User's Guide

Reliability Module

GoldSim
TECHNOLOGY GROUP

Partial funding for the GoldSim Reliability Module was provided by NASA.

GoldSim Version 14.0 (October 2021)

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

Visit us at our web site: www.goldsim.com
Email us at: software@goldsim.com

Contents

Chapter 1: Introduction

> **The major difference between a thing that might go wrong and a thing that cannot possibly go wrong is that when a thing that cannot possibly go wrong goes wrong, it usually turns out to be impossible to get at or repair.**
>
> **Douglas Adams, *Mostly Harmless***

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 Reliability Module is a program extension to GoldSim which allows you to probabilistically simulate the reliability and performance of complex engineered systems over time. GoldSim provides the ability to model the interdependence of components through requirements and fault trees, as well as the capability to define multiple independent failure modes for each component. This facilitates both reliability modeling and risk analysis.

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 Reliability Module.

Note: This document only describes the Reliability Module, a program extension to the GoldSim simulation framework. In order to take full advantage of all of the powerful features of the Reliability Module, you will eventually 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 Reliability Module?
- How Does GoldSim's Approach Differ From Other Reliability Modeling and Risk Analysis Approaches?
- What is the Difference Between Reliability Professional and Reliability Learning Edition?

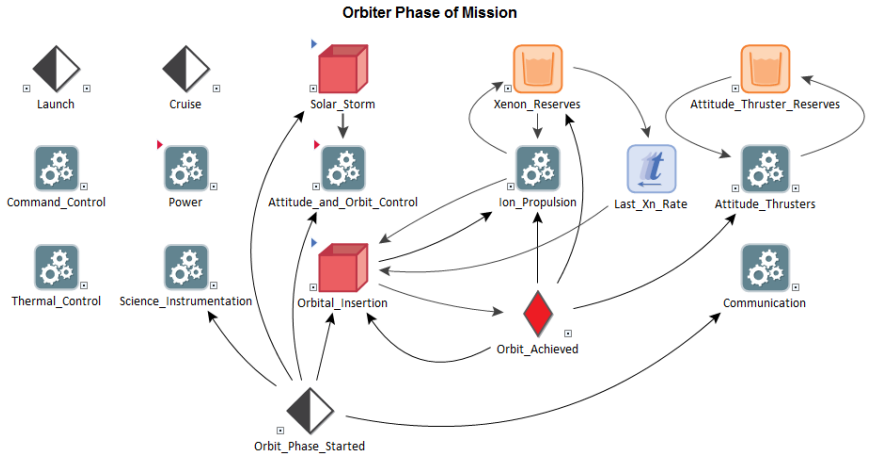
- Who Should Use the Reliability Module?
- How the GoldSim Documentation is Organized
- Learning to Use the Reliability Module
- Conventions Used in this Manual
- Activating the Reliability Module
- What Does the Reliability Module Add to the GoldSim User Interface?
- Using Help
- Technical Support, User Resources and Software Upgrades
- References

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 mine, a watershed, a forest, the organs in your body, the atmosphere) 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 which 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 the program to address specialized problems. The Reliability Module is one of these program extensions.

What is the Reliability Module?

Reliability engineering involves measuring and analyzing the ways that systems can fail (and be repaired) in order to increase their design life, and eliminate or reduce the likelihood of failures, downtime and safety risks. It involves developing a mathematical representation (a model) of an existing or proposed engineered system in order to predict the performance of the system over time. The system (e.g., a furnace) consists of multiple components (e.g., a blower, a burner). The output of these models typically consists of predictions of measures such as ***reliability*** (the probability that a component or system will perform its required function over a specified time period) and ***availability*** (the probability that a component or system is performing its required function at any given time). Reliability models are frequently used to compare design alternatives on the basis of metrics such as warranty and maintenance costs

For some systems, the analyst may be more concerned with ***risk analysis*** than with reliability. Risk analysis focuses on predicting the probability of those (presumably rare) failures that can lead to injury, loss of life, severe damage to the system, or perhaps damage to the surrounding environment. Hence, in a risk analysis, the output of the model typically is the probability of a particular unlikely, but high consequence outcome (e.g., catastrophic failure of the system), and identification of those events or components most likely to lead to that outcome. Risk analysis models are typically used to inform decisions about required levels of redundancy, and to evaluate system safety and risk.

The GoldSim Reliability Module is a program extension to GoldSim which allows you to probabilistically simulate the reliability and performance of complex engineered systems over time. The fundamental outputs produced by the Reliability Module consist of predicted reliability metrics (e.g., reliability and availability) for the overall system, and for individual components within that system. The Reliability Module can also be used to compute the probability of specific consequences (e.g., catastrophic failure of the system) to support risk analysis. GoldSim catalogs and analyzes failure scenarios, which allows for key sources of unreliability and risk to be identified.

How Does GoldSim's Approach Differ From Other Reliability Modeling and Risk Analysis Approaches?

When discussing how GoldSim differs from other approaches, it is useful to differentiate reliability modeling from risk analysis. The GoldSim Reliability Module can be used for both types of analysis. With conventional methods, however, these two types of analysis use very different types of tools (since they are focused on different types of results).

Conventional Approaches to Reliability Modeling

An excellent discussion of conventional approaches to reliability modeling is provided by Ebeling (1997), and readers who are not familiar with these approaches are encouraged to consult this text.

Most reliability modeling approaches involve the assumption of a static model, where the system configuration never changes (other than due to the failure/repair of components), and where its properties don't change with time. This is a convenient assumption, as it allows the use of simple techniques, such as closed form mathematical equations or reliability block diagrams. Markov chains are another conventional reliability approach, and although they introduce an element of dynamism, the system itself (and its properties) cannot change with time. Because of the simplifying assumptions required to use these conventional techniques, they may be inappropriate for some systems.

Some of the difficulties with using these approaches for complex systems are summarized below.

Closed-Form Equations. These methods are heavily dependent on classical models (i.e., they have been primarily developed for use with standard failure distributions like the Poisson and Weibull). Even if failure data can be fitted to a standard distribution, it is difficult to model complex systems with closed form equations. For example, if a system has two Weibull failure modes, they cannot be algebraically combined into a single Weibull failure mode for use with the Weibull reliability equation.

Reliability Block Diagrams. Reliability block diagram models are static and do not account for the highly dynamic nature of many systems. A reliability block diagram model also assumes the system is in steady state, and unless correction factors are used, assumes that all of its components are independent.

Markov Chains. Markov chains enumerate a number of system "states" and the probabilities for transitioning between these states. However, the number of transition probabilities (and the computational effort) required to solve a Markov chain grows exponentially with the number of states. Because of this "state-space explosion", in many cases a system must be greatly simplified in order to use a Markov chain approach.

Of course, the conventional approaches are appropriate for many systems, particularly when employed by an experienced practitioner. However, in some cases, a more realistic reliability model may be required.

Conventional Approaches to Risk Analysis for Engineered Systems

Risk analysis is a very broad field, utilizing a variety of quantitative approaches. In the current context, however, we are primarily concerned with risk analysis of complex engineered systems (e.g., nuclear power plants, infrastructure such as dams, and space and defense systems) that are composed of highly-reliable and frequently redundant components, which in most cases are required to have an extremely low risk of a catastrophic failure.

The conventional approach to risk analysis for such systems focuses on the analysis of initiating events and subsequent event sequences that could lead to failures, and on enumerating and calculating the probabilities of different outcomes through tree-based analytical procedures (event trees/fault trees). Stamatelatos et al, (2011) and Vesely et al. (2002) provide good descriptions of these approaches.

For many types of systems (e.g., nuclear power plant probabilistic risk assessments), these approaches work well. However, systems that are highly dynamic or have significant process variability can be very difficult to model realistically using event tree/fault tree approaches, and they require a tremendous amount of preprocessing effort.

As a result, an approach like GoldSim's that facilitates explicit representation of dynamics and variability potentially provides a powerful complement to existing methods.

The GoldSim Approach to Reliability Modeling and Risk Analysis

GoldSim is a general purpose dynamic, probabilistic (Monte Carlo) simulator. Dynamic simulation allows the analyst to develop a representation of the system whose reliability is to be determined, and then observe that system's performance over a specified period of time.

The primary advantages of dynamic probabilistic simulation are:

- The system can evolve into any feasible state and its properties can change suddenly or gradually as the simulation progresses.
- The system can be affected by random processes, which may be either internal (e.g., failure modes) or external.
- If some system properties are uncertain, the significance of those uncertainties can be determined.

In Monte Carlo simulation, the model is run many times with uncertain variables sampling different values each time (each run is called a ***realization***). These realizations are each considered equally likely (unless specialized sampling techniques are used), and can be combined to provide not only a mean, but also confidence bounds and a range on the performance of the system. In addition to the statistical data these realizations provide, multiple realizations may also reveal failure modes and scenarios that may not be apparent, even to experienced risk and reliability modelers.

In addition to providing a more accurate representation of uncertainty, GoldSim also allows you to create a more detailed and accurate representation of your system than can be achieved with even the most sophisticated risk and reliability methodology.

With GoldSim, you can:

Model Components that have Multiple Failure Modes: GoldSim allows you to create multiple failure modes for components, each of which can either be defined by a distribution or occur when a specified condition arises. Failures which occur according to a distribution do not have to use time as the control variable. For example, a vehicle might use mileage to define failure, while an aircraft might use the number of cycles.

Model Complex Interdependencies: In addition to providing a logic-tree mechanism to define relationships, GoldSim also allows you to model the more subtle effects of failure on other portions of the system. For example, you can easily model a situation where the failure of one component causes another component to wear more quickly.

Model the External Environment: Reliability elements in GoldSim are fully compatible with all other GoldSim elements. This means that the environment in which the system operates can also be modeled, and can affect and interact with the system.

These features and capabilities provide a powerful engine for realistically modeling the risk and reliability of complex engineered systems.

What is the Difference Between Reliability Professional and Reliability Learning Edition?

There are two versions of the Reliability (RL) Module: Reliability Professional and Reliability Learning Edition. Your copy of GoldSim will have one or the other (but not both), depending on your license. Reliability Learning Edition is included automatically with GoldSim Pro. Reliability Professional must be purchased separately.

Reliability Learning Edition provides all of the features and capabilities of Reliability Professional. The only difference between the two versions is that Reliability Learning Edition limits you to adding no more than ten reliability elements. Reliability Professional imposes no limits on the number of reliability elements that can be added.

Who Should Use the Reliability Module?

The GoldSim Reliability Module is intended for use by engineers, scientists, researchers and students who are interested in understanding and predicting the reliability and risk of complex 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 Reliability Module are a clear understanding of the physical 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, failure modes and events controlling the behavior of the system to be modeled. This should include a good understanding of the fundamentals of reliability modeling and risk analysis.
- 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** (a companion document to this manual, described below) provides a brief primer on this topic, along with suggestions for further reading.

How the GoldSim Documentation is Organized

The GoldSim Documentation Suite

The Reliability Module is a specialized extension to the basic GoldSim simulation framework. The document you are reading only describes the GoldSim Reliability 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.

How this Manual is Organized

This document is organized into seven chapters and two appendices.

The seven chapters are as follows:

- **Chapter 1: Introduction**. The remainder of this chapter discusses the information required for you to get started using the GoldSim Reliability Module, including conventions used in the manual, activating the module, using online help, and obtaining technical support.
- **Chapter 2: Getting Started with the Reliability Module**. This chapter provides an overview of the features and capabilities of the Reliability 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 Reliability Module.
- **Chapter 3: The Reliability Elements**. This chapter describes the basic features of GoldSim's two reliability elements.
- **Chapter 4: Running a Reliability Simulation**. This chapter discusses how to set up and run both dynamic and static reliability simulations.
- **Chapter 5: Advanced Reliability Modeling Concepts**. This chapter describes some of the more advanced features of the reliability elements.
- **Chapter6: Displaying Reliability Results**. This chapter describes the reliability results available in GoldSim.
- **Chapter 7: Example Reliability Module Applications**. This chapter describes a number of examples which illustrate the application of the various features and capabilities of the Reliability Module.

