Reliability

可靠性 Confiabilidade

موثوقية Fiabilité Affidabilità

信頼性 Αξιοπιστία Zuverlässigkeit

Confiabilidad Niezawodność Надежность

BlockSim®

START

Reliability SYNTHESIS® Elements

Use with: BlockSim9_QuickStart_Rev2.rsgz9

Quick Start Guide

BlockSim

Version 9

ReliaSoft®

ReliaSoft Publishing
A Division of ReliaSoft Corporation

Tucson São Paulo Warsaw Chennai Singapore

BlockSim 9 Quick Start Guide
Part Identification: RPQSB9

ReliaSoft Corporation
Worldwide Headquarters
1450 South Eastside Loop
Tucson, Arizona 85710-6703, USA
Tel: 1-520-886-0410   Fax: 1-520-886-0399
Sales and Information: 1-888-886-0410 (Toll-free in the U.S. and Canada)
ReliaSoft@ReliaSoft.com   http://www.ReliaSoft.com

10 9 8 7 6 5 4 3

BlockSim 9 Quick Start Guide

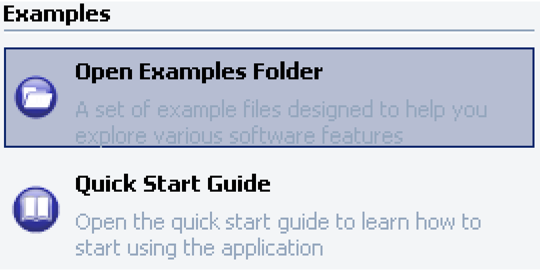
1

Thank you for your interest in ReliaSoft's BlockSim software tool. This Quick Start Guide has been designed to help you explore many of the software's key features by working through step-by-step instructions for some practical application examples.

In order to demonstrate a variety of different applications for the tools available in BlockSim, this guide asks you to imagine that you are a reliability engineer working for a computer company, then walks you through the steps you might take to answer a variety of questions about different systems. Please note that the sample data sets provided are fictional and intended for demonstration purposes. Furthermore, note that although this guide attempts to introduce you to some of the most frequently used tools in the software, BlockSim supports many other reliability engineering methods and applications. Within the software, you can choose **File > Help** to access a wide array of resources that will help you explore other software capabilities.

In addition to this introduction, the following chapters are presented in this guide.

The quick start repository that is installed with the software (called "BlockSim9_QuickStart_Rev2.rsgz9") contains a sample project with completed analyses for all of the examples. To access this file, choose **File > Help**, click **Open Examples Folder**, then browse for the file in the **BlockSim** sub-folder.



Tip: To preserve the integrity of the shipped example files, the software creates a copy of the file each time you access a repository in the Examples folder. The copy has the same name as the original file and is saved in the default documents folder for your computer (e.g., My Documents\ReliaSoft\Files). Use the copy to work on the example projects and save your changes. Any changes you make in the copy will not affect the original file.

When applicable, the instructions in this guide will refer to a completed folio in the Quick Start project using a notation such as (*"PC - Series Configuration" in the sample project*). While reading any example in this guide, you have the choice to:

- Examine how it was completed in the sample project.
- Follow the instructions to complete the analyses "from scratch."
- Copy data/analyses from the sample project to help you complete the analyses.

IMPORTANT: Note that it may sometimes be necessary to modify the data in the quick start repository to fit updated instructions or new examples in the latest printing of this quick start guide. This printing of the guide was designed for use with **BlockSim9_QuickStart_Rev2.rsgz9** (where _RevX indicates the database revision). If you try to use a different database revision, the sample projects may not exactly match the instructions printed here. 1) If this guide is older than the latest database revision installed on your computer, you can choose **File > Help > Quick Start Guide** to download the latest printing. 2) If this guide is newer than the latest database revision installed on your computer, you can choose **File > Help > Check for Update** to download the latest software service release.

Simple Analytical RBDs 2

A reliability block diagram (RBD) is a graphical representation of how the components of a system are reliability-wise connected. Each block in an RBD represents a component of the overall system or process that is represented by the RBD. You can define the individual blocks with the reliability and other characteristics of the components they represent.

In this chapter, you will perform system reliability analysis for a standard desktop computer using analytical RBDs. The objective of system reliability analysis is to create a model for the reliability of the system based on the known reliability models of lower level items.

In this chapter

- ✓ Reliability block diagrams (RBDs)
- ✓ Analytical diagrams
- ✓ Series configurations
- ✓ Parallel configurations
- ✓ *k-out-of-n* configurations
- ✓ QCP and plots

ReliaSoft.TV Watch a video of this example at http://www.reliasoft.tv/blocksim/startguide/blocksim_qsg_2.html.

2.1 Series Configuration

You are a reliability engineer, working for a computer manufacturer. This week, you are asked to determine the overall reliability model for a standard desktop computer based on the known reliability of its subsystems. For the sake of simplicity in this example, assume that the main subsystems are the hard drive, main board and power supply and that the computer will fail if any of its subsystems fails (i.e., series configuration). From prior analysis of testing and field data, you have the following reliability models for each of these subsystems:

- The hard drive's reliability follows a Weibull distribution with beta = 2 and eta = 70,000 hours.
- The main board's reliability follows a Weibull distribution with beta = 1.6 and eta = 92,000 hours.
- The power supply's reliability follows a lognormal distribution with log-mean = 10 hours and log-std = 0.7 hours.

Objectives

- Determine the MTTF or *mean life* of the system.
- Estimate the B10 life of the computer (i.e., the time by which 10% of computers of this type will be failed).
- Obtain a reliability plot, a *pdf* plot and a failure rate plot.
- Determine which of the components is most critical to reliability at the age of 1 year.

MTTF and MTBF: The terms MTBF (mean time between failures) and MTTF (mean time to failure) have often been interchangeably used to describe the average time to failure. The truth is, these two metrics are not the same and should not be used interchangeably.

When dealing with non-repairable components (as in the case of life data analysis) the metric sought is the mean time to failure or MTTF. It is only when dealing with repairable components (where the component may fail and be repaired multiple times during its operational life) that you calculate the mean time between failures or MTBF. The only time that the MTTF and MTBF are the same is when the failure rate is constant (i.e., when the underlying model is an exponential distribution).

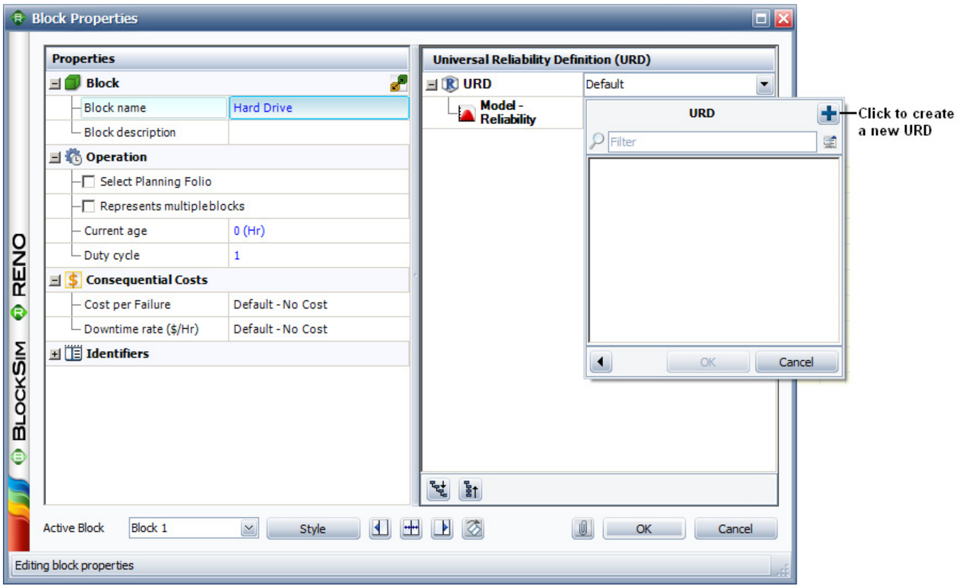
Solution

First, you create a new analytical RBD (*"PC - Series Configuration" in the sample project*) by choosing **Insert > Diagrams > Analytical RBD**.

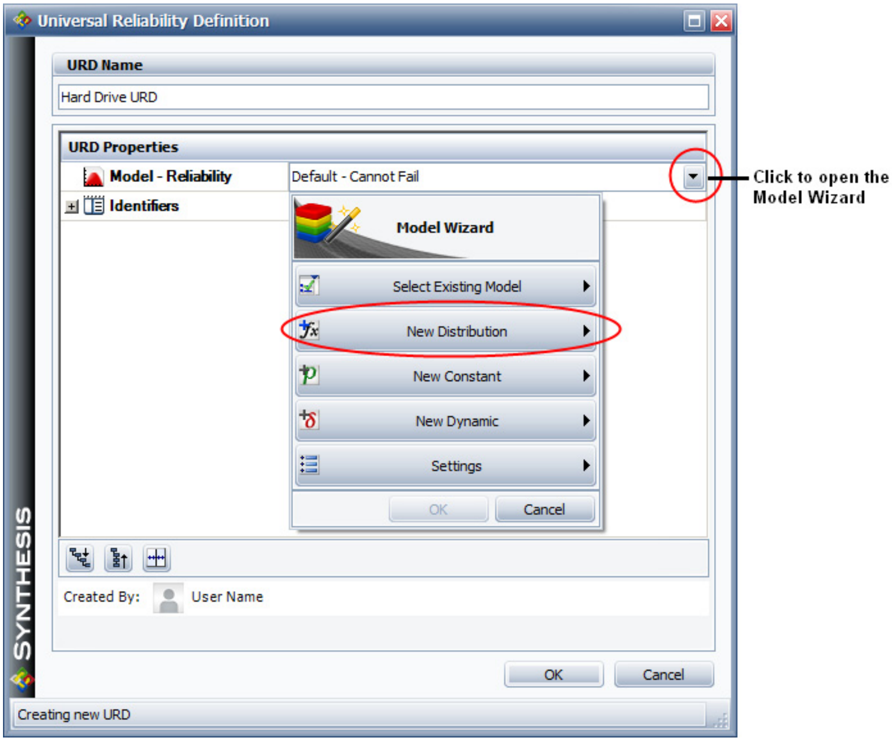
You add a block to the RBD by choosing **Diagram > Blocks > Add Block**.

This block will represent the hard drive. You double-click the block to open the Block Properties window and, in the **Block name** field, enter **Hard Drive**. To define the block's reliability properties, you create a new universal reliability definition (URD). URDs are resources that are used to represent a set of properties that can be applied to standard blocks in RBDs and to events in fault trees. These properties can include failure behavior and maintenance tasks.

To create a URD for the block, you click the **URD** field, click the arrow that appears in the field and then click the **Create New** icon in the URD Wizard that appears.

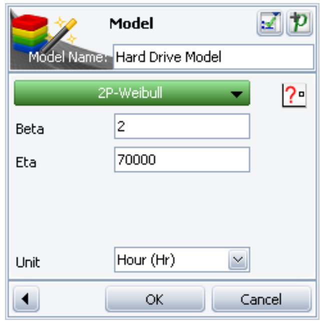


You name the URD **Hard Drive URD** and open the Model Wizard to create the model that defines the hard drive's reliability. In the Model Wizard, you click **New Distribution** to create a new time-varying model.



Tip: Once you have created a given category of model via the Model Wizard, your settings for that kind of model will be retained and shown the next time you create that same kind of model. For example, the first time you create a reliability model, you will choose whether to create a constant model or a distribution model. If you create a distribution model, then the next time you create a reliability model, you will automatically see the options for a distribution model. Therefore, if this is not the first time that you have used the Model Wizard, you may not see the Main page, as shown above. From any other page in the Model Wizard, simply click the New Distribution icon to go to the New Distribution page.

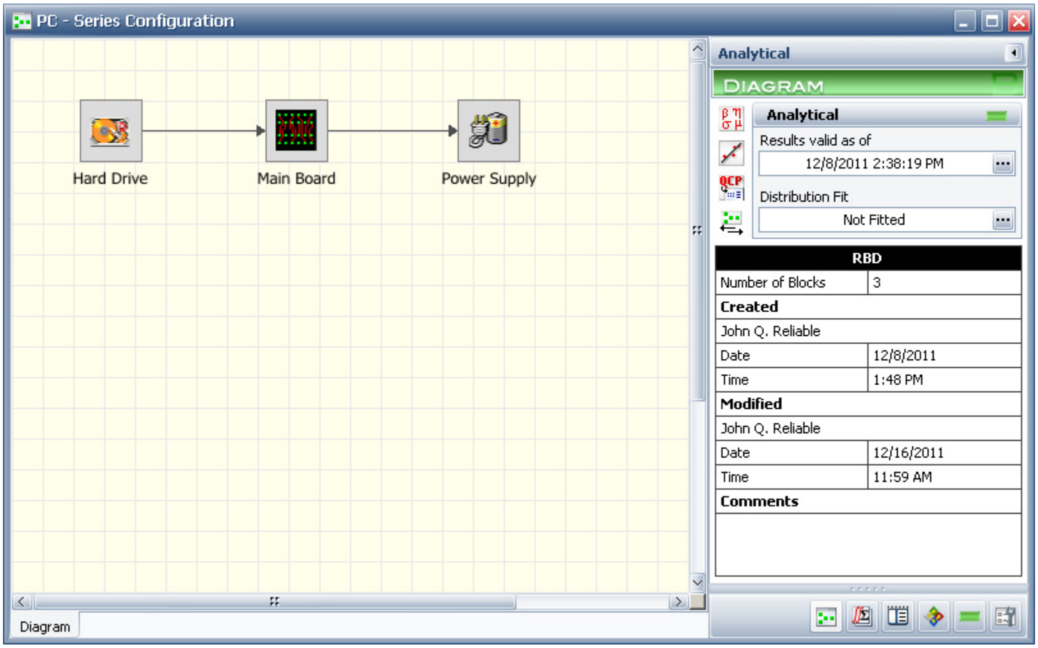
You name the model **Hard Drive Model**, select the **2P-Weibull** distribution from the drop-down list and specify a beta value of **2** and an eta value of **70,000**. You also use the **Unit** drop-down list to make sure that the eta value is in hours, as shown next.



