete claims. The key outputs are the cumulative covered and uncovered losses for the claims.

**Read More:** *The Insurance Element* on page 66

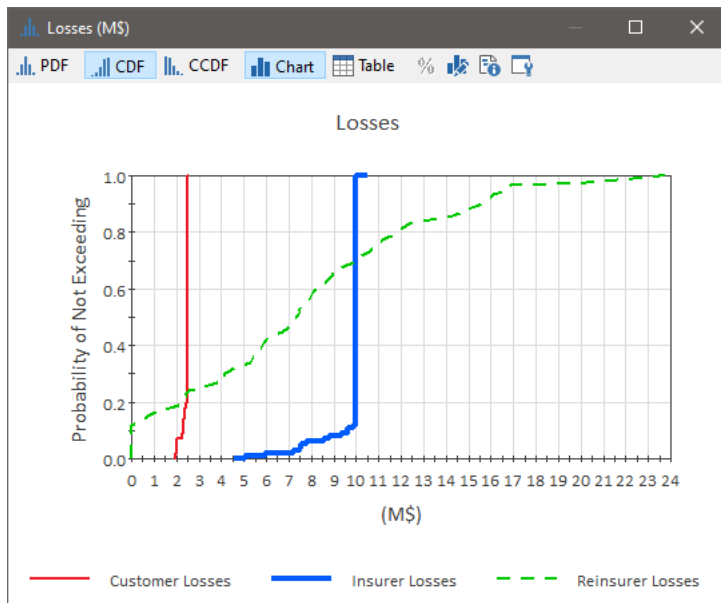
The Reinsurance.gsm example model can be found in the can be found in the Financial Examples folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu). In this model, two insurance elements are used to model a reinsurance arrangement (in which a primary insurer is insured for excess losses by a reinsurer):



Claim events occur 4 times per year on average, and the magnitude is sampled from a log-normal distribution. These claims are first submitted to the primary insurer, where the policy has a deductible and cap that are reset each year.

Deductibles paid by the customer and claims that exceed the cap are the responsibility of the customer (these amounts can be obtained from the Uncovered discrete change output or the Cumulative Uncovered output of the Insurance\_Carrier element). Claims paid by the primary insurer are then submitted to the reinsurer, where the contract has its own deductible and cap (this time over the life of the policy). Claims accepted by the second Insurance element are the responsibility of the Reinsurer, while claims that are not covered are the responsibility of the primary insurer.

The graph below shows the probability distributions of the cost of claims to the customer, primary insurer and reinsurer over five years:



Note that the customer has about an 80% probability of paying \$2.5 million (the customer's annual deductible times five years), and the insurer has about a 90% probability of paying \$10 million (the insurer's deductible, which is never reset).

---

# Glossary

## A

---

### **American Option**

An option that may be exercised at any time prior to the expiration date.

### **Asian Option**

An option that may be exercised only at the expiration date, and is valued based on the average price of the underlier over a given period of time. In GoldSim, this is always the time since the option was acquired.

## C

---

### **Call Option**

An option that gives the buyer the right to buy stock at a specified price (the exercise price) within a given period of time.

## E

---

### **European Option**

An option that may be exercised only at the expiration date.

## I

---

### **Internal Rate of Return**

A measure of the profitability of a project, venture or undertaking. It represents the discount rate at which the net present value of the project is equal to zero.

## N

---

### **Net Present Value**

The present value of a project's future cash flow (i.e., the cash value today of future returns) minus the initial investment.

## P

---

### **Put Option**

An option that gives the buyer the right to sell stock at a specified price (the exercise price) within a given period of time.





---

# Index

## A

Activating  
    Financial Module 7  
American options 62  
Asian options 62

## B

Blog 9

## C

Call options 62  
Cash Flow element  
    adjusting for inflation 56  
    advanced example 95  
    details 54  
    expenses 55  
    internal rate of return 58  
    IRR 58  
    net present value 57  
    NPV 57  
    outputs 57  
    overview 22  
    revenues 55  
    simple example 84  
Claims, for Insurance element 68  
Currencies 70  
    example 91  
    variable exchange rates 71

## D

Discrete events 13  
Dynamic simulation  
    for financial models 74

## E

European options 62  
Exchange rates 70  
    example 91  
    variable 71

---

## F

### Financial functions

- interest rate conversions 28
- returns and volatility 29
- simple compounding 28

### Financial Module

- activating 7
- built-in compounding functions 27
- comparison with spreadsheets 3
- element details 49
- example applications 81
- getting started 11
- introductory example 33
- learning to use 5
- list of elements 19
- overview 19
- running a simulation 73
- saving results 77
- user requirements 4
- viewing results 77
- what is 3

### Fund element

- actual withdrawal rate 54
- actual withdrawals 54
- deposits 52
- details 50
- initial balance 52
- interest rate 51
- negative balances 52
- outputs 54
- overview 20
- resetting value 52
- simple example 82
- simulating a loan 92
- withdrawals 52

## H

Help system 8

## I

Inflation, representing in Cash Flow element 56

### Insurance element

- advanced example 101
- claims 68
- details 66
- outputs 69
- overview 26
- policy properties 67
- simple example 89

---

Internal rate of return, Cash Flow element 57

Investment element

advanced example 97, 99

details 58

growth definition 59

output 61

overview 23

purchases 60

sales 60

simple example 86

unit value 59

IRR, Cash Flow element 57

## M

Maintenance program 8

Manual

organization 5

Model Library 9

Monte Carlo options

for a Financial model 76

Monte Carlo simulation

for financial models 76

## N

Net present value, Cash Flow element 57

NPV, Cash Flow element 57

## O

Option element

acquiring 63

advanced example 99

details 61

exercising 64

outputs 66

overview 24

simple example 87

type 62

unit value 64

valuing 64

Organization of manual 5

## P

Put options 62

## S

Spreadsheets

comparison with Financial Module 3

---

Support 9

## T

Technical support 9

Time Options, for a Financial model 74

Timesteps

    inserting for financial events 75

Training 9

Triggering 13