The manual also includes two appendices:

- **Appendix A: Mathematical Details of the Reliability Module**. This appendix describes the mathematical details incorporated into the reliability elements.
- **Appendix B**: **Failure Mode Import Spreadsheet Format**. This appendix describes the format used when failure mode data is imported from a spreadsheet.

Learning to Use the Reliability 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 Reliability Module.

1. **Learn how to use the basic GoldSim framework first**. In order to use the Reliability 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 is available from the GoldSim splash screen, and can also be accessed from the main GoldSim menu (**Help | Tutorial…**).

In addition to the Tutorial, a free "hands-on" online training Course (titled "Introduction to GoldSim") is available that 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 Reliability 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. You can find the Course here: https://www.goldsim.com/Courses/BasicGoldSim/.

2. **Read "Getting Started with the Reliability Module"**. This is available both within the Help File and in the Reliability Module User's Guide (as Chapter 2). This provides an introduction to and a "quick tour" of the GoldSim Reliability Module. It presents the basic concepts of how risk and reliability 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 Reliability Module (page 15).

3. **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 Reliability Module.

4. **Open and explore the example files**. When you install GoldSim, a folder labeled "Reliability 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 Reliability Module model files. These examples are introduced and discussed in Chapter 7. These example model files are an excellent way to begin to experiment with the Reliability Module.

   Read more: Chapter 7: Example Reliability Module Applications (page 145).

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 Reliability 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 Guide 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, using

the index and table of contents as your guide. Each section of the User's Guides is 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. **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:

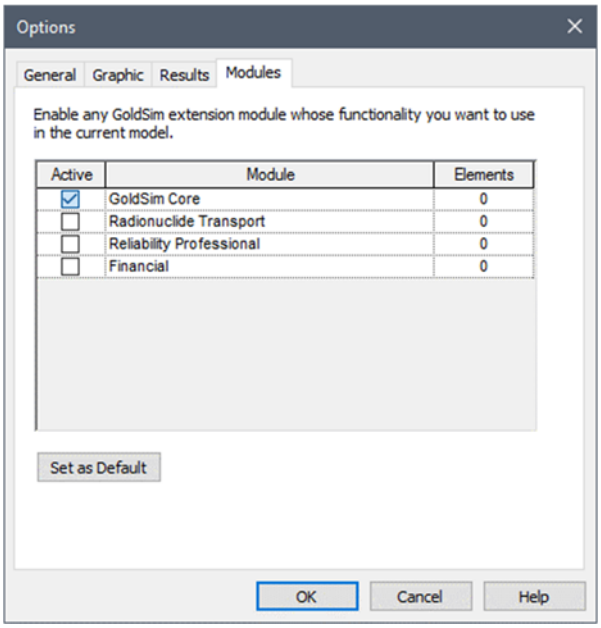
Convention	Description
Important Terms	New and important terms are presented in *bold italics*. These terms all appear in the Glossary of Terms at the end of the document.
File | Open…	Menus and menu selections are separated by a vertical bar. File | Open… means "Access the File menu and choose Open"
Failure Modes	Dialog buttons and tabs are identified in **Bold** font.
CTRL+C	Key combinations are shown using a "+" sign.. 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.

Activating the Reliability Module

In order to access the features and capabilities of the Reliability (RL) Module, either RL Learning Edition or RL Professional must be activated on your machine. Your copy of GoldSim will have one or the other (but not both). RL Learning Edition is automatically included with GoldSim Pro. RL Professional must be purchased separately.

Read more: What is the Difference Between Reliability Professional and Reliability Learning Edition? (page 6).

You can determine which version of the Reliability Module you have and whether ti is activated on your machine by selecting **Model| Options…** from the main menu, and selecting the **Modules** tab:



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

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.

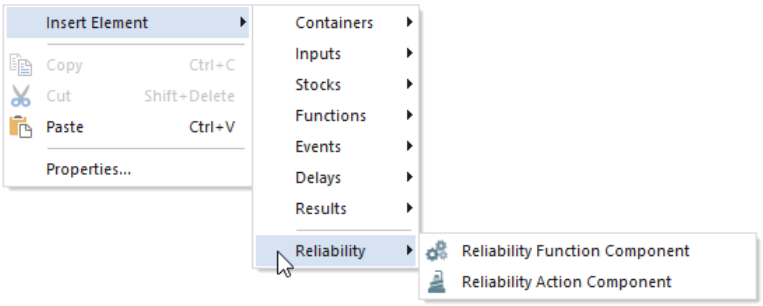
If you deactivate a module (such as the Reliability Module), any specialized elements associated with that module will be deleted (if any are present) and any menu options will be removed in the current file. If you make a module active, the various options associated with that module are made available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

Note: If you try to open a file that contains more than 10 reliability elements, and you are licensed to use Reliability Professional but it is not currently active, GoldSim will automatically activate the module and open the file. If, however, you are not licensed to use Reliability Profession, GoldSim will not be able to open the file (and will display an error message).

What Does the Reliability Module Add to the GoldSim User Interface?

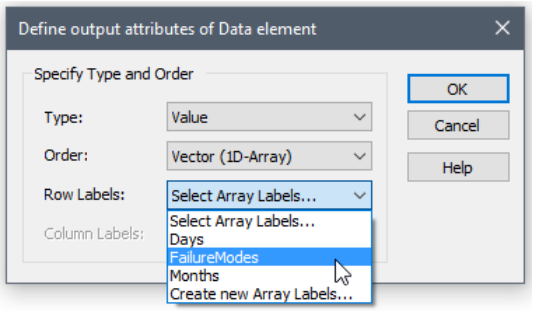
The Reliability Module is a program extension to the GoldSim simulation framework. As such, it simply adds some additional features to the user interface. The user interface components which the Reliability Module adds can be summarized as follows:

- An additional item, **Reliability**, is added to the context-sensitive (right-click) menu in the graphics pane (shown below), to the context-sensitive menu for Containers, and to the **Model**| **Insert** menu on the menu bar:



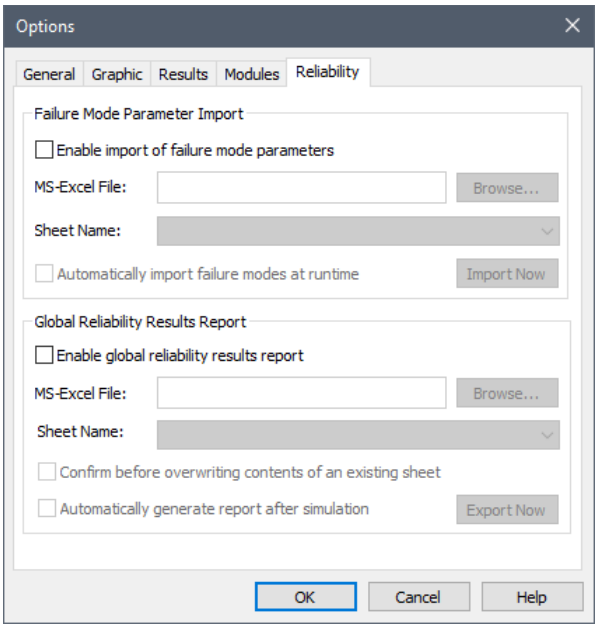
- Pressing the RL Element button in the Element toolbar provides access to a menu for inserting a RL Module element:

- The *FailureModes* array label set is added to the drop-down lists available when defining the output attributes for an element:



  This set of array labels is indexed from 1 to 10 (and cannot be edited), and is used to reference failure modes.

- A **Reliability** tab is added to the Options dialog (accessed via **Model | Options**) for defining options which are specific to the Reliability Module:



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 phone. 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.

Details regarding the GoldSim Maintenance Program can be found at www.goldsim.com/Web/Products/BuyGoldSim/Pricing/MaintenanceProgram/.

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** (https://goldsim.zendesk.com) is the primary portal for technical support. You can submit your questions directly from the Help Center. If your register and log in through the Help Center, you will be able check the status and view a history of all of your support requests.

The Help Center also includes:

- The **GoldSim Forum**, where you can 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.

Details regarding Advanced Technical Support can be found at www.goldsim.com/Web/Resources/TechnicalSupport/.

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):

- A free **Online Training Course** that 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.
- 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 through these resources:

- The GoldSim LinkedIn Group, which is primarily used for announcements (e.g., new versions, interesting applications). You can join the Group here: www.linkedin.com/groups/1798413)
- Periodic email newsletters are sent two to three times per year. To be added to the newsletter list, contact us via the GoldSim Help Center (https://goldsim.zendesk.com).

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.

References

The references cited in this chapter are listed below:

Ebeling, C.E, 1997, An Introduction to Reliability and Maintainability Engineering, McGraw-Hill, Boston.

Stamatelatos, M. et al., 2011, Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners, Second Edition, NASA/SP-2011-3421, NASA Headquarters, Washington, D.C.

Vesely, W. et al., 2002, Fault Tree Handbook with Aerospace Applications, NASA Office of Safety and Mission Assurance.

Chapter 2: Getting Started with the Reliability Module

> A 10 cent fuse will protect itself by destroying the $2000 radio to which it is attached.
>
> Robert Livingston, *Flying the Aeronca*

Chapter Overview

This chapter provides a brief overview of some key concepts of the GoldSim simulation framework, provides an overview of the features and capabilities of the Reliability 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 Reliability 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 those portions of the manual where you can obtain further information.

Note: While this chapter provides an introduction to the features of the Reliability Module, the technical details are spelled out in Chapters 3 through 6.

In this Chapter

This chapter discusses the following:

- Basic GoldSim Concepts Necessary to Use the Reliability Module
- Overview of the GoldSim Approach to Risk and Reliability Modeling
- A Simple Reliability Module Example

Basic GoldSim Concepts Necessary to Use the Reliability Module

The Reliability Module is a program extension to the GoldSim simulation framework. In order to take full advantage of all of the powerful features of the Reliability 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 Reliability 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 Reliability 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

The concepts are summarized below. If you are not comfortable with these concepts, you should return to the Tutorial and/or consult the portions of the **GoldSim User's Guide** or Help file that discuss these concepts (and are cross referenced in each of the three summary sections listed below).

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 and randomness by specifying some inputs as probability distributions or random events.
- 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 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 Reliability 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 Reliability Module, it is discussed further here.

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

However, to fully utilize all the features of GoldSim and the Reliability Module, you will eventually want to learn more about the details of discrete event modeling, and can do so by consulting the portions of the **GoldSim User's Guide** or Help file that discuss these concepts.

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 Reliability Module because the systems being modeling are controlled primarily by discrete occurrences (such as failures and repairs).

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 or a Pool) which understand how to process it.

Within the Reliability Module, discrete event modeling is a fundamental feature of the elements used to model reliability and risk, and is utilized in the following ways:

- Reliability elements (which are used to model system components such as pumps, engines, relays, and switches) can be triggered to turn on and off. They can also be triggered to be replaced or repaired.
- One class of Reliability elements (Action elements) are used to model components which must respond to a control command or condition (e.g., switches, relays). Events provide the command or condition that tells the element to carry out its action.
- Reliability elements output discrete event signals to mark a number of occurrences, such as whether or not a requested action was successful and when a component starts or stops operating. These events can then be used to subsequently trigger other elements.

Basic GoldSim Elements that Support Modeling of Events

In addition to elements within the Reliability 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 reliability elements to represent complex risk and reliability models.

For example, the outputs of these elements can trigger reliability elements to turn on or off. In other cases, discrete event signals from reliability elements may trigger them to perform a particular action (set a Status element to True or False).