You then follow similar steps to create the Main Board block, using the Main Board URD and the Main Board Model (2-parameter Weibull, beta = 1.6, eta = 92,000 hours), and the Power Supply block, using the Power Supply URD and the Power Supply Model (lognormal, log-mean = 10 hours, log-std = 0.7 hours).

Once you have added all the blocks to the diagram, you need to connect them. You enter the Connect Blocks mode by choosing **Diagram > Settings > Connect Blocks**.

You then connect the blocks in series by drawing a connector from the Hard Drive block to the Main Board block and another connector from the Main Board block to the Power Supply block, as shown next.

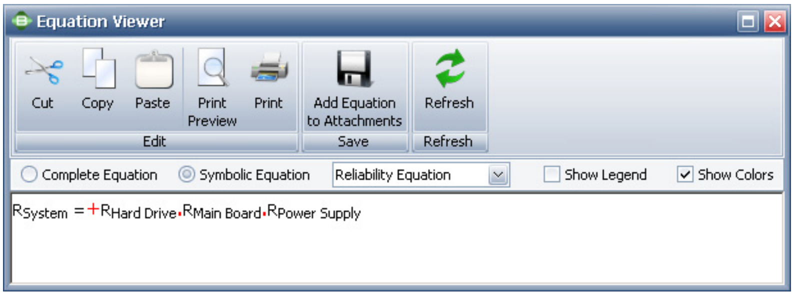


Note: The picture above shows the diagram in the sample project, which has had images added to the blocks. For more information about altering the appearance of blocks and/or diagrams, please see the online help at http://help.synthesis8.com/blocksim_reno9/index.htm#diagram_settings.htm.

You exit the Connect Blocks mode by choosing the command on the Ribbon again.

To analyze the diagram, you click the **Analyze** icon on the control panel.

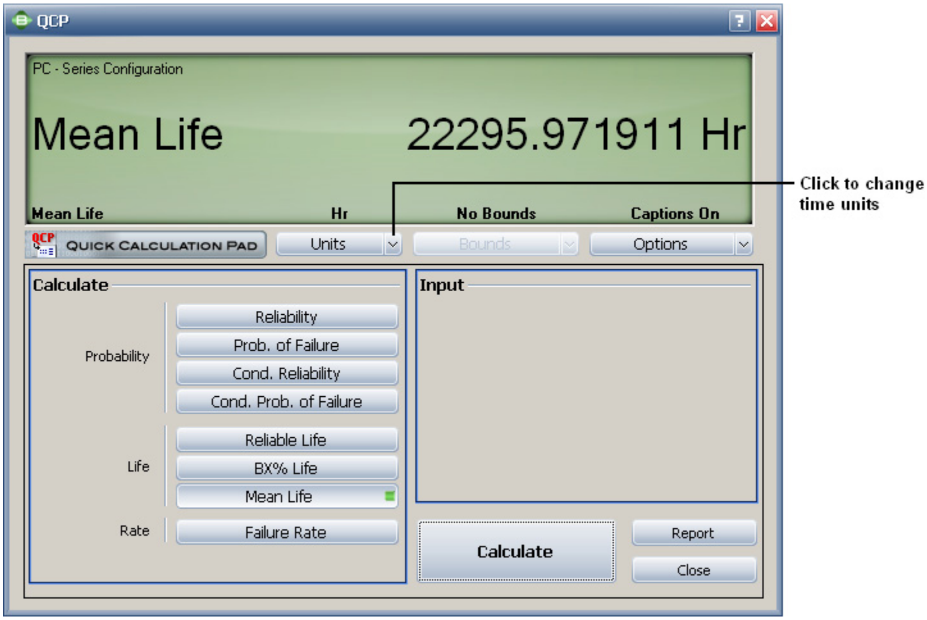
You then choose **Analysis > Tools > Show Algebraic Solution** to view the system equation in the Equation Viewer.



From this, you determine that the reliability of the system is equal to the product of the reliabilities of the components in this simple series configuration. Having determined the system reliability equation, you can now calculate the other metrics of interest.

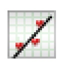
You open the analytical QCP by clicking the icon on the control panel.

You select to calculate the **Mean Life** and use the **Units** drop-down list to make sure the results will be returned in hours. The calculated result tells you that the average time to failure for the system is approximately 22,296 hours, as shown next.

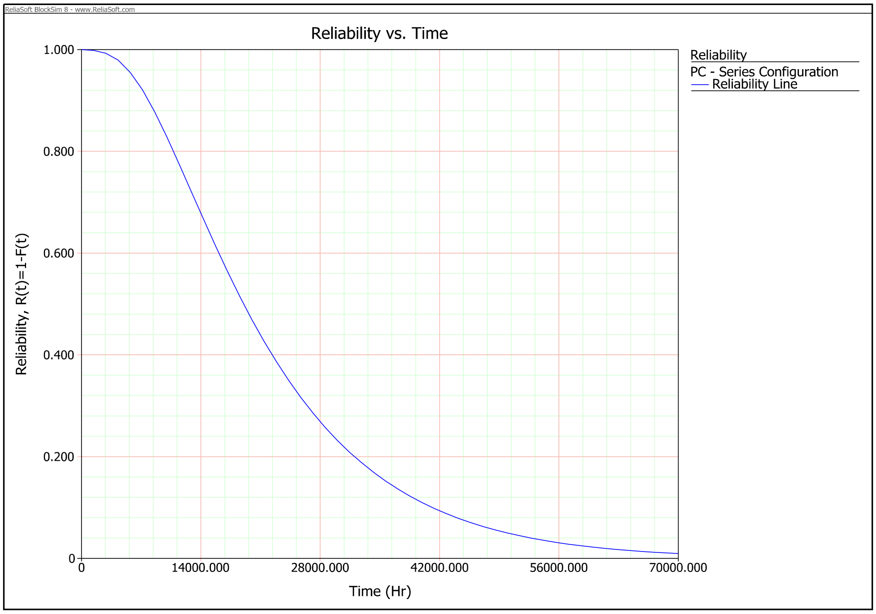


You then select **BX% Life** and enter **10** for the **BX% Life At** input. The calculated result tells you that 10% of the population are expected to fail by approximately 7,864 hours.

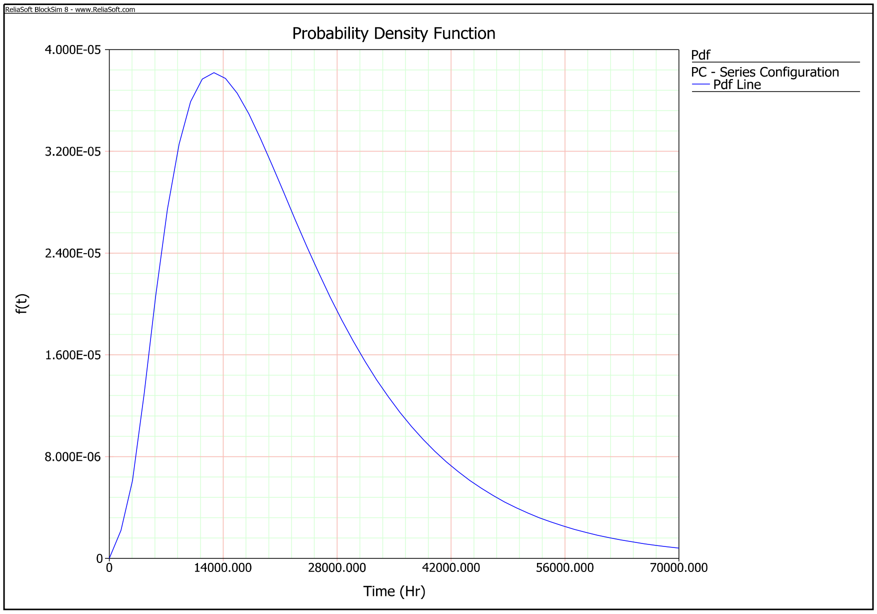
To view the plots of interest, you click the **Plot** icon on the control panel.

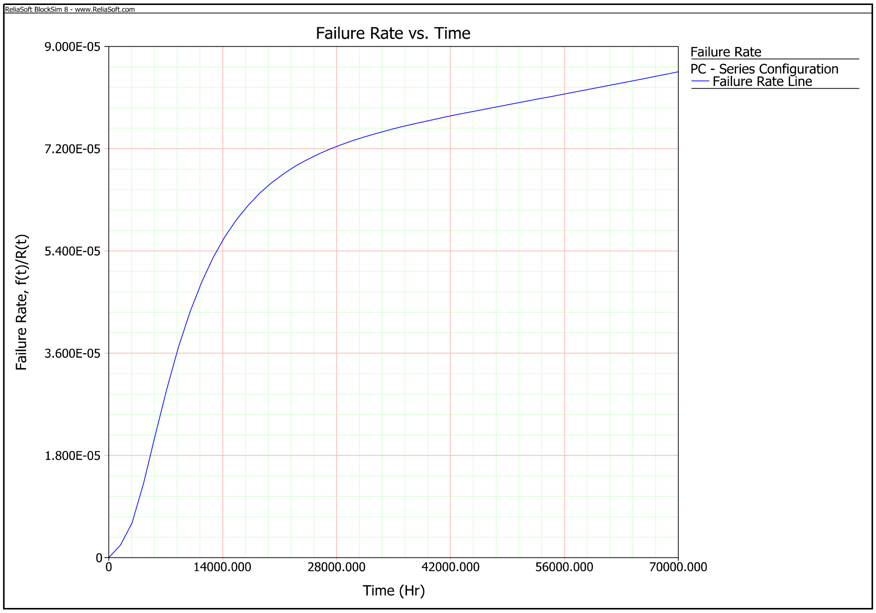


On the control panel of the plot sheet, you use the **Plot Type** drop-down list to choose the **Reliability vs. Time** plot. This displays the reliability of the computer over time. The plot shows a decreasing reliability, which generally indicates an underlying degradation mechanism with time. In other words, the older the computer, the less reliable it will be.

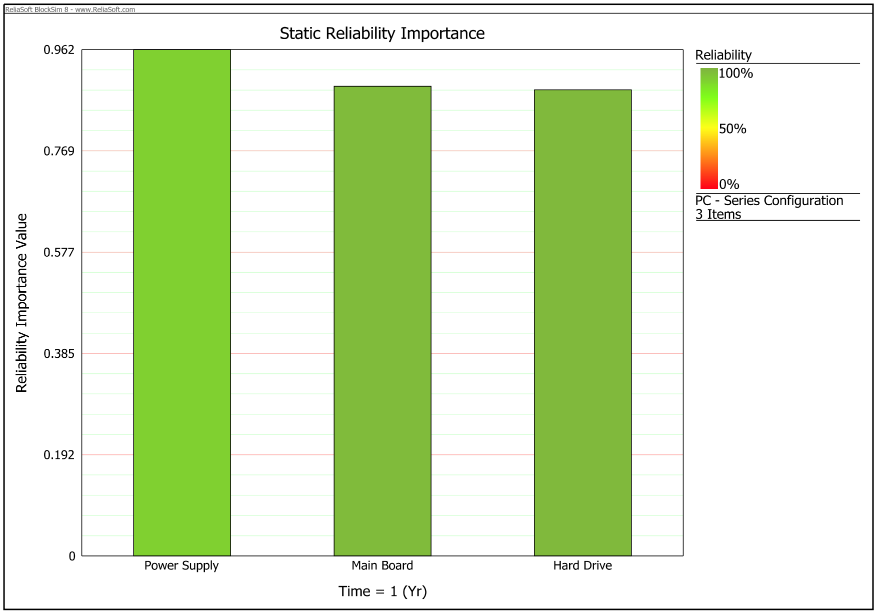


You then follow similar steps to view the *pdf* plot and the failure rate plot, which are shown next.





To determine which of the components is most critical to reliability at the age of 1 year, you choose the **Static RI** plot from the **Plot Type** drop-down list. You then choose **Year** from the **Units** drop-down list, enter **1** in the **At Time** field and click the **Plot** icon again to redraw the plot. The resulting plot shows that the Power Supply block has the highest *reliability importance* after a year of operation.