Some of the event elements that are likely to be most useful to risk and reliability 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.
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.

Note that elements such as the Triggered and Timed Events are often used to create initiating events (e.g., events external to the system being modeled, such as an earthquake or a random interruption) that can lead to the failure of a system component.

Defining Triggers

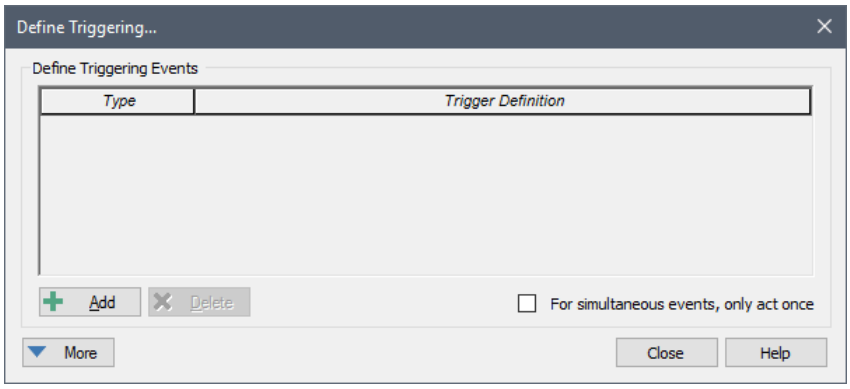
All discrete event modeling involves *triggering* of one form or another. When you trigger an element, you are telling it that a discrete event has occurred that you want the element to respond to.

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:

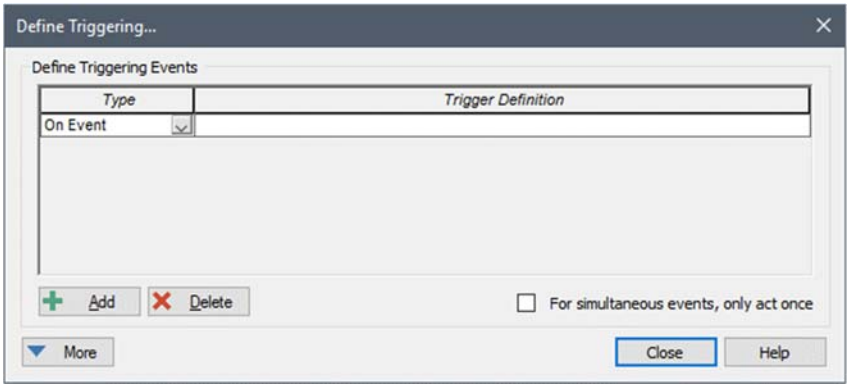
Note: Depending on the element, the label for the trigger button will not always be "Trigger…".

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

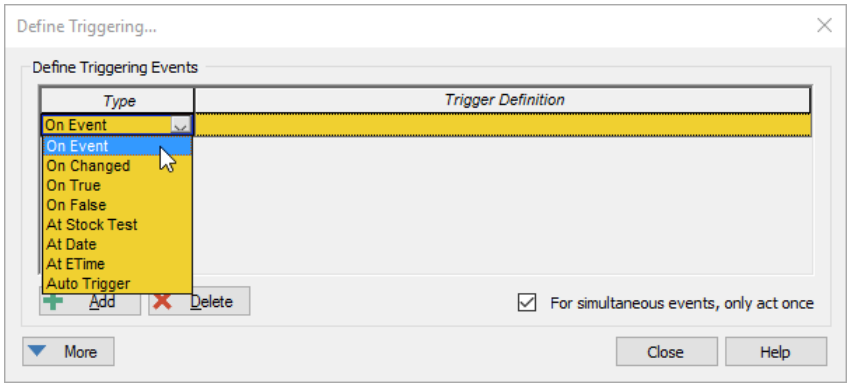
The Triggering dialog for every element looks like this:



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 of the form "A>B", "A>= B", "A<B", "A<=B", or "A=B" where A is the primary output of a Reservoir, a Pool or a Fund (from the Financial Module). 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 >=17days. 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.

Overview of the GoldSim Approach to Risk and Reliability Modeling

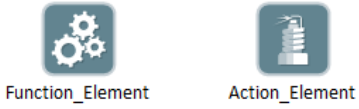
The Reliability Module uses an approach that is substantially different from most current approaches to investigating system reliability and risk. This section is intended to provide you with a broad overview of the GoldSim reliability modeling and risk analysis approach.

The goal of most reliability and engineering risk analysis models is to establish the lifespan, reliability, degree of availability, or probability of failure of a system. But how do we model a system's performance in GoldSim? Below we introduce the GoldSim approach to reliability and risk modeling by considering how one might model the reliability of a personal computer.

The Reliability Elements

The first step to modeling reliability in GoldSim, as it is in any other reliability and risk analysis 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 reliability elements themselves.

GoldSim provides two types of reliability elements: the Function element and the Action element:



Function elements are used to model components which operate continuously once turned on. Typical examples of components modeled by Function elements include pumps and engines.

Action elements are typically used to represent components which must respond to a control command or condition. Typical examples of components modeled by Action elements include switches and relays.

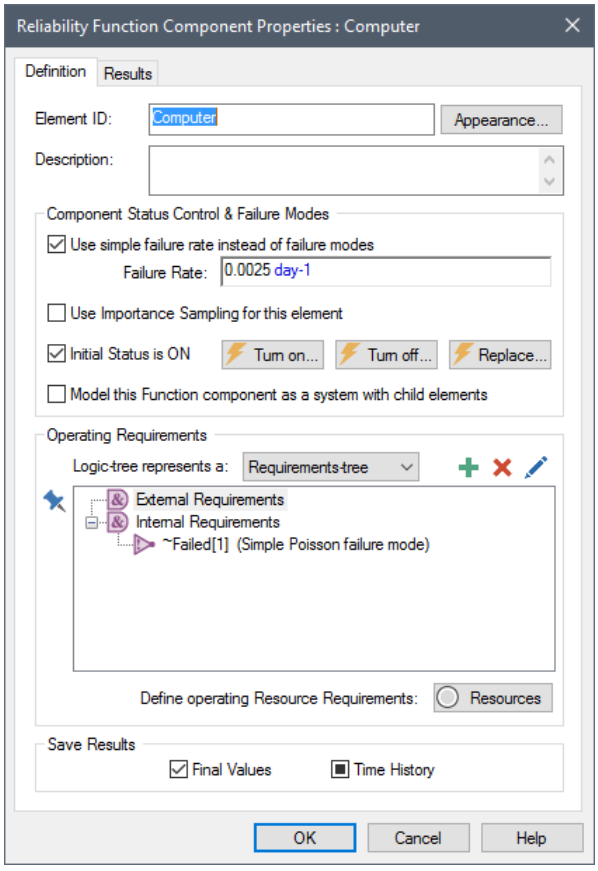
Both element types can fail, as well as be repaired and maintained. In addition, the Action element has the ability to determine whether or not a requested action has been completed in a satisfactory manner.

Read more: Chapter 3: The Reliability Elements (page 49).

Top-Down Modeling Using the Reliability Module

GoldSim encourages a top-down approach to modeling. In a top-down approach, the system is initially represented in a simple manner, and then evolves (and becomes more detailed and realistic) as more information about the system is obtained. A top-down approach to modeling is inherently an iterative approach. To illustrate this, let's consider an example of modeling the reliability of a computer.

Following a top-down approach, our first attempt at modeling the reliability of the computer might be very simple. In particular, we might begin by using a single Function element to represent the computer, and assign an exponential (constant) failure rate to model the computer's reliability. The Function element's dialog for this simple representation is shown below:



In this particular case, we have specified that the reliability of the computer can be modeled by assuming that it has an exponential failure rate, with a mean of 1/400 days (the **Failure Rate** input field is set to 0.0025 day-1, or 1/400 days).

A quick look at the dialog reveals a number of options and fields which can be used to incorporate additional detail and complexity into the model as the system becomes better quantified. Once a preliminary model has been developed and run, users will typically develop an improved, more realistic model using some of the more advanced features of the Reliability Module.

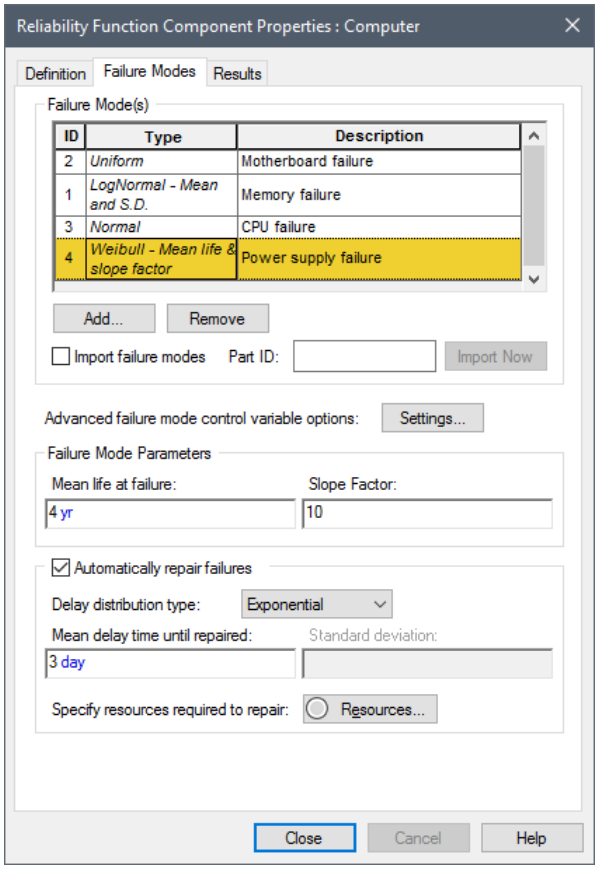
We will illustrate some of these advanced features below as we add some detail to our reliability model for the computer.

Read more: Overview of the Function Element (page 50).

Adding Failure Modes to a Reliability Element

There are a number of ways that a computer could fail. Hence, lumping these various failure modes into a single exponential rate is surely a simplistic approach. Furthermore, our simple model does not represent repairs. If our objective is to predict the availability of the computer, we need to simulate repairs too.

To facilitate this, GoldSim allows us to move beyond a simple exponential failure rate, and specify multiple distinct and independent failure modes. In the dialog below, we have specified five separate failure modes for the computer:



Each failure mode represents a particular type of failure (e.g., failure of the power supply), and each type of failure has its own distribution, which does not have to be exponential. For example, the Power Supply failure mode uses a Weibull distribution (a common failure distribution for reliability models).

Each failure mode can also be specified to be automatically repaired (with the time until it is repaired being different depending on the failure mode). In the example above, if the computer fails due to the Power Supply failure mode, it is assumed to be repaired in approximately 3 days (repair time is modeled using a distribution, and is therefore variable).

It is important to note that the ***failure mode control variable*** (i.e., the variable that is referenced by the failure mode to determine when failure occurs) for failure modes does not have to be absolute or operational time. In the case of a car you might build a model that uses the car's mileage to determine when

failure occurs, while in the case of an airplane, you might base failure modes on the number of takeoffs and landings. In these cases, other elements in your GoldSim model would be used to model the cumulative mileage or cumulative number of takeoffs and landings. In fact, this is one of the most powerful features of the Reliability Module; because it is part of a general-purpose simulation framework it can incorporate many factors into the analysis that other approaches cannot.

Similarly, failure mode parameters do not have to be constant during the course of the simulation. You might change the failure mode parameters to reflect the current state of the system, (e.g., you could specify that a failure of a car's rear braking system increases the wear on its front braking system), or to reflect the effects of the operating environment (e.g., a computer's failure rates could be influenced by the temperature and humidity of the room).