Reliability Importance: Reliability importance is a measure of how much effect each component in a system has on the overall reliability of the system. In general, the least reliable component in a series system has the greatest effect on the reliability of the system. For more complex configurations, it is less obvious which components have a higher reliability importance.

2.2 Parallel Configuration

The company decides to add redundancy to the system by incorporating an additional hard drive in parallel with the first hard drive.

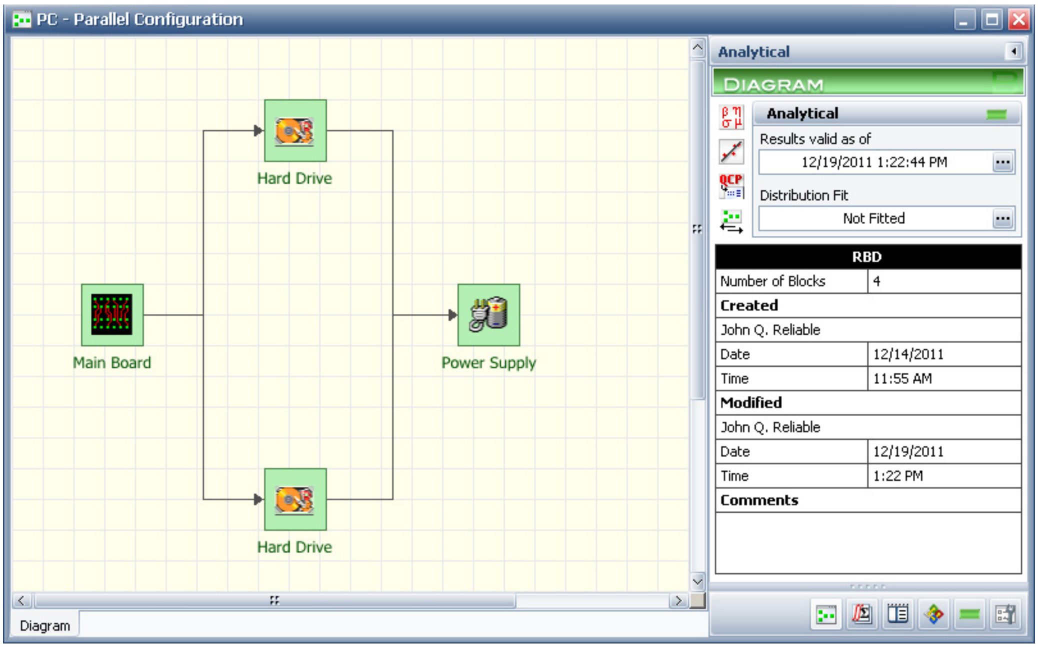
Objectives

- Determine the MTTF or mean life of the system.
- Estimate the B10 life of the computer.
- Compare the reliability, *pdf* and failure rate plots for the series and parallel configurations.

Solution

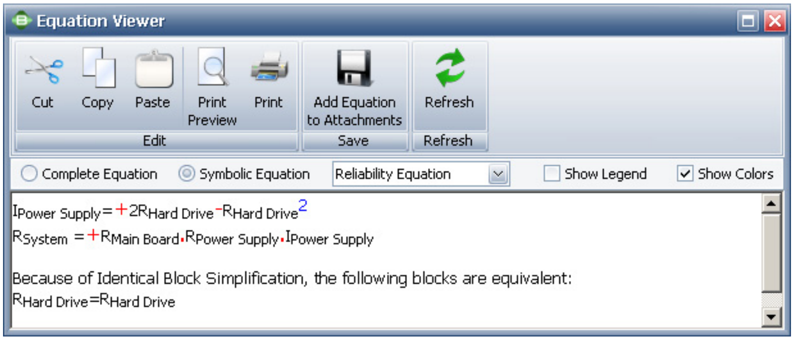
Since the new diagram will be using the same components as the first diagram, you decide to duplicate the original series configuration diagram (*"PC - Series Configuration" diagram in the sample project*) and modify the copy. You select the existing diagram in the current project explorer and choose **Project > Current Item > Duplicate > Duplicate**. You then rename the new diagram (*"PC - Parallel Configuration" in the sample project*) by selecting it in the current project explorer and pressing **F2**.

You will need two hard drives for the updated system configuration, so you select the Hard Drive block and press **CTRL+C**, then press **CTRL+V** to paste a copy of the block. You then select and delete each existing connector and arrange and connect the blocks as shown next.



Tip: By default, the connectors used in the diagram are straight lines. To use a different style in the diagram, choose **Format > Styles > Diagram Style** to open the Diagram Style window. On the Connector page, use the **Bend Style** field to specify the desired bend style for all connectors used in the diagram.

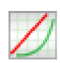
You analyze the diagram and then open the Equation Viewer to see how the system reliability equation differs for the new configuration.



You then open the analytical QCP and perform the same calculations for this diagram as you did for the series configuration. You find:

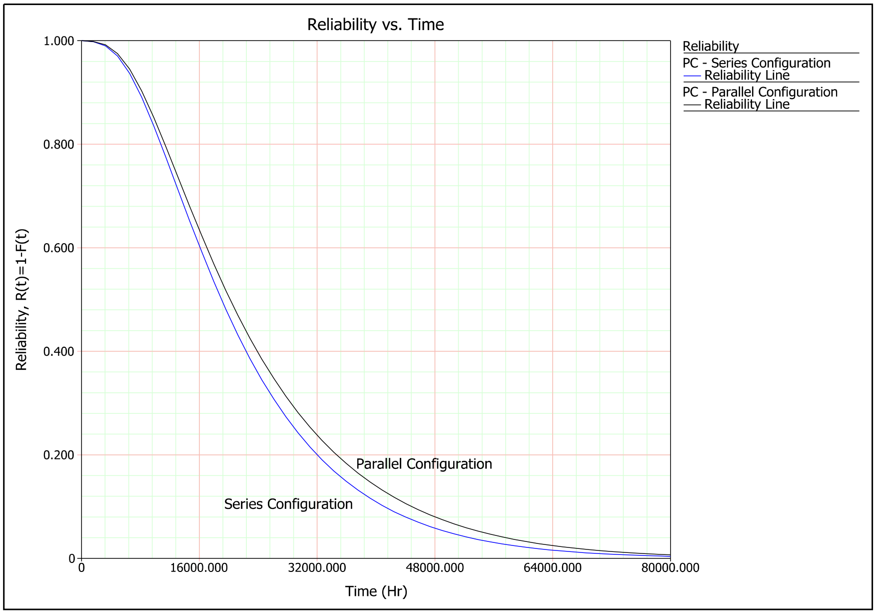
- The mean life for the computer with two hard drives in parallel is approximately 23,949 hours (compared to the 22,296 hours for the series configuration).
- The B10 life for this configuration is approximately 8,260 hours (compared to 7,864 hours for the series configuration).

For a graphical view of this comparison, you create an overlay plot by choosing **Insert > Reports and Plots > Overlay Plot**.



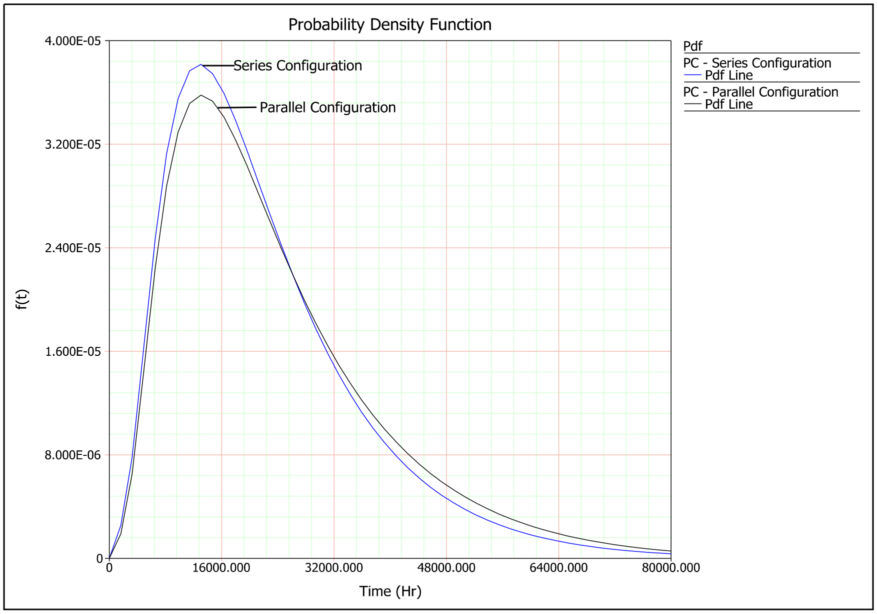
When prompted to select which diagrams to plot, you select the series configuration and the parallel configuration (*"PC - Series Configurations" and "PC - Parallel Configuration" in the sample project*).

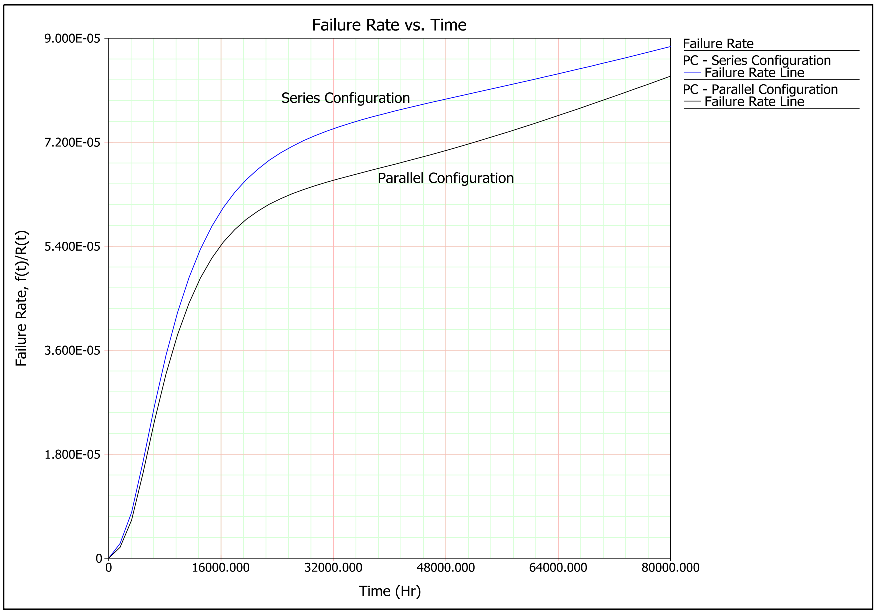
On the control panel of the plot sheet, you switch the plot type to a **Reliability vs. Time** plot. The following overlay plot (*"Overlay - Series and Parallel" in the sample project*) shows that while the reliability of the parallel configuration follows the same curve as that of the series configuration, it is increased.



Tip: You can add labels to the plot by holding down the **CTRL** key and then clicking in the plot. You can then edit the text by double-clicking the labels. To position the text on the plot, you can click the label and then drag it by its handle. For more extensive annotations you can use the RS Draw utility, which is a metafile graphics editor that allows you to insert text, draw objects, paste another picture into the plot and rearrange the objects in the plot. You can open the utility by clicking the **RS Draw** icon on the control panel.

You then compare the *pdf* plots for the two diagrams, and then the failure rate plots, as shown next (with annotations added via RS Draw to make the plots easier to interpret).





2.3 *k-out-of-n* Configuration

The company decides to test a different computer configuration where three hard drives are available, and two of the three must be functioning at any given time.

Objectives

- Determine the MTTF or mean life of the system.
- Estimate the B10 life of the computer.
- Compare the reliability, *pdf* and failure rate plots for the series, parallel and *k-out-of-n* configurations.

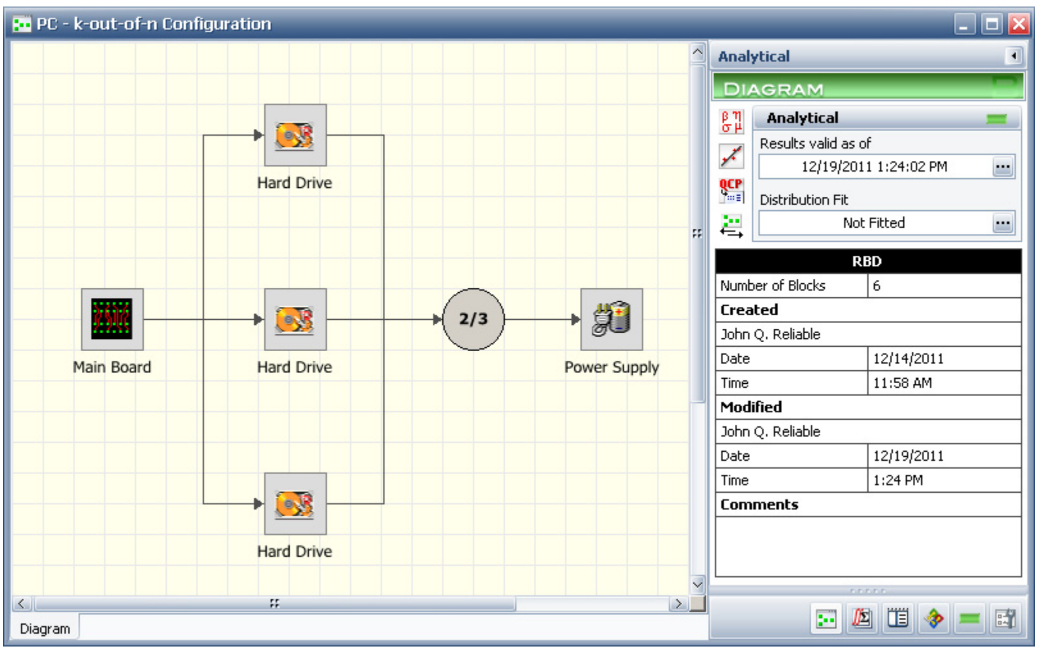
Solution

You duplicate the parallel configuration diagram (*"PC - Parallel Configuration" diagram in the sample project*) by right-clicking it in the current project explorer and choosing **Duplicate** on the shortcut menu and rename the copy (*"PC - k-out-of-n Configuration" in the sample project*).

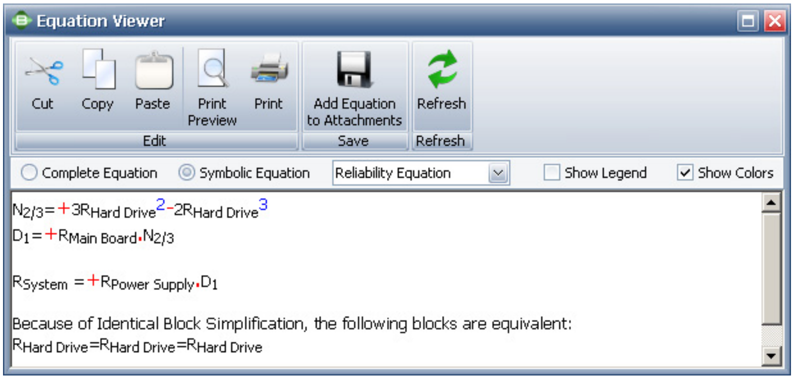
You will need three hard drives for the updated system configuration, so you select one of the Hard Drive blocks and press **CTRL+C**, then press **CTRL+V** to paste another copy of the block. You also add a node block by choosing **Diagram > Blocks > Add Node**. In the Block Properties window for the node block, you enter **2** in the **Number of paths required** field.

Node blocks and k-out-of-n configurations: Node blocks allow you to specify a form of redundancy known as *k-out-of-n* redundancy. A *k-out-of-n* node can have *n* paths leading into it and requires that *k* out of *n* paths must function for the system to function.

You then select and delete each existing connector and arrange and connect the blocks as shown next.



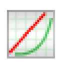
You analyze the diagram and then open the Equation Viewer to see how the system reliability equation differs for the new configuration.



You open the analytical QCP and perform the same calculations for this diagram as you did for the series configuration. You find:

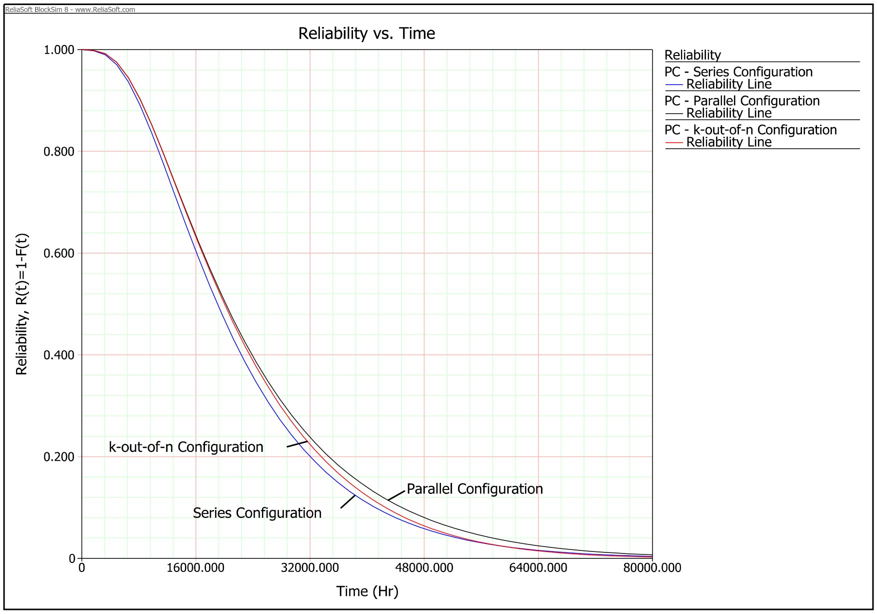
- The mean life for this configuration is approximately 23,151 hours (compared to 23,949 hours for the parallel configuration and 22,296 hours for the series configuration).
- The B10 life for this configuration is approximately 8,249 hours (compared to 8,260 hours for the parallel configuration and 7,864 hours for the series configuration).

For a graphical view of this comparison, you create an overlay plot by choosing **Insert > Reports and Plots > Overlay Plot**.

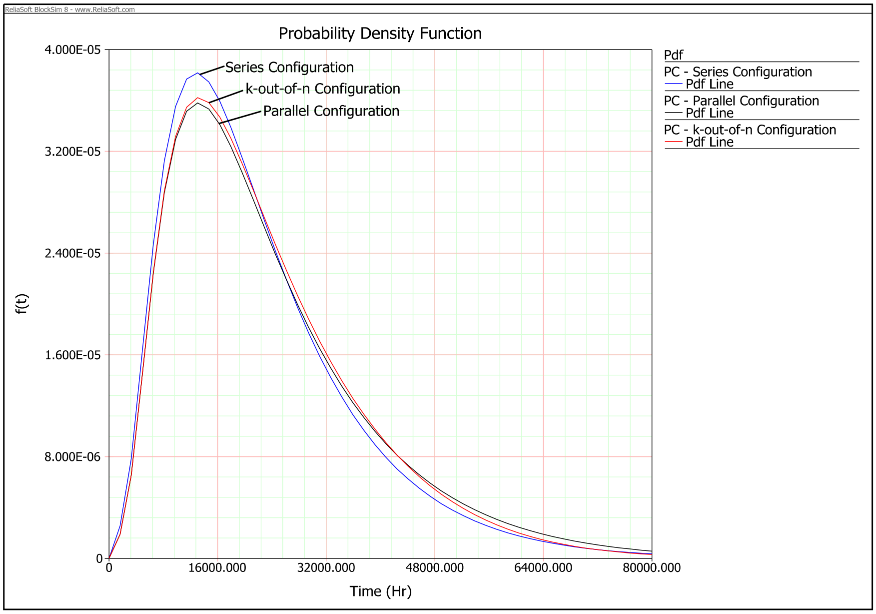


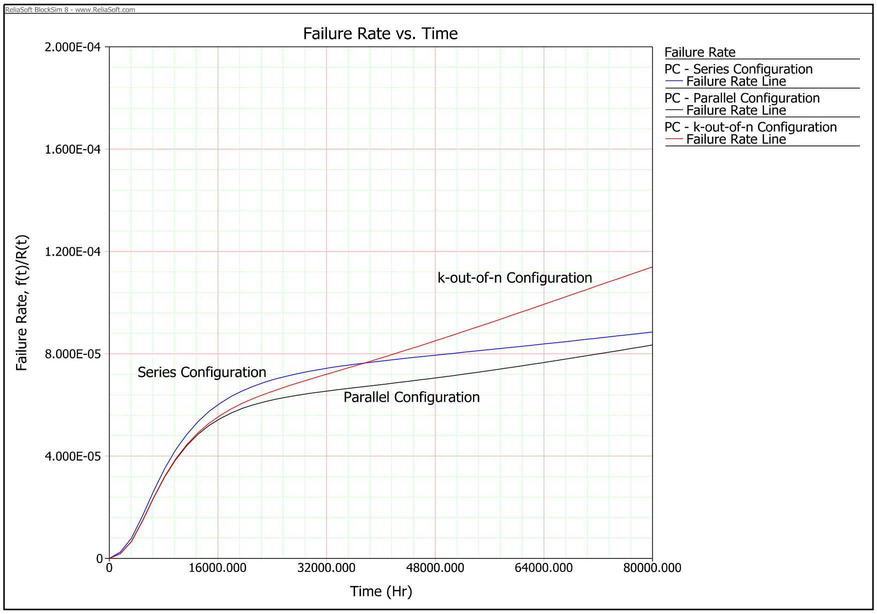
When prompted to select which diagrams to plot, you select all three diagrams that you have been working with.

On the control panel of the plot sheet, you switch the plot type to a **Reliability vs. Time** plot. The resulting overlay plot (*"Overlay - Series, Parallel, k-out-of-n" in the sample project*) is shown next (with annotations added via RS Draw to make the plot easier to interpret).



You then compare the *pdf* plots for the three diagrams, and then the failure rate plots, as shown next (with annotations added via RS Draw to make the plots easier to interpret).




ReliaSoft BlockSim 8 - www.ReliaSoft.com
Failure Rate vs. Time
Failure Rate, f(t)/R(t)
2.000E-04
1.600E-04
1.200E-04
8.000E-05
4.000E-05
0
0
16000.000
32000.000
48000.000
64000.000
80000.000
Time (Hr)
k-out-of-n Configuration
Series Configuration
Parallel Configuration
Failure Rate
PC - Series Configuration
Failure Rate Line
PC - Parallel Configuration
Failure Rate Line
PC - k-out-of-n Configuration
Failure Rate Line

Using Subdiagrams in Analytical Calculations

3

Subdiagram blocks represent other diagrams within the project. Using subdiagram blocks allows you to maintain separate diagrams for portions of a system and to incorporate those diagrams as components of another diagram. This also allows you to generate and analyze extremely complex diagrams representing many subsystems, sub-subsystems, etc.

In this chapter, you will examine the same computer system from the previous chapter in greater detail using subdiagrams.

In this chapter

- ✓ Reliability block diagrams (RBDs)
- ✓ Fault trees
- ✓ Subdiagram blocks
- ✓ Mirrored blocks
- ✓ QCP and plots

ReliaSoft.TV

Watch a video of this example at http://www.reliasoft.tv/blocksim/startguide/blocksim_qsg_3.html.

3.1 Using a Subdiagram

The model used for the reliability of the main board during the initial analysis of the desktop computer system was an initial estimate. Subsequently, you are able to obtain more information on the subsystems that make up the main board, including their actual reliability models. For the sake of simplicity in this example, assume that the main board is made up of the following components, in series:

- Memory, with reliability following an exponential distribution with MTTF = 1,000,000 hours.
- A PCI board, with reliability following a Weibull distribution with beta = 2.1 and eta = 100,000 hours.
- A CPU, with reliability following a Weibull distribution with beta = 1.3 and eta = 500,000 hours.

Objectives

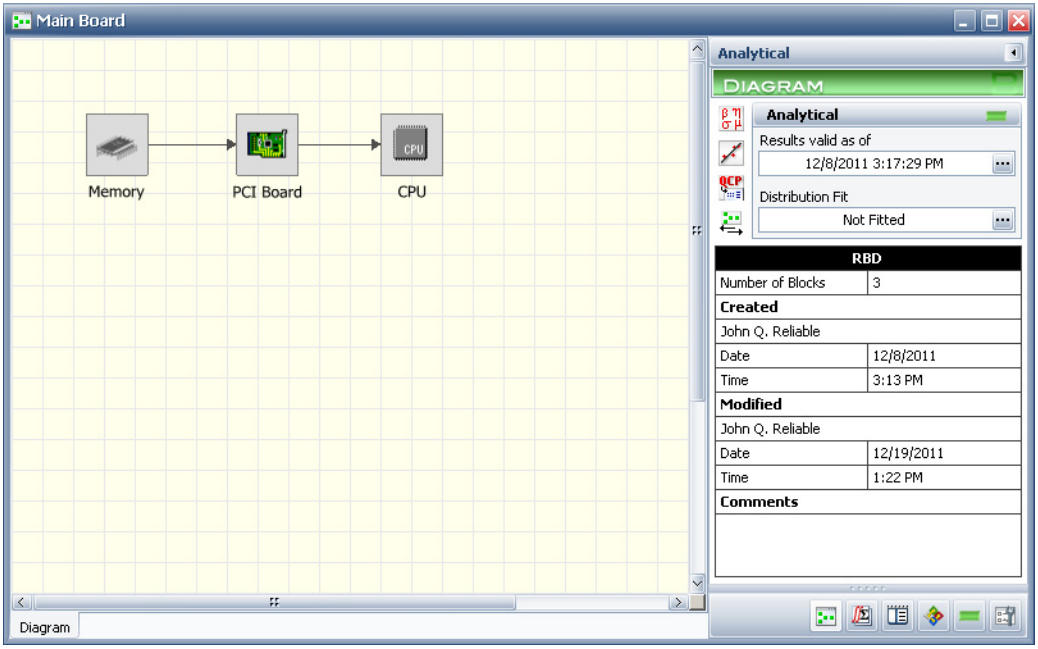
- Determine the MTTF or mean life of the main board.
- Estimate the B10 life of the main board.
- Determine which of the components is most critical to reliability at the age of 5 years.
- Create a new version of the series configuration diagram for the computer, using your diagram of the main board as a subdiagram.