Note: By default, failures are assumed to be fatal to the component. That is, if a particular failure mode occurs, the component itself fails and is no longer operative. However, if desired, you can specify that a failure mode is non-fatal in order to model more complex failure mode behavior.

Read more: Defining Failure Modes (page 92); Modeling Coupled and Non-Fatal Failure Modes (page 105).

Modeling Hierarchical Systems of Components

Although modeling a computer using multiple failure modes is an improvement over using a single exponential failure mode, perhaps we determine that modeling something as complex as a computer requires us to model the individual components of the computer separately.

Because GoldSim and the Reliability Module are hierarchical by nature, we can quickly transform our simple single element model into a system with a number of subcomponents. Again, the intention is to allow our model to evolve (and hopefully improve) as the system becomes better quantified.

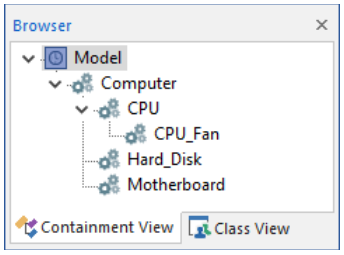
In this case, we can convert the model of the computer into a system with child (sub) elements. (On the main page of the Function dialog, there is an option to **Model this Function component as a system with child elements**). When we do this, there is a slight change in the appearance of the Computer element:

A small red triangle now appears in the upper left-hand corner of the element, and clicking on this takes you into a new layer of the model *within the computer*. If you are familiar with GoldSim, you will recognize this triangle as the same symbol that normally appears on GoldSim Container elements. In fact, when we convert a reliability element into a system, the element retains all of its properties, but now also functions as a Container. (In fact, it actually becomes a ***localized Container***).

Note: Localized Containers are discussed in detail in Chapter 10 of the **GoldSim User's Guide**.

In our example, we might choose to place a CPU, a Motherboard and a Hard Disk inside the Computer element, and model each of these components in as much detail as we wish:

Note, however, we are not limited to one level of hierarchy in GoldSim. In fact, we can create an unlimited number of subcontainers (subcomponents). In the figure below, a browser view of the model is provided, showing the hierarchy of components. In this case, a Function element representing a CPU Fan has been placed inside the CPU Function element, adding another level to the model hierarchy:



Of course, each of these subcomponents can have its own failure modes.

What are the benefits of using hierarchical modeling in order to model a reliability system? The most obvious is that it makes the model much more intuitive and easy to browse. If we use hierarchical modeling, we have just one element in the top level of the model for the computer, where we would otherwise have five. While this may be a subtle distinction with just one computer, it is a feature which becomes very valuable when one needs to model a bank of 50 computer servers.

In addition, due to the logical relationships that exist between components that are actually hierarchical in the real world (e.g., a computer contains a CPU which contains a CPU fan), it may not be possible to represent some kinds of systems in a realistic or straightforward manner without defining such a hierarchy. This is discussed further below.

Read more: Modeling a Reliability Element as a System with Child Elements (page 58).

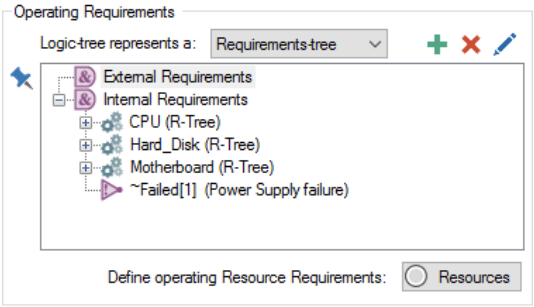
Representing Logical Relationships Between Components

In nearly all multi-component systems, logical relationships will exist between the components (e.g., "in order for component A to operate, component B or component C must be operating"). These relationships can be represented within the reliability elements by defining either a ***requirements-tree*** or a ***fault-tree***. In a requirements-tree, the top level node must evaluate to true for the component to operate, and for a fault-tree, the top level node must evaluate to false for the component to operate. These trees are used to model the logical relationships between components, and to propagate the effects of changes in element status to affected components.

In the case of the Computer element, it requires an operational CPU, motherboard and hard disk in order to function properly. In the case of the CPU, it needs its fan to be operational in order to function properly. Of course, each of these subcomponents can have its own set of failure modes. Let's assume for the purpose of this example that all of the failure modes for the Computer (except one) are assigned to the subcomponents. One failure mode

(Power Supply Failure) is assigned to the parent element (the Computer), since we have decided not to explicitly represent the power supply as another element.

GoldSim can easily represent these relationships, but it provides a slightly different approach to fault trees than many reliability engineers will be used to. In particular, there is no need to develop a "global" requirements- or fault- tree for the entire system. The user is only required to define the immediate relationships between components. In this case, the computer's requirements tree would include four nodes, indicating that in order for the Computer to operate, the CPU, Hard Disk and Motherboard must all be operating, and the Computer itself must not have failed due to the Power Supply failure mode:



Note that these are listed as "Internal Requirements" because the components and the failure mode are internal to the Computer. Failure modes for the Computer itself are added automatically to this list by GoldSim. Any sub-components need to be added manually (as appropriate).

"External Requirements" can be used to specify requirements external to the Computer (e.g., that electricity is available).

Read more: Failure Modes and Internal Requirements (page 95); Defining Operating Requirements for Reliability Elements Using Logic Trees (page 64).

Using GoldSim's Probabilistic Simulation Engine

Once a model of the Computer 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 when a component will fail; we can only describe the failure (and repair) process statistically. For example, if we had 100 identical computers, their failure (and repair) histories would not be identical. They would display a distribution of behaviors.

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 not carried out actual tests on the components, the parameters describing their failure modes 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 a calculating a large number of "realizations" (potential futures). In our computer example, this would be equivalent to simulating the behavior of a large number of computers through time.

Read more: Using Monte Carlo Simulation in Your Reliability Model (page 84).

Viewing and Analyzing Results

GoldSim provides a number of reliability analysis tools that become available at the end of a simulation.

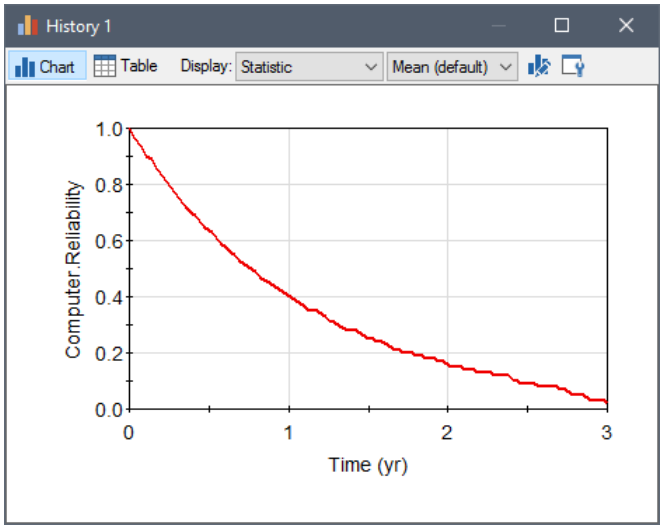
The most fundamental result is the summary of the mean reliability and availability statistics over the duration of the simulation:

Summary

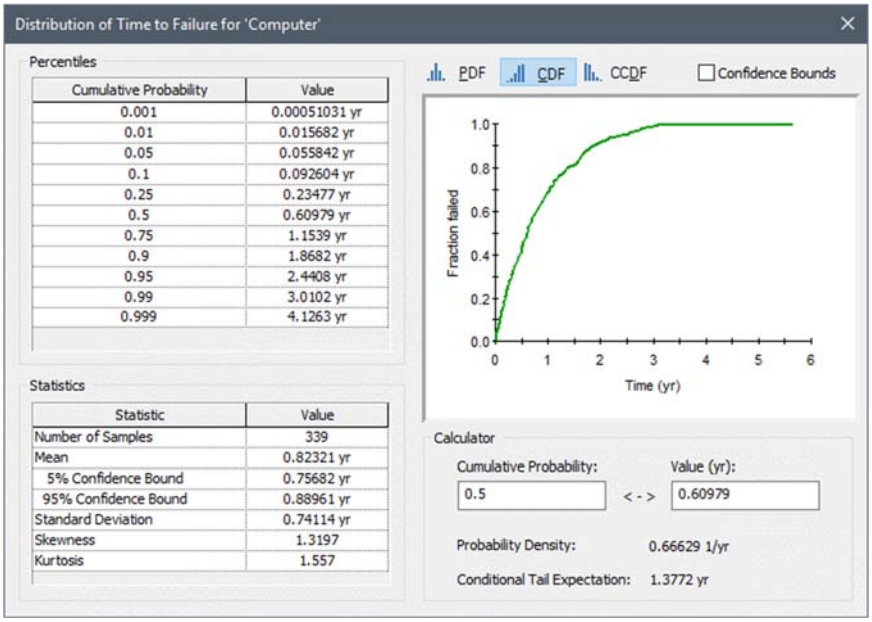
Results for 100 realizations with mean duration of 3 yr.

Measure	Confidence Bounds		
	5%	Mean	95%
Operational Availability:	0.930	0.943	0.956
Inherent Availability:	0.930	0.943	0.956
Reliability:	0.004	0.020	0.047

A wealth of additional information is also available. For example, a time history of the mean reliability can be displayed:

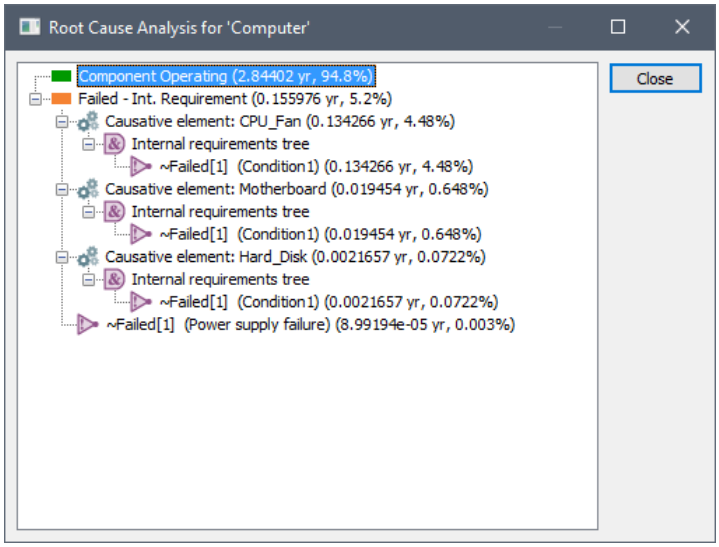


Also, statistical analyses of failure and repair time distributions can be displayed:

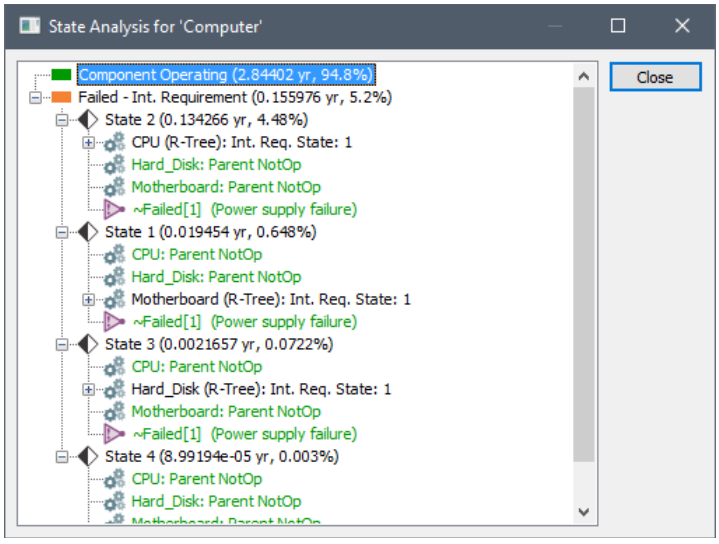


One of the most powerful types of result processing available within the Reliability Module is ***causal analysis.*** Because GoldSim records all of the unique conditions which result in failure of each individual element during a

simulation, it can provide analyses that identify failure scenarios and failure root causes for each reliability element:



A related analysis shows the different failure scenarios for a system, where each scenario corresponds to a specific state of the system's components. Each state is presented as a logic-tree, with each component that is not operating shown highlighted (in red):



Read more: Chapter 6: Displaying Reliability Results (page 129).