Solution

First, you create a new analytical RBD (*"Main Board" in the sample project*) for the main board and add three blocks to the diagram. You configure the blocks as:

- The Memory block, using the Memory URD and the Memory Model (1-parameter exponential, mean time = 1,000,000 hours).
- The PCI Board block, using the PCI Board URD and the PCI Board Model (2-parameter Weibull, beta = 2.1, eta = 100,000 hours).
- The CPU block, using the CPU URD and the CPU Model (2-parameter Weibull, beta = 1.3, eta = 500,000 hours).

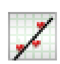
You then connect them as shown next.



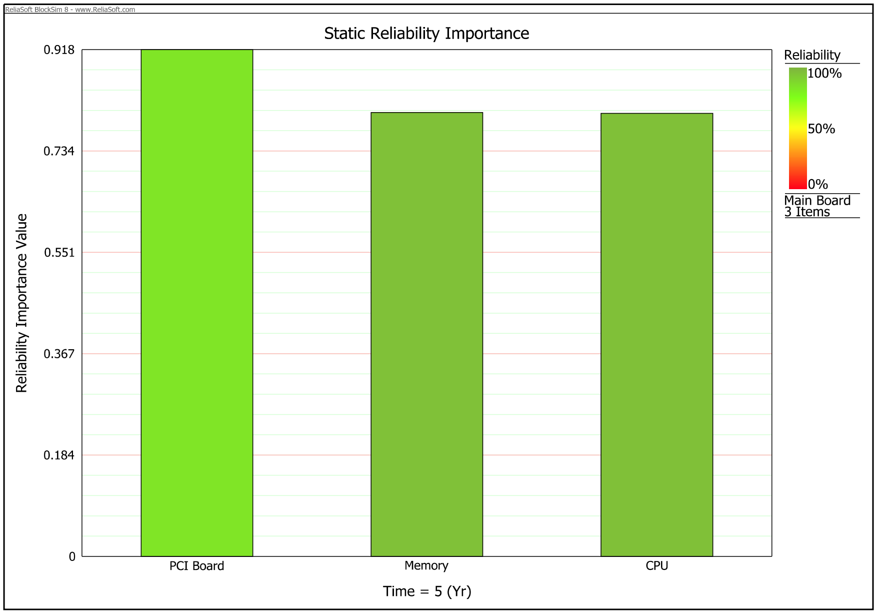
You analyze the diagram and then open the analytical QCP and perform the mean life and BX% life calculations. You find:

- The mean life for the main board is approximately 79,026 hours.
- The B10 life for the main board is approximately 25,833 hours.

To determine which of the components is most critical to reliability at the age of 5 years, you click the **Plot** icon on the control panel.

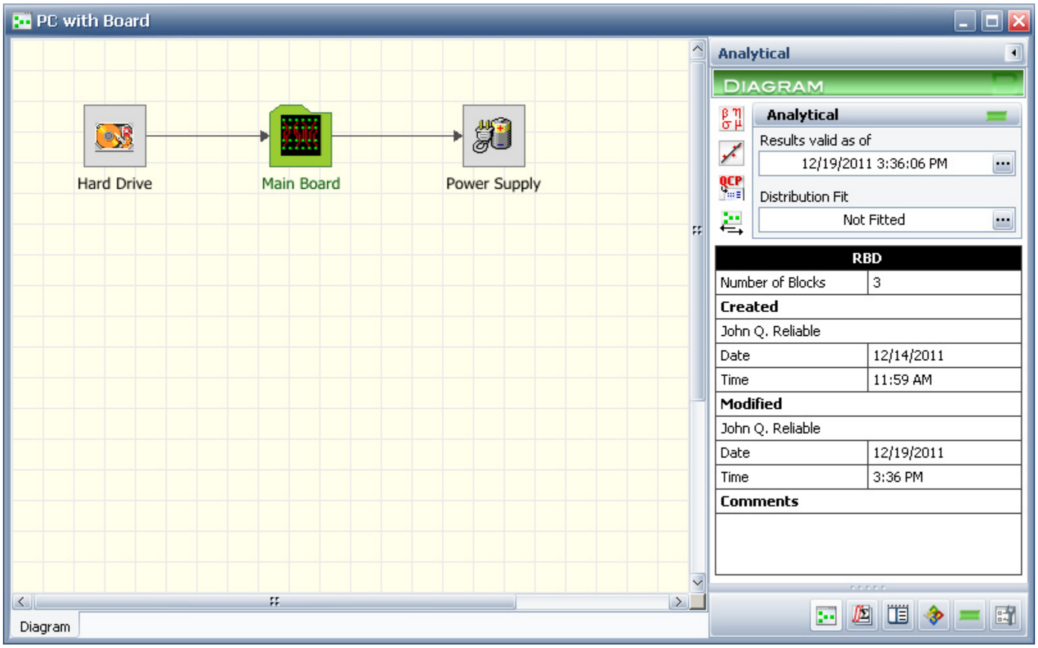


On the control panel of the plot sheet, you choose the **Static RI** plot from the **Plot Type** drop-down list. You then choose **Year** from the **Units** drop-down list and enter **5** in the **At Time** field. The resulting plot shows that the PCI Board block has the highest reliability importance after 5 years of operation.



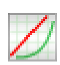
To incorporate this improved model of the main board's behavior into the computer's system configuration, you duplicate the original series configuration diagram (*"PC - Series Configuration" diagram in the sample project*) by right-clicking it in the current project explorer and choosing **Duplicate > Duplicate** on the shortcut menu. You then rename the new diagram (*"PC with Board" in the sample project*) by selecting it in the current project explorer and pressing **F2**.

In the new diagram, you right-click the Main Board block and choose **Convert Block To > Subdiagram** on the shortcut menu. In the Select Diagram for the Block window that appears, you select the diagram representing the main board (*"Main Board" in the sample project*). The resulting diagram is shown next.



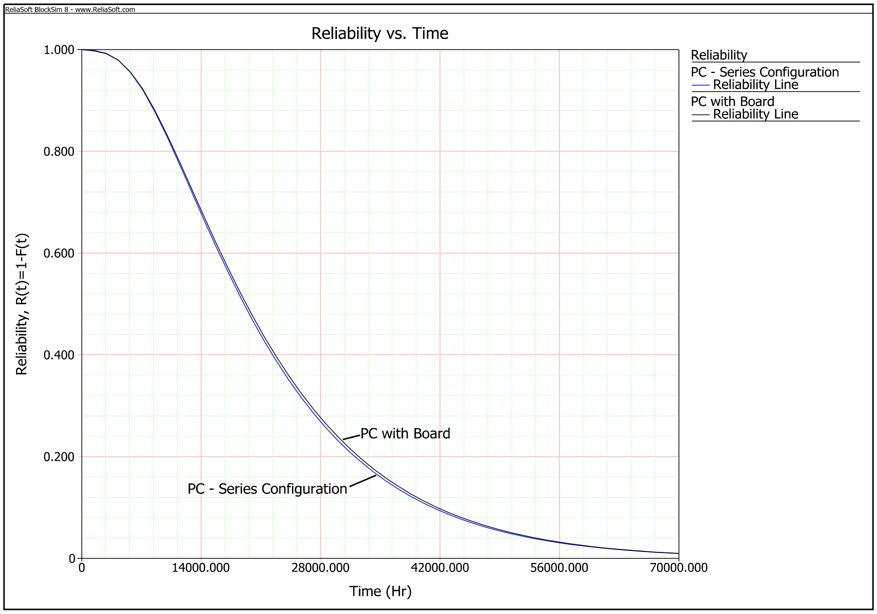
Note: A subdiagram is represented in the diagram by a subdiagram block in the shape of a folder. Double-clicking a subdiagram block opens the associated subdiagram. If you need to access the Block Properties window for a subdiagram block, select the block and either choose **Diagram > Properties > Block Properties** or press **CTRL+E**.

You analyze the diagram and then, to see how the improved model affects results, you create an overlay plot by choosing **Insert > Reports and Plots > Overlay Plot**.



When prompted to select which diagrams to plot, you select the original series configuration diagram (*"PC - Series Configuration" in the sample project*) and the new diagram that uses the subdiagram (*"PC with Board" in the sample project*).

On the control panel of the plot sheet, you switch the plot type to a **Reliability vs. Time** plot. The plot (*"Overlay - Series, PC with Board" in the sample project*) appears as shown next (with annotations added via RS Draw to make the plots easier to interpret).



This plot shows that the results from the two diagrams are similar, but not identical.

3.2 Fault Trees

Continuing with this example, let's assume that the reliability of the memory chips is due to known failure modes. There are six known failure modes: H1, H2, S1, S2, E1 and E2.

Assume that each failure mode is described by an exponential model with:

- MTTF of 2,500,000 hrs for H1 and H2.
- MTTF of 1,000,000 hrs for S1 and S2.
- MTTF of 1,500,000 hrs for E1 and E2.

H modes are hard modes and if either one occurs, then the chip will fail. In addition, the chip will fail if either S1 or S2 occurs along with either E1 or E2. In other words, the system will fail if [S1 AND (E1 or E2)] happen or [S2 AND (E1 or E2)].

In a logic statement, this could be phrased as follows:

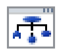
H1 OR H2 OR {[S1 AND (E1 OR E2)] OR [S2 AND (E1 or E2)]}

Objectives

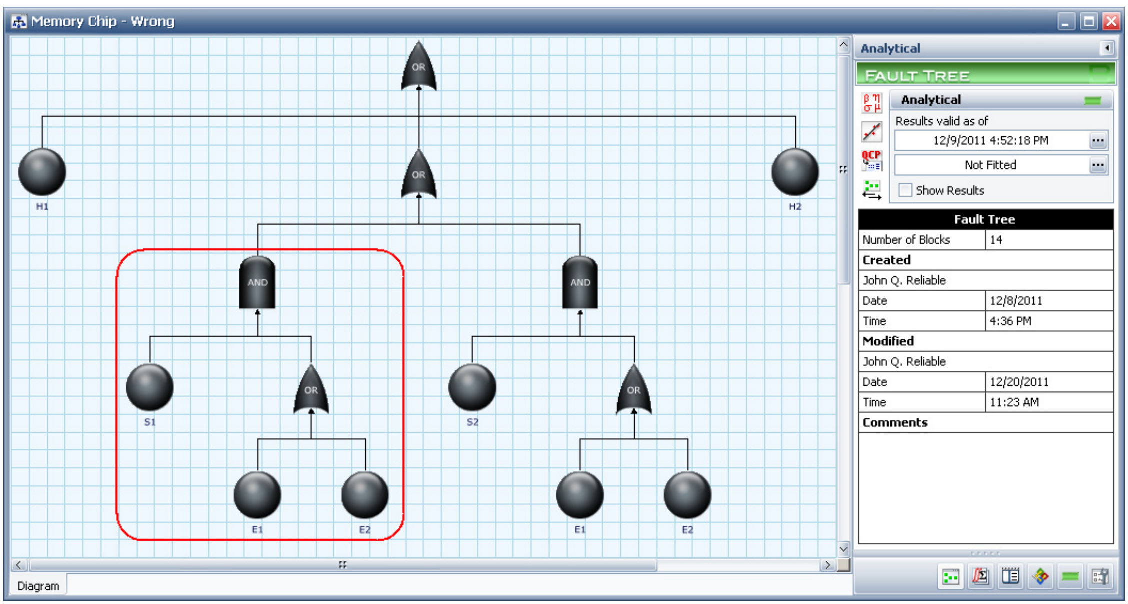
- Create a fault tree diagram representing the failure modes.
- Create a corresponding RBD.
- Compare the diagrams.

Solution

First, you create a new analytical fault tree (*"Memory Chip - Wrong" in the sample project*) by choosing **Insert > Diagrams > Analytical Fault Tree**.



The diagram that you will build is shown next.

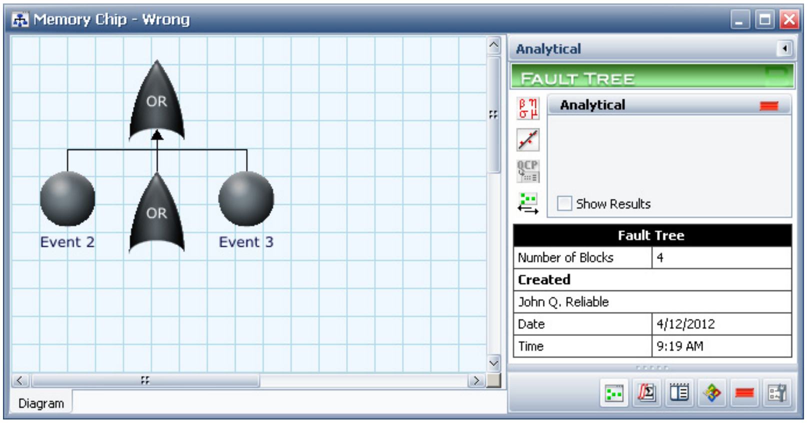


In this diagram, failure will happen if H1 happens, or if H2 happens, or if one of the subsets of other events below the lower OR gate happens. The circled portion of the diagram indicates that this particular subset of events happens if S1 happens and either E1 or E2 also happens.