Documenting Your Reliability 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.

As discussed previously, 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 Reliability Module Example

Having provided a conceptual overview of the Reliability Module, this section provides a more hands-on introduction to the same concepts by walking through the steps necessary to build and edit a simple reliability model. Although it is not necessary to do so, it will be helpful if you actually follow along and build and edit the simple model as it is discussed below.

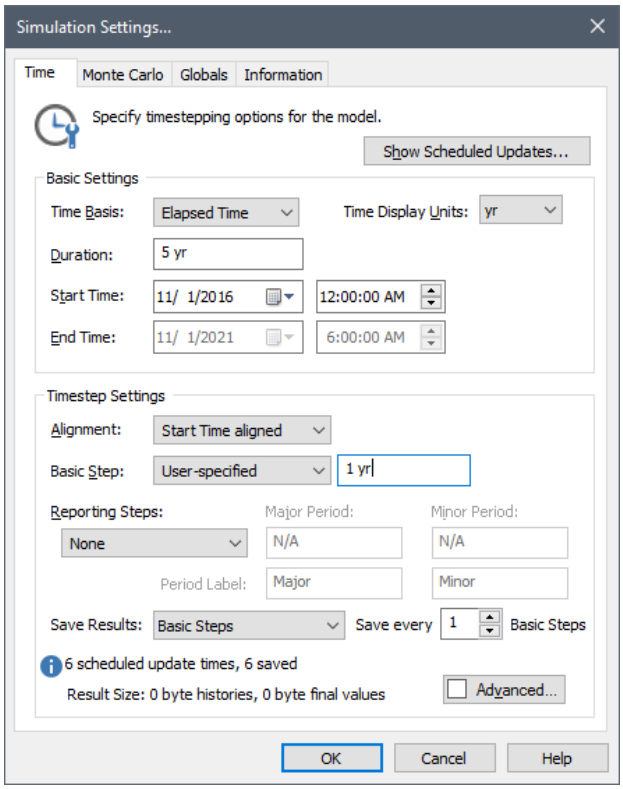
In order to do so, however, you must be able to carry out the basic GoldSim actions discussed in the GoldSim Tutorial (e.g., adding elements, modifying simulation settings, running a model). Therefore, if you wish to build and edit the model described below, and you have not already taken the Tutorial, 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: Creating a Dynamic Reliability Model

One of the key distinctions between traditional reliability approaches and the GoldSim reliability approach is the way that time is treated. In many traditional reliability approaches, such as closed-form reliability equations, time is nothing more than a variable in the equation representing the service life of the component. GoldSim is quite different in that it is a dynamic simulator, meaning that events can occur, conditions can change and components can interact as time progresses over the course of the simulation.

Let's explore this concept by constructing a simple GoldSim model. 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**).

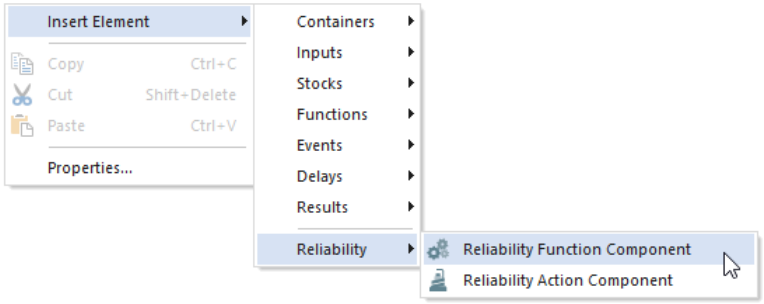
Let's start by specifying the simulation settings (by pressing **F2**). When specifying the simulation settings, we will specify that the simulation will be an Elapsed Time simulation with a duration of five years (5 yr), using a 1 yr **Basic Step**. Define the **Time Display Units** in years (yr):



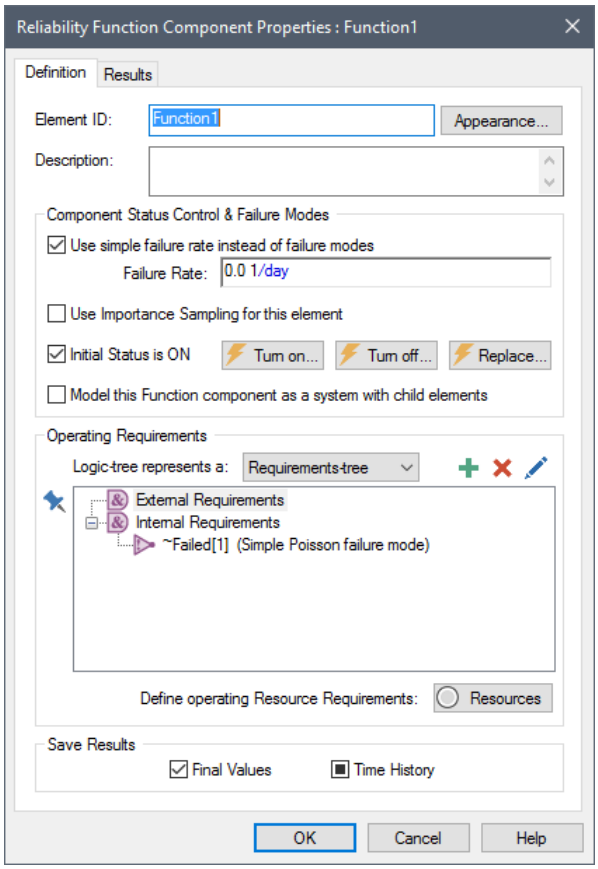
At this point, we do not need to edit the **Monte Carlo** tab, as we will start by running a single realization (the default).

Step 2: Adding a Reliability Function Element

Let's start by insert a new Function reliability element into the new model. We can do this by right-clicking in the graphics pane, and selecting **Insert Element|Reliability|Function Component** from the context menu:



This will insert the element and open the Function element's Properties dialog:

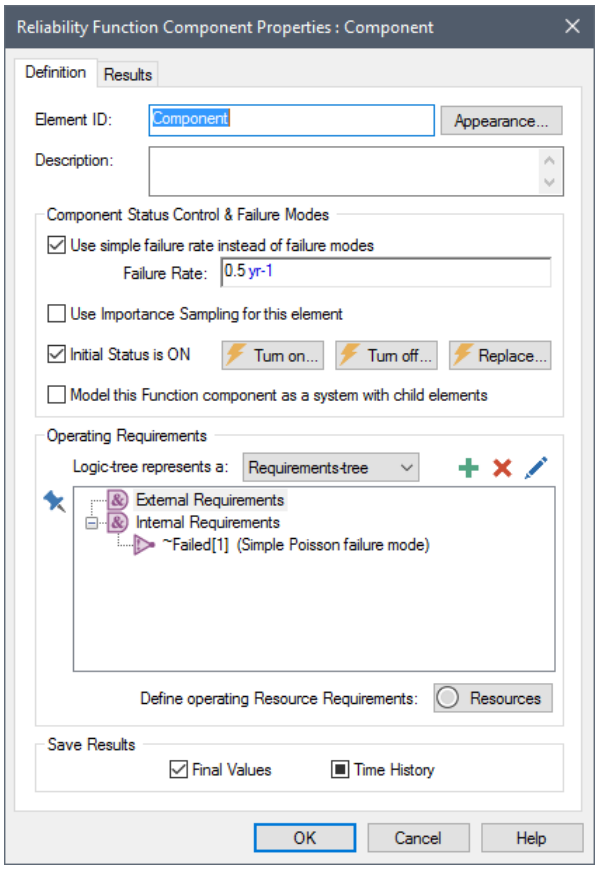


If this is your first close look at a reliability element, you will notice that it has some common features with the other basic GoldSim elements that you've seen in the GoldSim Tutorial. The **Element ID** and **Description** fields are identical (and the same naming rules apply to reliability elements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

Below the **Description** field, there is a section called "Component Status Control & Failure Modes". This sections allows you to define the failure behavior for the component (whether you use a simple, unrepaired exponential failure mode, or more advanced failure modeling), and buttons that allow you to set up triggering logic to turn the component on and off, or replace the component with a new one. It also contains an option to control whether or not Importance Sampling is used (a technique that can be used to facilitate representation of failure modes that occur very infrequently).

At the bottom of the dialog, there is an "Operating Requirements" section that allows you to specify logical relationships and conditions for the successful operation of the element. Note that it automatically includes an Internal Requirement that the default failure mode (Simple Poisson failure mode) has not occurred.

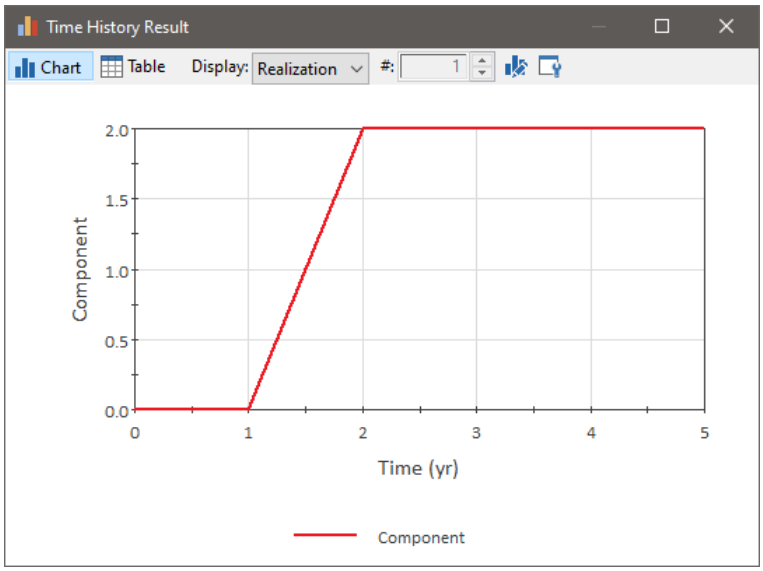
Now that we've taken a quick look at the Function element, let's change the **Element ID** (simply call it "Component"), and specify a simple failure rate of 0.5 yr-1 (equivalent to a mean time to failure of 2 years):



We now have a component with a mean failure time of 2 years, and a service life (the duration of the simulation) equal to five years. 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 Function 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 may look different, since this represents one random realization of the system.

What are we looking at here? This is actually the status of the Component over the course of the simulation. 0 indicates that the component is operating, and 2 indicates that the component has failed due to unmet Internal Requirements (in this case, the occurrence of the Simple Poisson failure mode).

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:

1. 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 years, 1 year, 2 years, 3 years, 4 years and 5 years (as we have five timesteps). Values for time history plots are recorded at these user specified timesteps.
2. GoldSim also updates the model upon the occurrence of certain kinds of "events". The failure of a reliability element is such an event. Hence, in this case, if the Component failed at 1.57 years, GoldSim would insert a new timestep at this point and update the model.

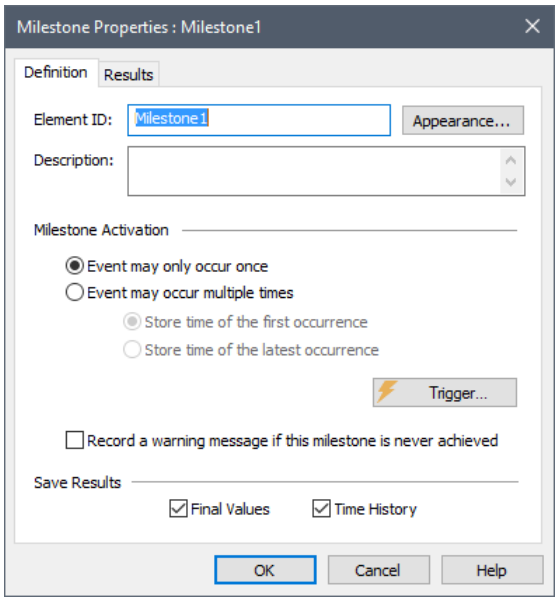
Note, however, that although the elements in the model are updated when such an event occurred, values for time history plots are only recorded at the user specified timesteps (in this case, 0, 1, 2, 3 ,4 and 5 years).

Given this information, we now know that the result shown above indicates that in this particular random realization (again, yours may be different), the Component failed sometime between 1 and 2 years. In the next step, we will determine exactly when the Component failed.

Step 4: Determining the Time of Failure Using a Milestone Element

In order to determine exactly when the Component failed, we will add a Milestone element to the model. A Milestone element is an element that is part of the basic GoldSim framework. The element is quite simple. In its simplest implementation, it is triggered by an event, and simply records the time when it was triggered.

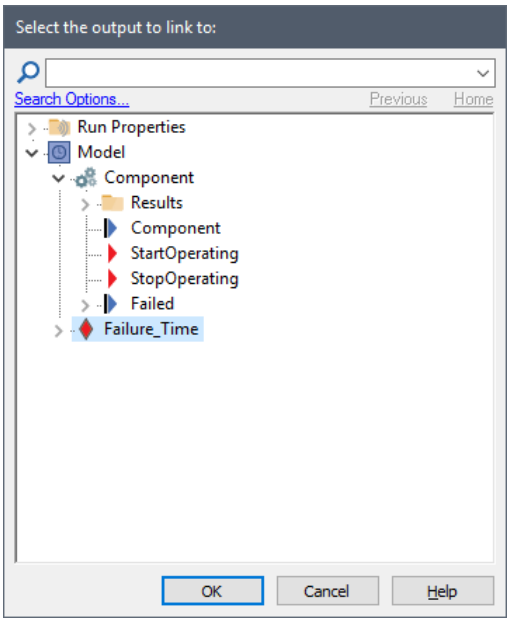
To insert the Milestone element, we need to return to Edit Mode by pressing **F4**. We can then insert the element by right-clicking in the graphics pane, and selecting **Insert Element|Events|Milestone** from the context menu. A Milestone element will be inserted and its property dialog will be displayed:



Change the element's name to "Failure_Time". Since the Component can only fail once in this simple model, the default Milestone Activation settings do not need to be changed.

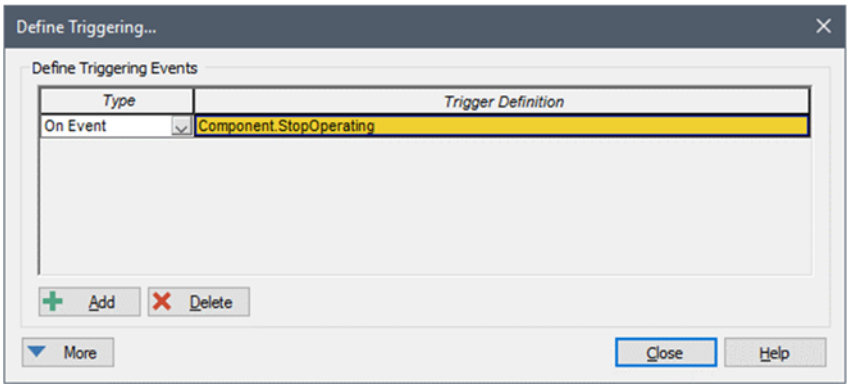
We now need to trigger the Milestone when the Component stops operating. This can be done as follows:

1. Press the **Trigger…** button.
2. Press the **Add** button in the Trigger dialog to add a trigger.
3. The default Type will be On Event. This can be left unchanged. To specify the **Trigger Definition**, right-click in the field and select **Insert Link** from the context menu.
4. Expand the Component element by clicking on the > sign. The following will be shown:



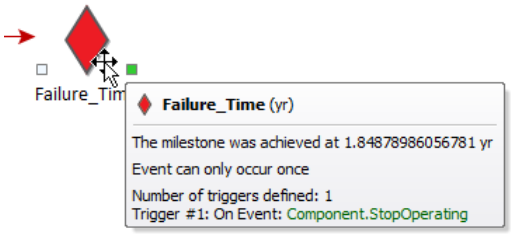
5. Double-click on the StopOperating output. This is an event that is emitted by the element when the Component stops operating.

At the end of this process, the Trigger dialog should look like this:



Close the Trigger dialog and the Milestone's Properties dialog and run the model again (**F5**).

After you rerun the model, you can mouse over the Milestone element to see the time that the Component actually failed. In the example shown below, you can see that it failed at approximately 1.85 years:



Note: As pointed out previously, your result will be different, since this represents one random realization of the system

Step 5: Increasing the Level of Time Discretization

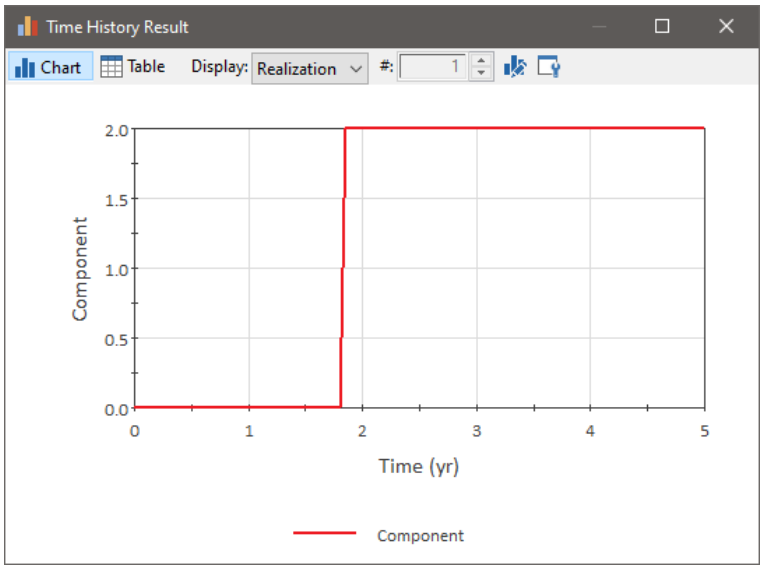
Let's add some additional timesteps and rerun the model to see how the time history plot changes.

To do this, you'll need to return to Edit Mode by pressing **F4**. Then, open the Simulation Settings dialog (by pressing **F2**) and change the **Basic Step** length to 0.05 yr.

We can now close the Simulation Settings dialog and rerun the model (**F5**).

If you mouse over the Milestone element, you will notice that the time of its occurrence hasn't changed from when you looked at them the last time your ran the model (since by default, random numbers generated to produce the events are repeatable).

However, the time history plot will change significantly. We can see this by right-clicking on the Function element and then left-clicking on **Time History Result…** in the context menu to view a time history plot of the result:



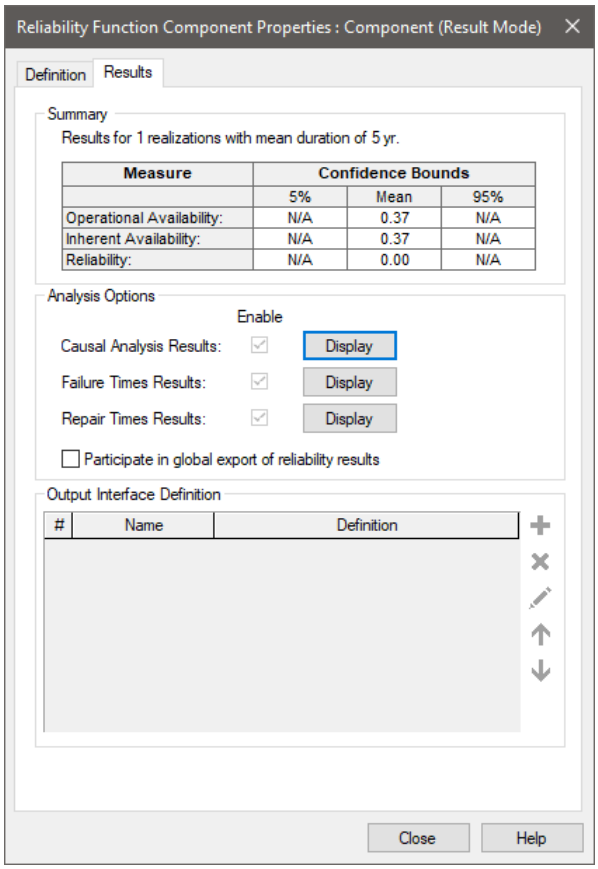
Notice how the change in the component's status from 0 to 2 is now seen to indeed occur around 1.85 years due to the additional timesteps.

Note: As pointed out previously, your result plot will be different, since this represents one random realization of the system.

Step 6: Computing Reliability and Availability

Although simple time histories of failure for a single realization may be of some interest, what we are really interested in doing is computing the reliability and availability of the Component.

We can view these results by opening the Component's property dialog and viewing the **Results** tab:



Note: As pointed out previously, your result will be different, since this represents one random realization of the system

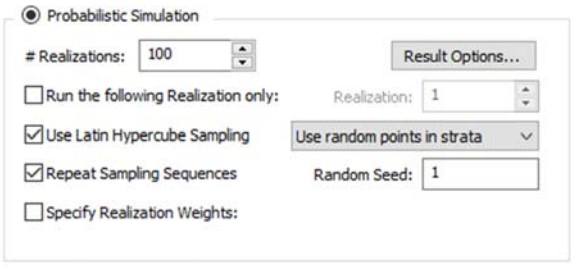
The results shown above show an Availability of 0.37 and a Reliability of 0. What does this mean? The problem is that these metrics cannot be accurately determined by running a single realization of the model. The Availability is simply being computed here as the time the Component was operating (in this case about 1.85 years) divided by the total simulation time (5 years). The Reliability is zero because the Component failed in this particular realization.

We cannot draw accurate statistical conclusions with a sample size of one. In order to discern the expected behavior of the system, and the boundaries of that behavior, we need to either run multiple realizations or, if the system was being repaired, run a single realization for a long time so that it models multiple failure and repair cycles. In the next step, we will do the former.

Step 7: Running Multiple Realizations of a Reliability Model

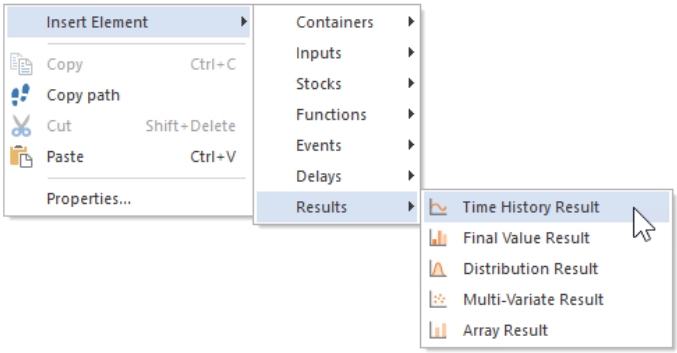
In order to discern the expected reliability and availability of the system, and the range of that behavior, we will run multiple realizations of the system using Monte Carlo simulation.