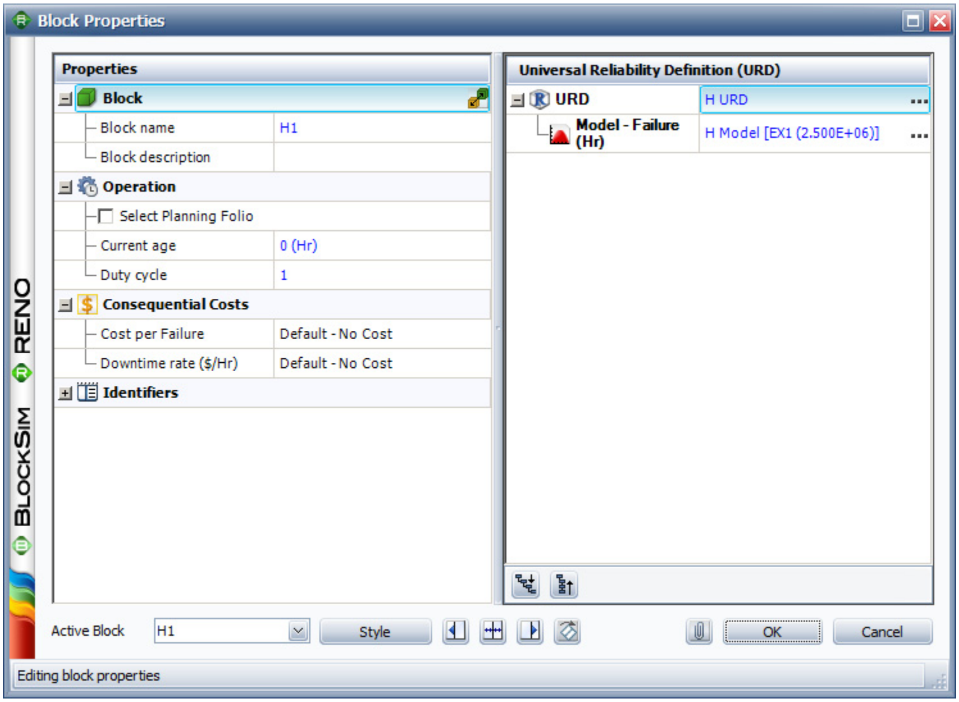
To build the diagram, you start by adding an OR gate as the top block, since the failure can be caused by several modes or combinations of modes. To do this, you choose **Fault Tree > Gates and Events > Add Gate > Add OR Gate**.

With the OR gate selected, you add two events below it by choosing **Fault Tree > Gates and Events > Add Event** twice.

With the OR gate still selected, you add another OR gate below it. The fault tree looks like the one shown next.



You then select the first event and modify its block properties so that it is named **H1**. You create a new URD called **H URD**, using a new model called **H Model** (1-parameter exponential, mean time = 2,500,000 hours), and apply it to the event.



You name the other event **H2** and apply the H URD to that event as well.

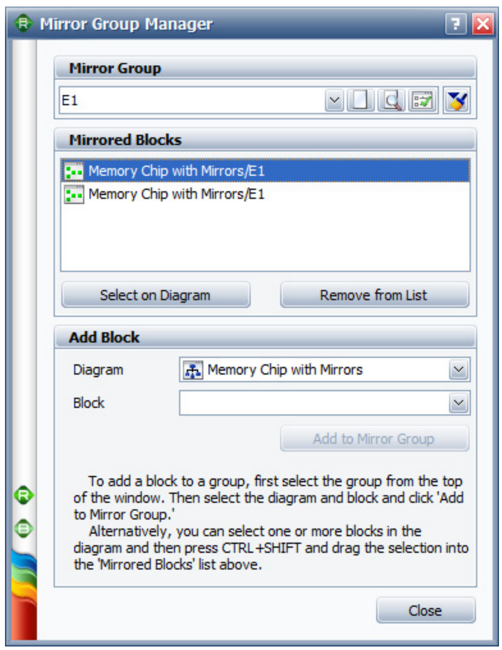
You select the lower OR gate and add two AND gates below it. Each of these gates represents the combination of an S mode and one of the E modes, as given in the logic statement. You add the remaining gates and events, and define the block properties for the events so that S1 and S2 use the S URD and the S Model (1-parameter exponential, mean time = 1,000,000 hours) and E1 and E2 use the E URD and the E Model (1-parameter exponential, mean time = 1,500,000 hours).

You look at the fault tree that you have created and realize that there is a problem. Events E1 and E2 appear in both branches of the fault tree below the second OR gate. However, the E1 in the left branch is the same as the E1 in the right branch, and its probability of happening should be counted only once. If it is represented twice within the diagram, it will be counted twice. The same is true for E2. You see that you need to fix this using mirrored blocks, which allow you to represent a single block multiple times.

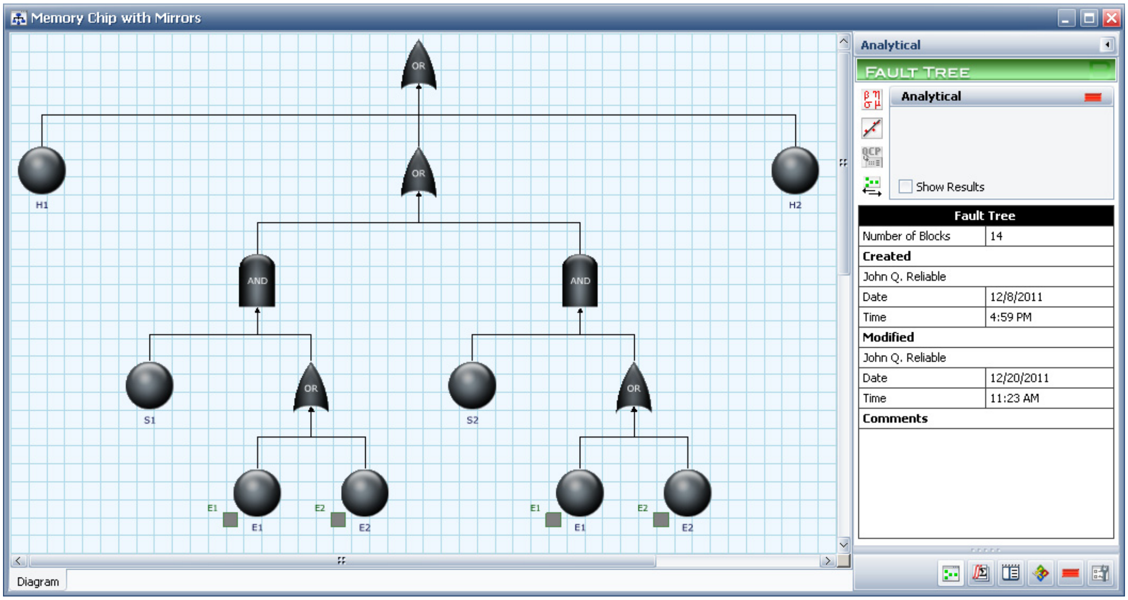
You duplicate the fault tree diagram, giving yourself a fresh copy to work on (*"Memory Chip with Mirrors" in the sample project*). You open the Mirror Group Manager by choosing **Project > Data Management > Mirror Group Manager**.

In the **Mirror Group** area, you click the **Create New** icon.

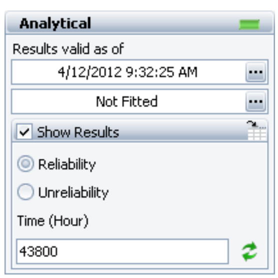
In the Mirror Group window that appears, you name your new mirror group **E1**. In the **Add Block** area of the Mirror Group Manager, you select your new fault tree diagram in the **Diagram** field and one of the **E1** events in the **Block** field, then click **Add to Mirror Group**. You then add the other E1 event to the mirror group.



You create the E2 group and follow the same steps to add the E2 events to that group. The resulting diagram is shown next. The mirrored blocks are indicated by a gray square at the lower left corner of the block and the name of the mirror group is displayed along with the indicator



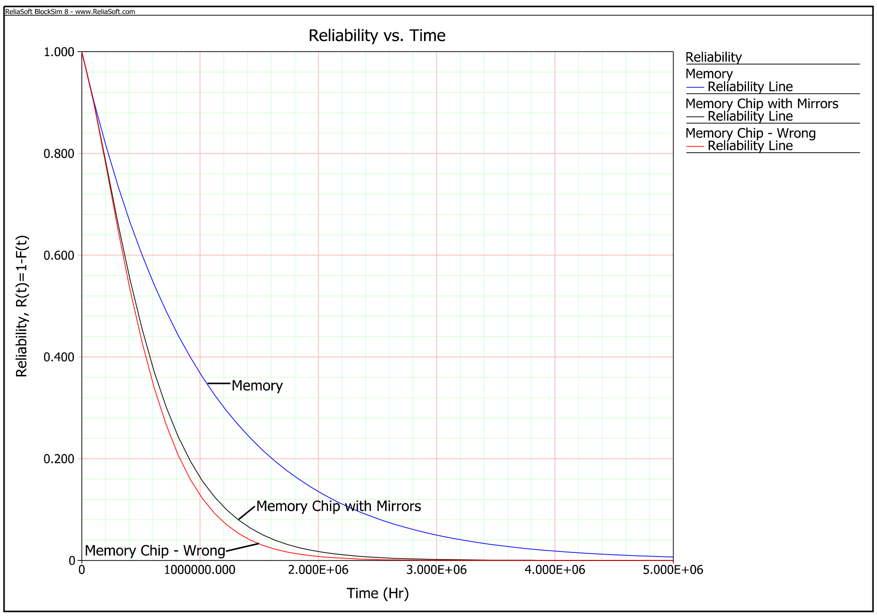
This diagram correctly represents the failure mode behavior. To understand how mirroring affects the results of the diagrams, you analyze both the original fault tree and the new one using mirrored blocks (*"Memory Chip - Wrong" and "Memory Chip with Mirrors" in the sample project*). You then select the **Show Results** check box on the control panel for each diagram, select to show the results as **Reliability** and specify to view the results at **43,800 hours** (which is equal to 5 years), as shown next.

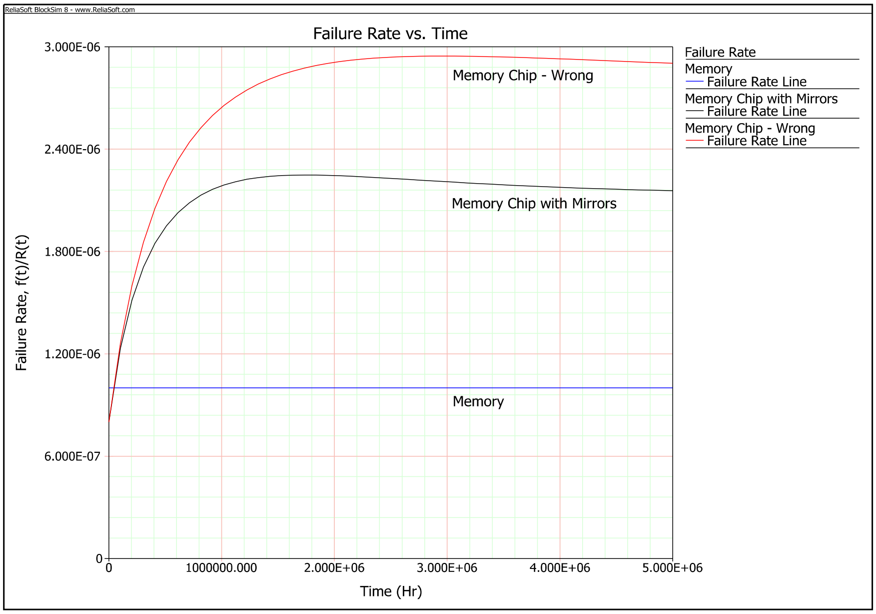


This will display the reliability of each block at the specified time beside the block in the diagram. You click the **Refresh** icon to update the results shown in the diagram.