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 make sure the **Probabilistic Simulation** radio button is selected. Then change the **# Realizations** to 100:

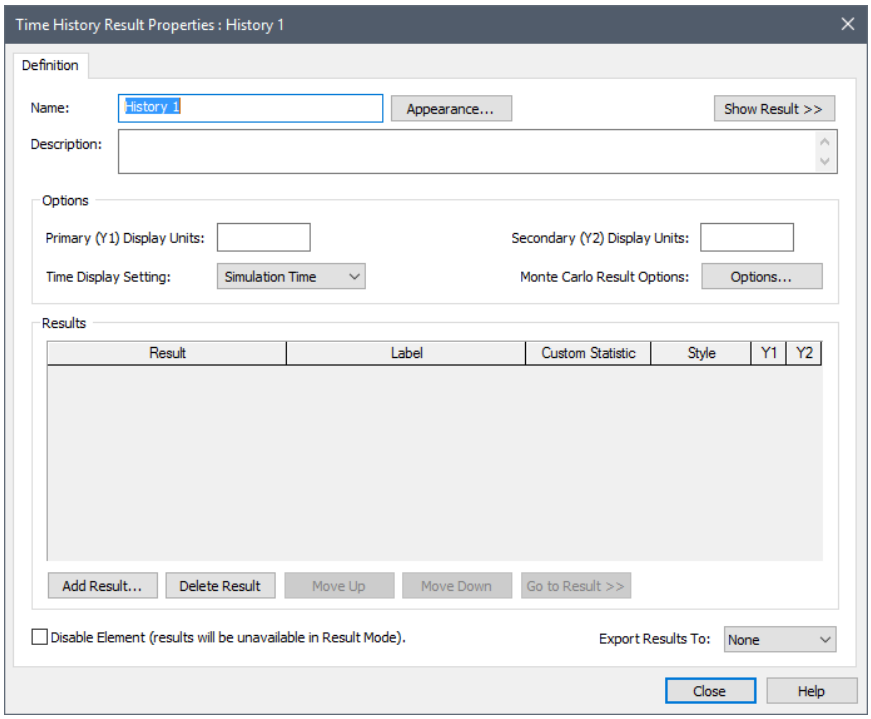


Close the Simulation Settings dialog.

When running multiple realizations, in order to view time history results, you must connect the output of interest to a Time History Result element. To do so, right-click in the graphics pane and insert a Time History Result element:



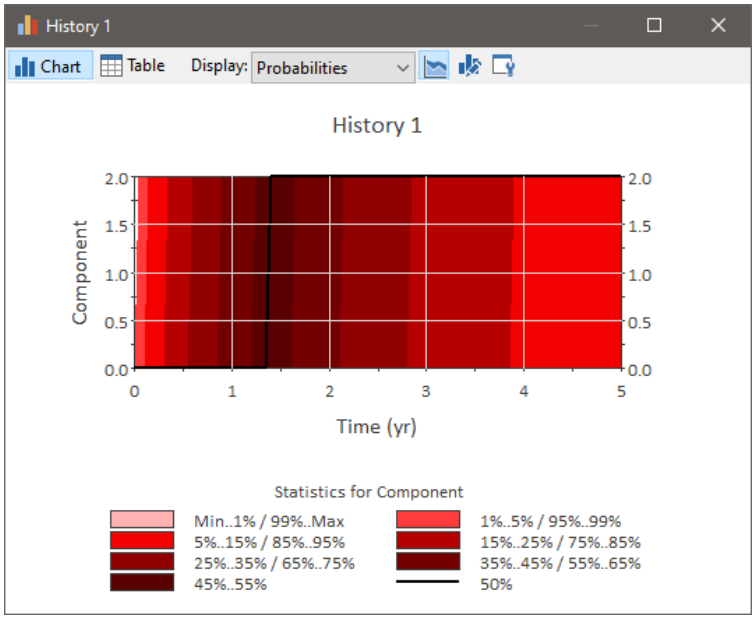
The following dialog will appear:



Press the **Add Result…** button and select Component. Then close the dialog and run the model (by pressing **F5**).

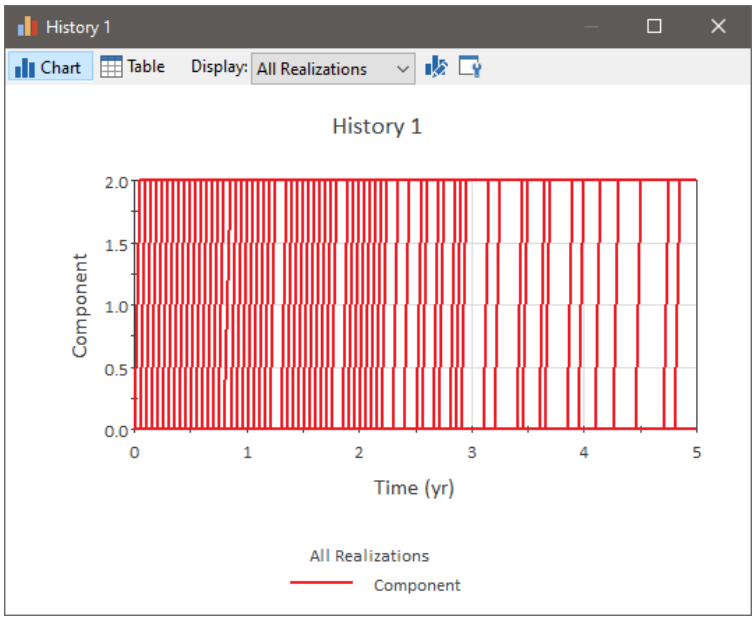
Step 8: Viewing Monte Carlo Results for a Reliability Model

After running our simple Monte Carlo simulation of the Component, we can view the result 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:



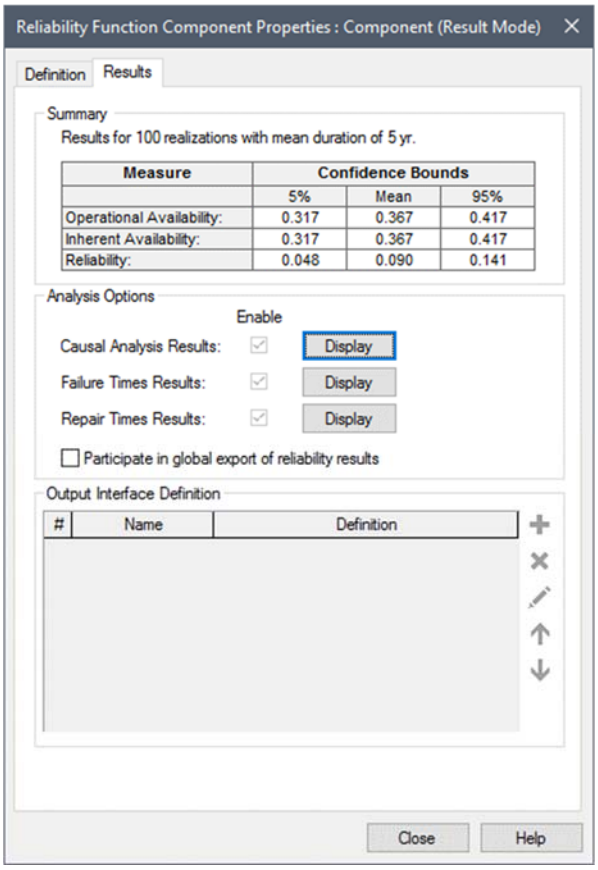
When you initially view a time history plot of multiple realizations, GoldSim displays a probability history view, showing you the median (black line), and percentiles on the Component's status over time. In this example, the various histories are simply vertical lines, hence the percentile boundaries are vertical lines.

Things become a bit clearer if we switch from displaying Probabilities" to displaying "All Realizations" (by selecting this option from the **Display** drop-list at the top fo the chart)::



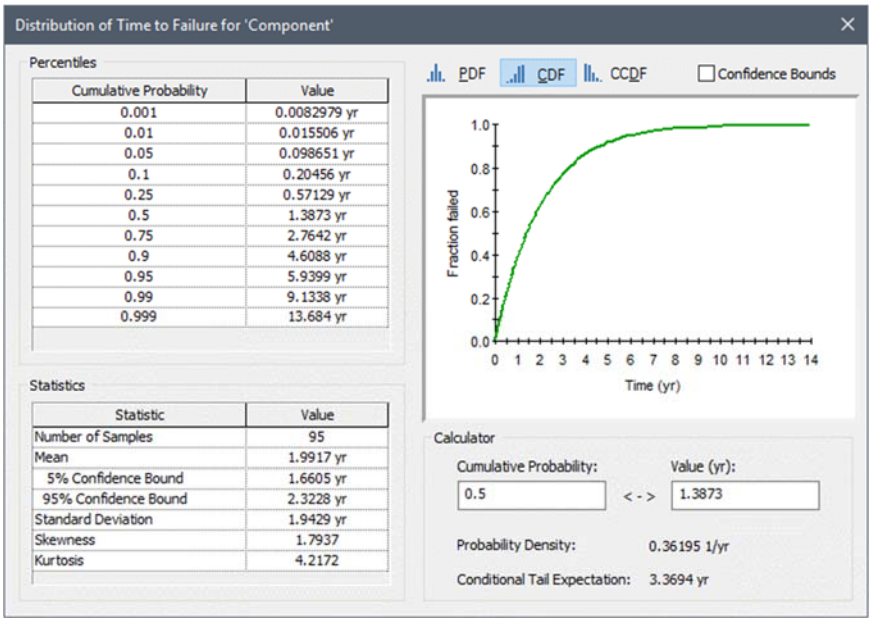
Each line represents a different realization (with the Component failing at different times). Statistical results on the expected behaviour of a system are produced by combining the results of these realizations.

We can see these statistical results by viewing the **Results** tab of the Component element:



Notice that the reliability value is now nonzero (i.e., there is a small, but nonzero probability that the Component will operate for the full five year period). GoldSim displays Availability and Reliability with confidence bounds, Note that the range between the confidence bounds would shrink if we increased the number of realizations.

Now click on the **Display** button to the right of **Failure Times Results** label. This will display a distribution of the time of failure for the Component:

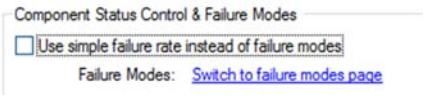


Step 9: Editing Failure Modes and Adding Automatic Repair

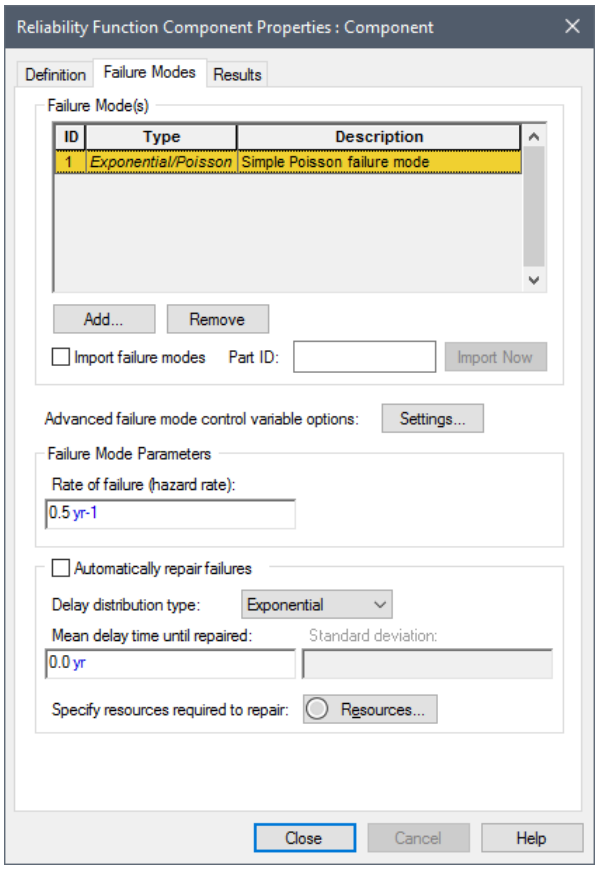
Let's add some additional detail to the Component's behaviour by adding failure modes and repair.