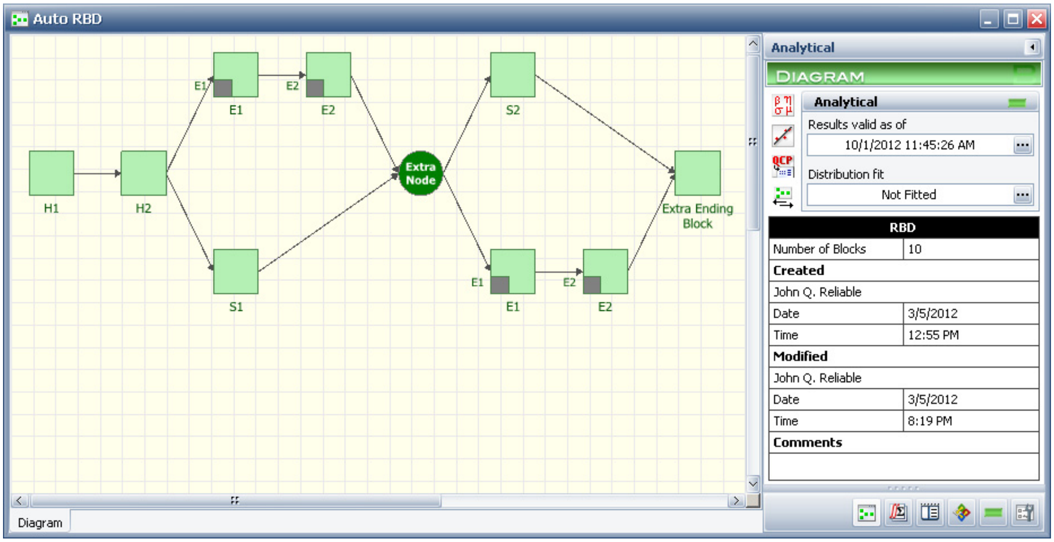
Looking at the top level OR gate, you can see that the overall reliability for the system when modeled with mirrored blocks is 96.097274%, as opposed to the figure of 96.087785% obtained without mirrored blocks.

You would like to see how the reliability of these diagrams compares over time. In addition, you want to know how their results compare to the original model used for the memory chip in the main board. In order to do this, you first create a new analytical RBD (*"Memory" in the sample project*) with a single block, Memory, using the Memory URD. You analyze the diagram. You then create an overlay plot that includes the "Memory" diagram and both fault trees (*"Overlay - Memory" in the sample project*) and view the Reliability vs. Time plot and the Failure Rate vs. Time plot as shown next (with annotations added via RS Draw to make the plots easier to interpret).

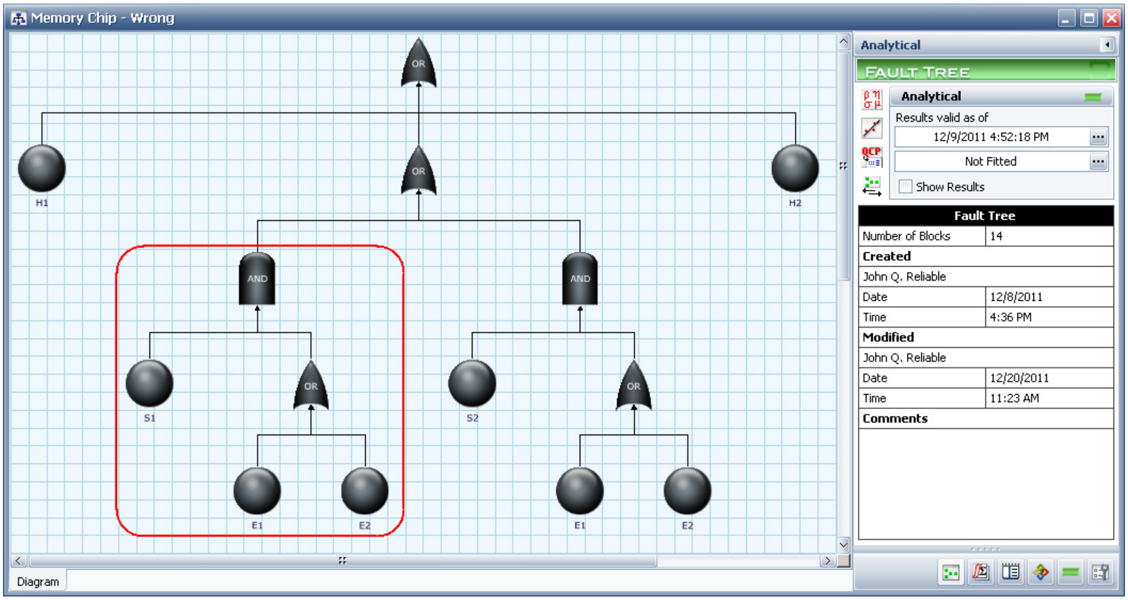




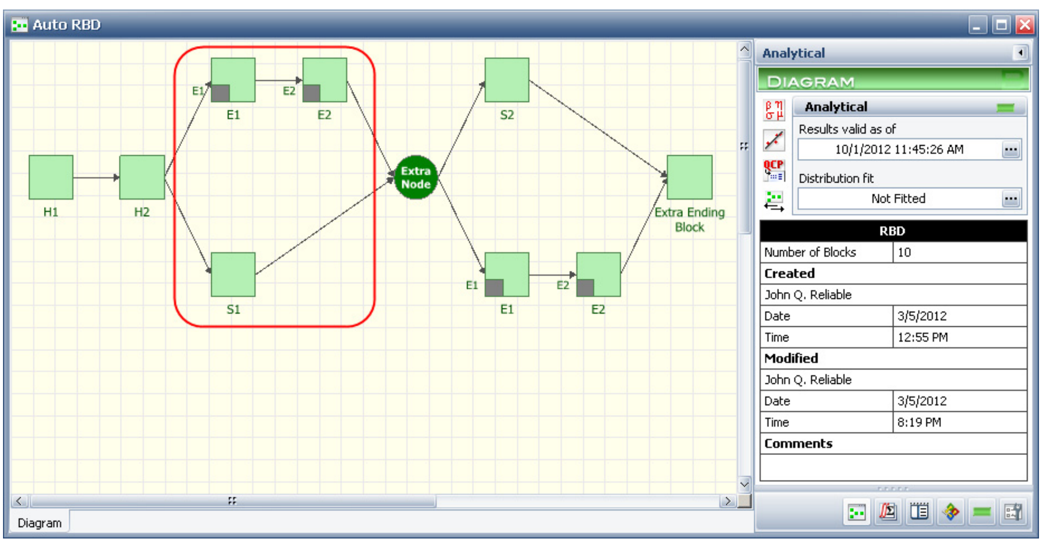
To see how this configuration can be represented in an RBD, you return to the fault tree that uses mirror blocks and choose **Fault Tree > Conversion > Create RBD**. An RBD that is equivalent to the fault tree (*"Auto RBD" in the sample project*) is added to the project. Note that the blocks in the diagram shown next have been rearranged for structural clarity, but no changes have been made to any blocks or connections.



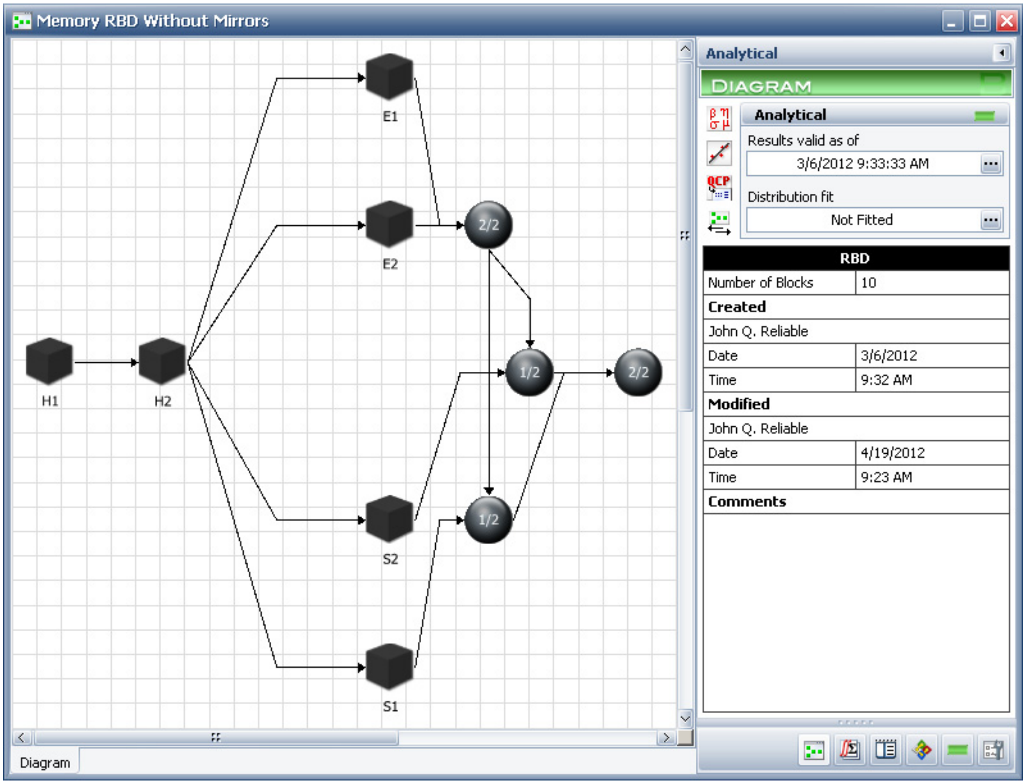
While fault trees consider failure combinations, RBDs consider success combinations. For example, as noted previously, the circled portion of the fault tree shown next indicates that this particular subset of events happens (i.e., failure occurs) if failure mode S1 happens and either failure mode E1 or failure mode E2 also happens.



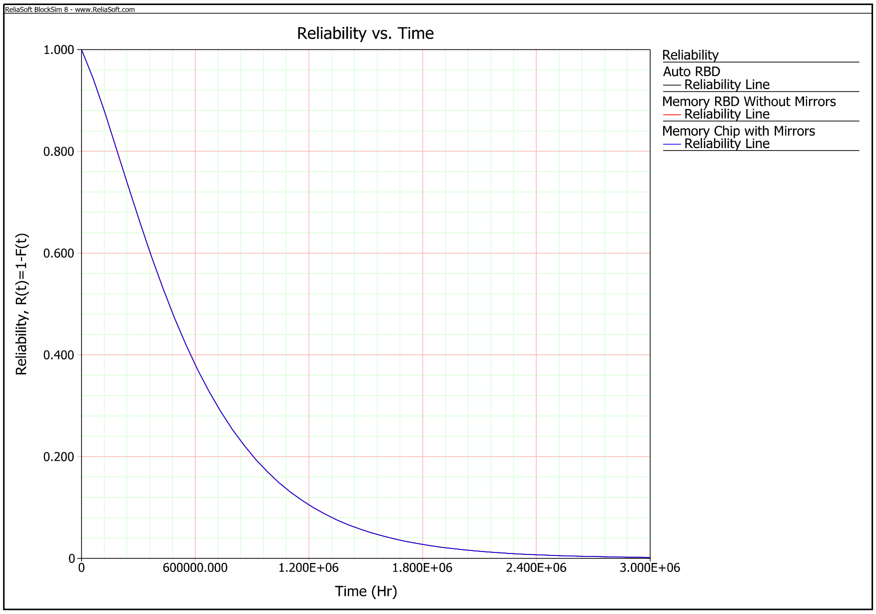
This corresponds to the circled portion of the RBD shown next, which indicates that the subsequent event happens (i.e., the diagram succeeds) if either S1 succeeds (i.e., the S1 failure mode does not happen) or both E1 and E2 succeed (i.e., neither the E1 nor the E2 failure mode happens).



To create an RBD representation of this configuration without mirror blocks, you add another analytical RBD (*"Memory RBD Without Mirrors" in the sample project*) to the project. Using a series of node blocks, you model the possible success paths using only a single copy each of E1 and E2, as shown next.



To verify that this RBD, the automatically created RBD and the fault tree with mirrors are equivalent, you create a Reliability vs. Time overlay plot (*"Overlay - RBDs and FT" in the sample project*) that includes all three diagrams (*"Memory RBD Without Mirrors," "Auto RBD" and "Memory Chip with Mirrors" in* *the sample project*). You see that all of the lines on the plot are in the exact same location, indicating that the diagrams are, indeed, equivalent.

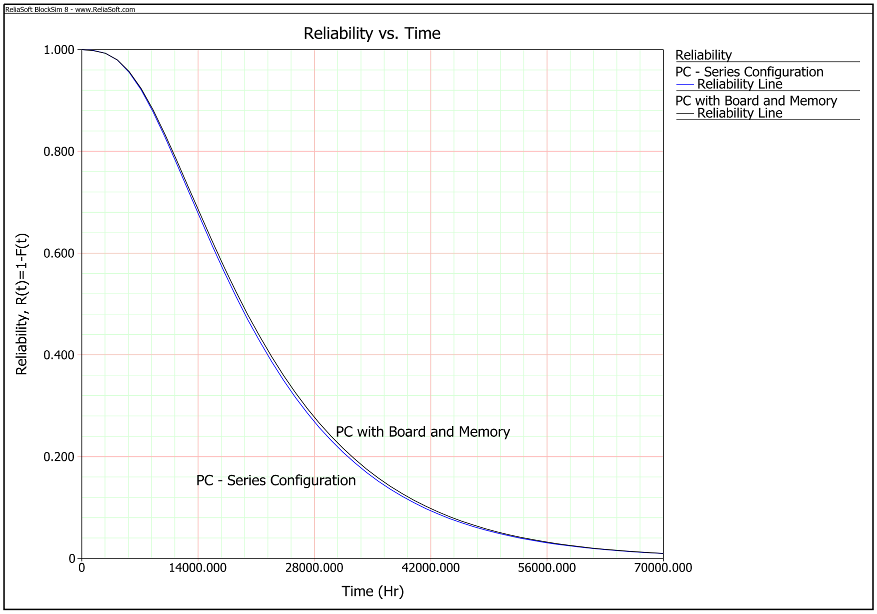


3.3 Fault Trees as Subdiagrams

Now that you have created several accurate models of the failure modes that contribute to the memory chip's failure, you can incorporate any one of these diagrams into the analysis of the main board as a subdiagram.