By default, the Function element allows you to specify a simple exponential failure mode that is not repaired. However, in many cases, the component's behaviour cannot be accurately modeled in such a simple way. Moreover, many high value systems do not "fail" permanently – they are repaired and returned to service.

To demonstrate this capability, let's return to Edit Mode (**F4**) and edit the failure modes for the Component. To do so, double-click on the Component element and clear the **Use simple failure rate instead of advanced failure modes** checkbox in the element's property dialog:

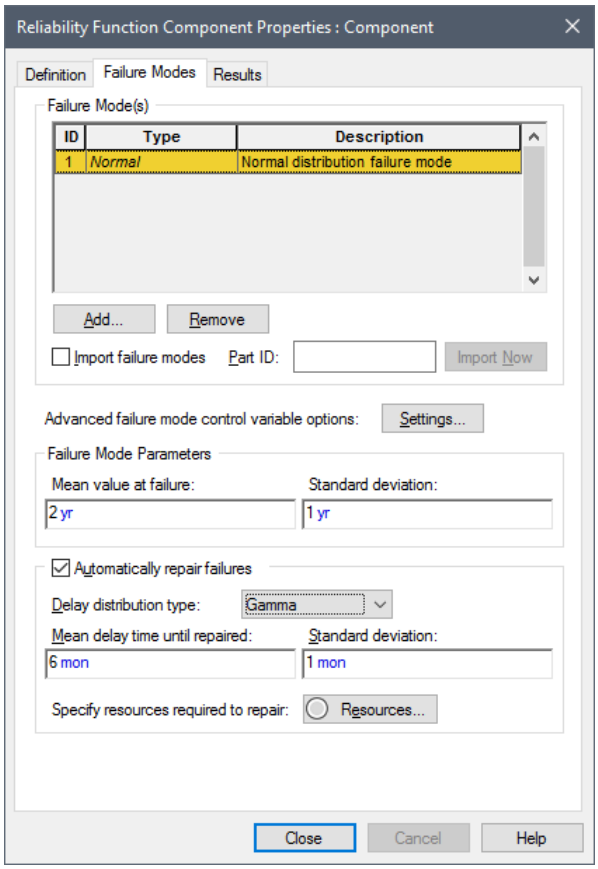


You'll notice that a new tab, called **Failure Modes** will appear. If you click on this tab you'll see the following dialog:



You'll notice that the initial exponential failure mode has been transferred over automatically by GoldSim. Let's delete this failure mode (press the **Remove** button), and add a Normal failure mode (by pressing the **Add** button and selecting "Normal"). Specify a mean of 2 yr and a standard deviation of 1 yr.

You'll also notice that there is an **Automatically repair failures** option. Check that option, and specify a repair with a Gamma delay distribution, a mean time to repair of 6 months and a standard deviation of one month (month is abbreviated as "mon" in GoldSim). The dialog should then look like this:

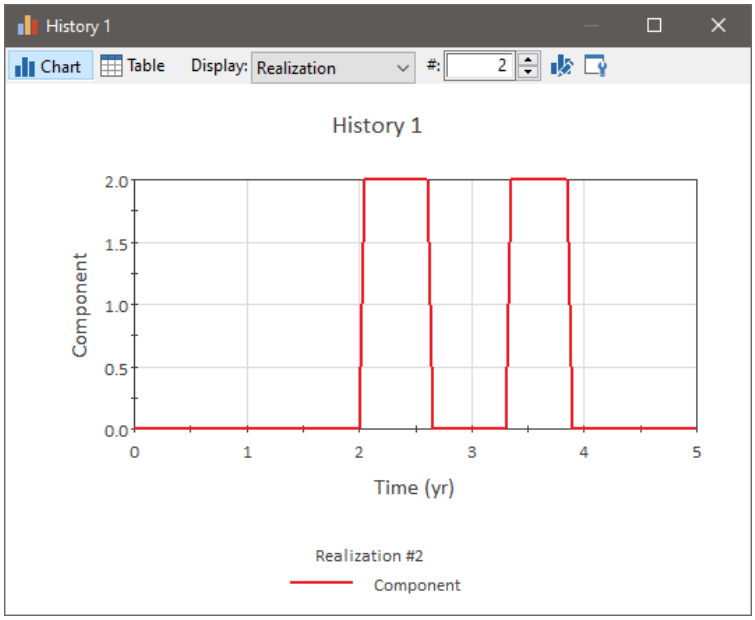


If we wanted to, we could add additional failure modes. For the sake of simplicity, however, we'll stick with one for now. Press **Close** to close this dialog.

Delete the Milestone element (as it will generate warnings since it will now be triggered multiple times each realization due to the repair of the failure mode), and re-run the model (**F5**).

Let's view a time history plot of the status of the Component by double-clicking on the Time History Result element. From the **Display** drop-list at the top of the chart, select "Realization". This shows one realization at a time. You can toggle through realizations using the **Realization** spin control at the top of the window.

A typical result should look something like this:



As can be seen, whenever the Component fails, it is repaired about 6 months later. If we look at the **Results** tab for the Component, we will see that the Availability is relatively high (due to the repairs):

Summary

Results for 100 realizations with mean duration of 5 yr.

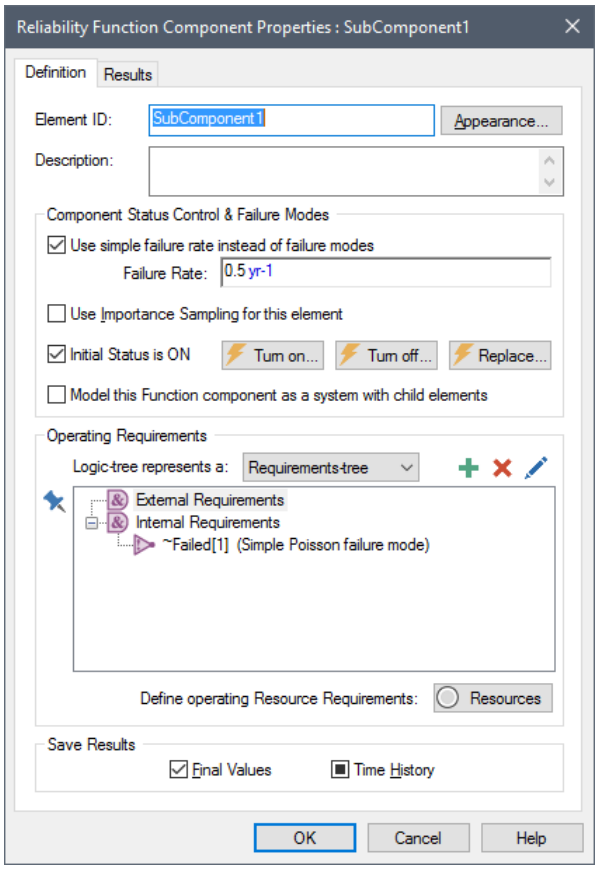
Measure	Confidence Bounds		
	5%	Mean	95%
Operational Availability:	0.820	0.831	0.841
Inherent Availability:	0.820	0.831	0.841
Reliability:	0.000	0.000	0.000

Step 10: Adding Hierarchy (Sub-Components) to a Reliability Model

Modeling the Component as a single object is the simplest way to represent the system. Since the Component actually consists of sub-components (each of which can fail in its own way), however, we may decide to model the system hierarchically. Because GoldSim and the Reliability Module are hierarchical by nature, we can quickly transform our simple single element model into a system with a number of subcomponents.

To convert the model of the Component into a system with child (sub) elements, return to Edit Mode (**F4**), open the Component's dialog and check the option to **Model this Function component as a system with child elements**. Then close the dialog. You will notice that there is a slight change in the appearance of the Component element. In particular, a small red triangle now appears in the upper left-hand corner of the element (the Component is now equivalent to a Container in GoldSim).

Clicking on the triangle takes you into a new layer of the model *within the Component*. After doing so, let's add three new Function elements (named SubComponent1, SubComponent2, and SubComponent3). Each one will be defined identically, with a failure rate of 0.5 yr-1:



Of course, we could have defined multiple failure modes and automatic repairs for each subcomponent, but for simplicity, we will use a simple failure rate here.

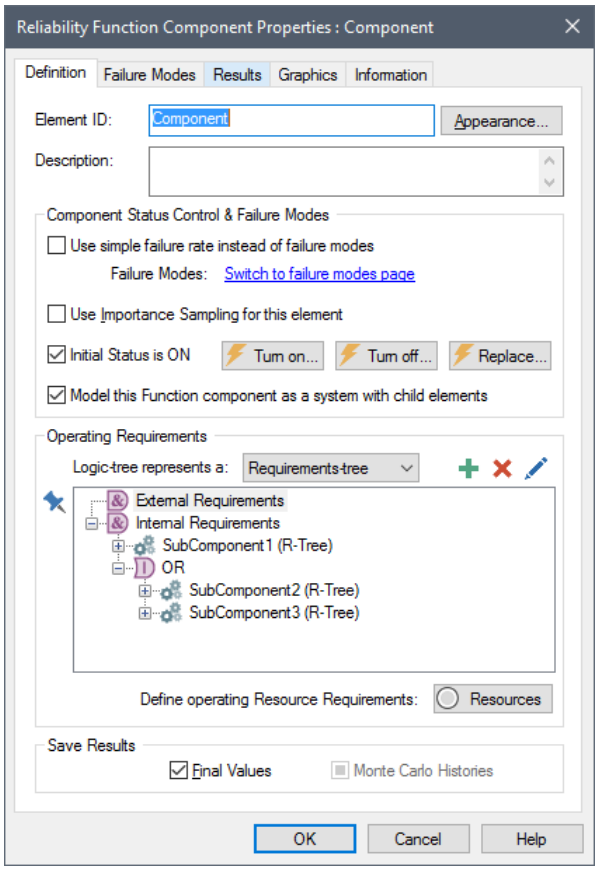
After creating these three new elements, move one level back up in the hierarchy, and open the Component element again. Go to the **Failure Mode** tab and delete the failure mode. That is, instead of specifying a failure mode for the Component, we want to specify that the Component fails when its subcomponents fail. We do this using the Operating Requirements section of the Component's dialog.

Let's assume that in order for the Component to operate, SubComponent1 must be operating and either SubComponent2 or SubComponent3 must be operating (SubComponent2 and SubComponent3 are redundant). We can do this by building an Internal Requirements tree for the Component as follows:

1. Left-click on Internal Requirements (which is an AND node).
2. Click on the "+" sign above the tree. This will display a menu to add a new node under the Internal Requirements.
3. Select "RL Component" from this menu. This will display a browser. Find SubComponent1 and double-click on it.
4. Click on the "+" sign again and Select "OR-Gate" from the menu.
5. The Or-Gate will be selected. Click on the "+" sign again and Select "RL Component" from the menu. This will display a browser. Find SubComponent2 and double-click on it.

6. Click on the "+" sign again and Select "RL Component" from the menu. This will display a browser. Find SubComponent3 and double-click on it.

At the end of this process, the dialog for the Component should look like this:



This indicates that the Component does not fail directly. Rather, it remains operable as long as SubCompenent1 has not failed and either Subcomponent2 or Subcomponent3 has not failed (i.e., it fails if SubComponent1 fails or both SubComponent2 and SubComponent3 fail).

If you wish, you can rerun the model now to see how this new representation of the system impacts the reliability.

Where Do I Go From Here?

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

The next four chapters (Chapters 3, 4, 5 and 6) provide detailed explanations of the features introduced here. The final chapter (Chapter 7) discusses a number of example models that illustrate how to represent a variety of systems. These example models are automatically installed with GoldSim.

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 four chapters describing the details of the Reliability Module to fully understand how the examples were constructed.