You duplicate the existing "Main Board" diagram. In the new diagram (*"Main Board with Memory" in the sample project*), you right-click the Memory block and choose **Convert Block To > Subdiagram** on the shortcut menu. In the Select Diagram for the Block window that appears, you select the fault tree representing the memory's failure modes (*"Memory Chip with Mirrors" in the sample project*). You then repeat this process to create a duplicate of the "PC - Series Configuration" diagram (*"PC with Board and Memory" in the sample project*) using the "Main Board with Memory" diagram as a subdiagram in place of the original Main Board block. You then analyze this new diagram.

To see how the more detailed modeling achieved via subdiagrams affects the overall system results, you create a Reliability vs. Time overlay plot (*"Overlay - Subdiagrams" in the sample project*) that includes the newest RBD (*"PC with Board and Memory" in the sample project*) and the original series configuration RBD (*"PC - Series Configuration" in the sample project*). You see that while the curves are very similar, including the subdiagrams with detailed information about the memory chip reveals a slightly higher reliability, as shown next (with annotations added via RS Draw to make the plots easier to interpret).



Improving Reliability 4

Often, you will have a reliability requirement for your system at a certain time of operation. BlockSim offers several tools to help you determine whether your system meets that requirement and, if it does not, which components should be improved.

In this chapter, you will work with a storage area network that includes the computer system from previous examples in a complex configuration.

In this chapter

- ✓ Reliability block diagrams (RBDs)
- ✓ Subdiagrams
- ✓ QCP
- ✓ Reliability Importance Plots
- ✓ FRED Reports
- ✓ Allocation Analysis

Watch a video of this example at http://www.reliasoft.tv/blocksim/startguide/blocksim_qsg_4.html.

4.1 Improving a Complex System

Assume a storage area network with two servers, four hubs, two switches and two network-attached storage devices.

- The hubs' reliability follows an exponential distribution with MTTF = 200,000 hours.
- The switches' reliability follows an exponential distribution with MTTF = 100,000 hours.
- The storage devices' reliability follows an exponential distribution with MTTF = 50,000 hours.
- The servers are as defined in the previous example (*"PC with Board and Memory" in the sample project*).

The storage area network's reliability-wise configuration involves a complex redundancy, where the failure combinations in the table shown next would cause a system failure.

System Failure Combinations																				
Server 1	x									x	x	x	x	x						
Server 2	x					x			x							x		x		x
Switch 1		x	x							x					x	x				
Switch 2		x		x		x	x				x									
Hub 1						x	x	x	x										x	x
Hub 2									x						x	x	x	x	x	
Hub 3							x	x			x		x	x					x	
Hub 4										x		x		x	x		x		x	
Storage 1				x	x							x					x	x		
Storage 2			x		x			x					x							x

In this table, each column represents a combination of failed components that will cause system failure. For example, in the first column, we see that the combination of the failure of Server 1 and the failure of Server 2 causes system failure.

Objectives

- Determine the system reliability equation, *pdf* equation and failure rate equation.
- Determine the system reliability at one year of operation.
- Assuming that the reliability requirement is 95% for one year, determine what components should be improved, and by how much, to meet the requirement.

Solution

Because you will want to view some block level results in this example and mirrored blocks are not shown in plots that show block level results, you decide that you would prefer to use a diagram without mirrors to represent the computer system that you have been working with.

You create a copy of the "Main Board with Memory" diagram for use in this example (*"Main Board - No Mirrors" in the sample project*). You select the subdiagram block and press **CTRL+E** to open the Block Properties window, where you change the selection in the **Based on diagram** field to base the block on the memory RBD that does not use mirrors (*"Memory RBD Without Mirrors" in the sample project*).

Next, you create a copy of the "PC with Board and Memory" diagram (*"PC - No Mirrors" in the sample project*). You open the Block Properties window for the subdiagram block and select to base it on the main board RBD without mirrors that you just created (*"Main Board - No Mirrors" in the sample project*).

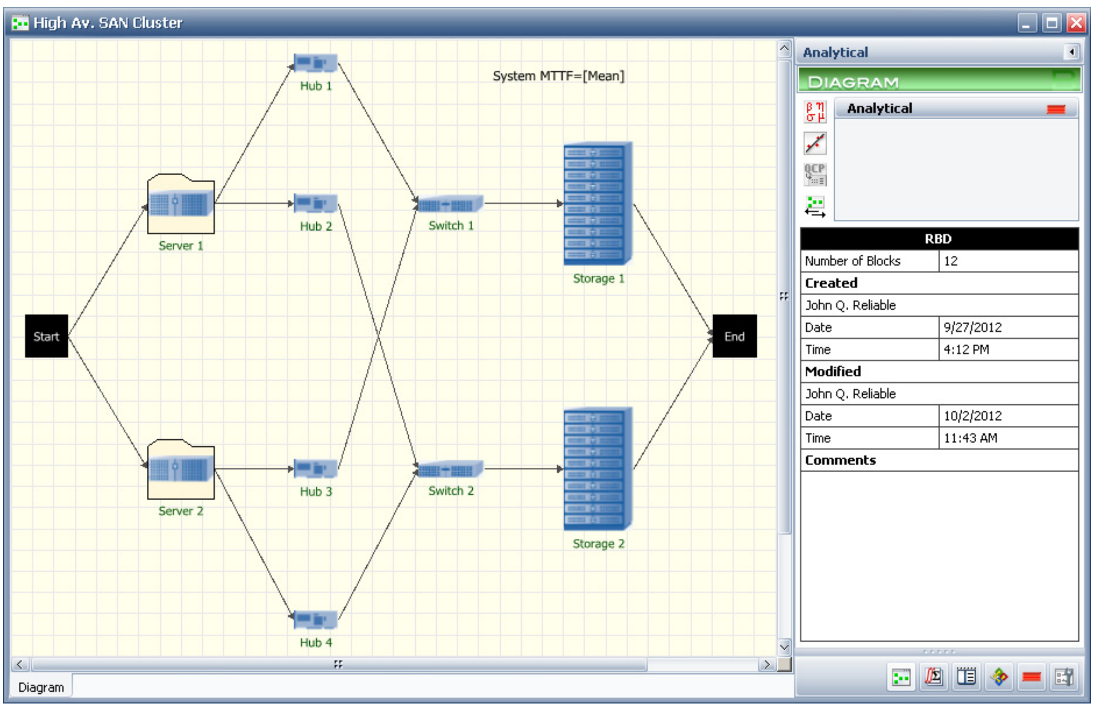
Finally, you create a new analytical RBD (*"High Av. SAN Cluster" in the sample project*) for the storage area network. You add eight standard blocks to the diagram and configure them as:

- Four hubs (Hub 1, Hub 2, Hub 3 and Hub 4), using the Hub URD and the Hub Model (1-parameter exponential, mean time = 200,000 hours).
- Two switches (Switch 1 and Switch 2), using the Switch URD and the Switch Model (1-parameter exponential, mean time = 100,000 hours).
- Two storage devices (Storage 1 and Storage 2), using the Storage URD and the Storage Model (1-parameter exponential, mean time = 50,000 hours).

You then add two subdiagram blocks to the diagram, each representing the server subdiagram (*"PC - No Mirrors" in the sample project*).

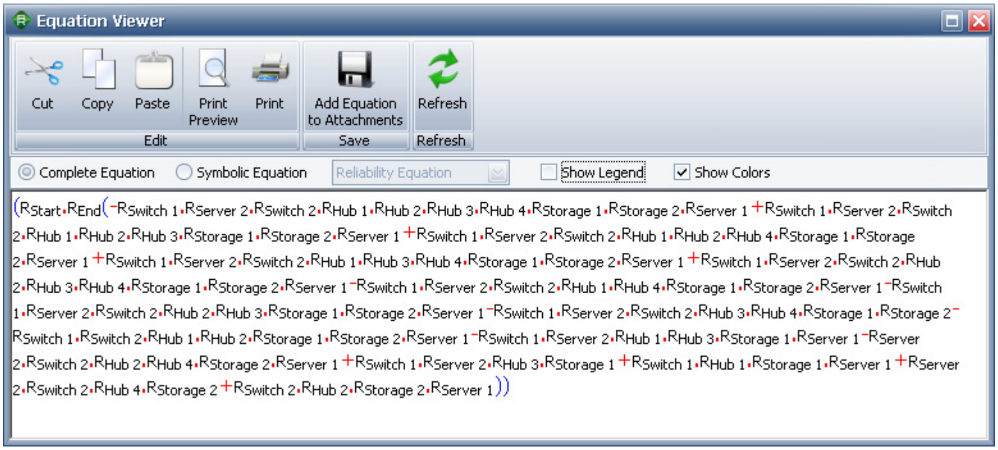
Finally, you add a start block and an end block with no reliability properties (i.e., that cannot fail).

You then connect the blocks as shown next.

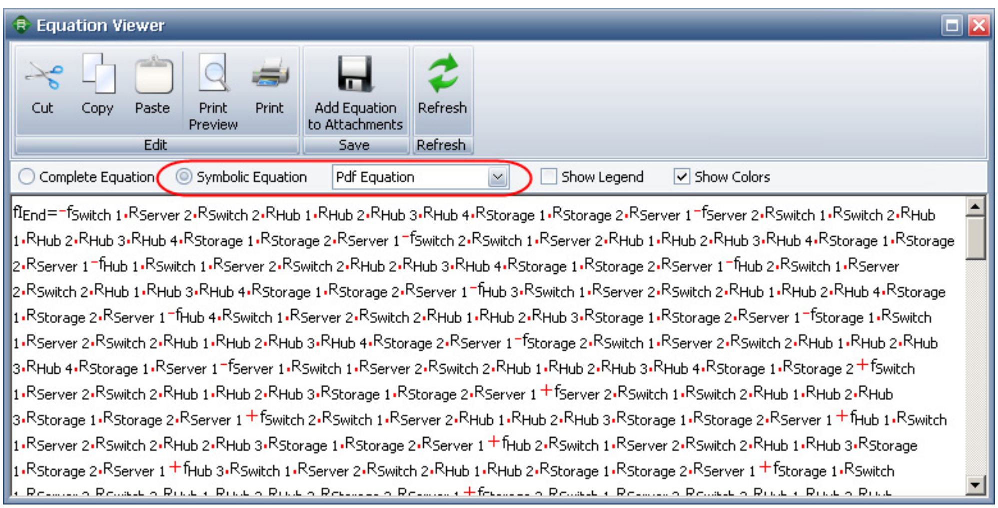


You analyze the diagram and then choose **Analysis > Tools > Show Algebraic Solution** to open the Equation Viewer.

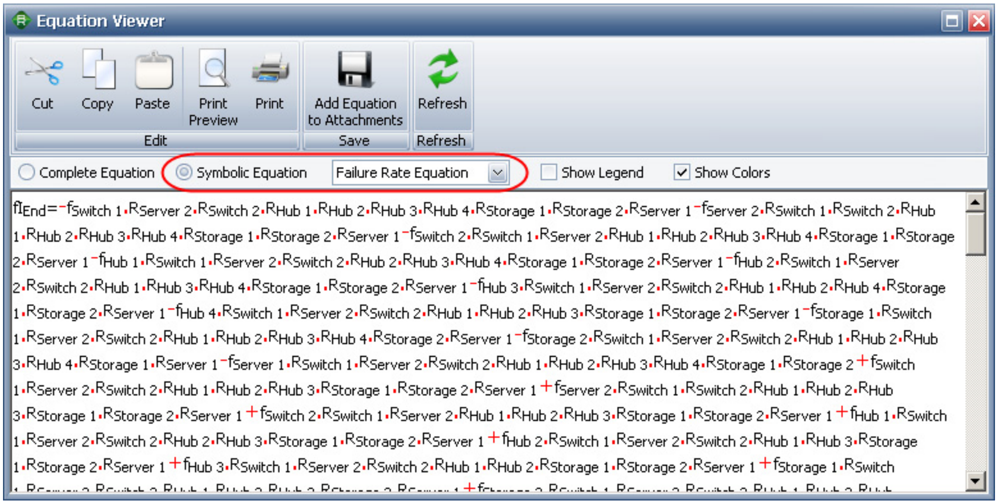
In the Equation Viewer, you select the **Complete Equation** option. The complete system reliability equation is displayed, as shown next.



You then select the **Symbolic Equation** option and choose **Pdf Equation** from the drop-down list to view the system *pdf* equation. Note that only part of the equation is shown next.

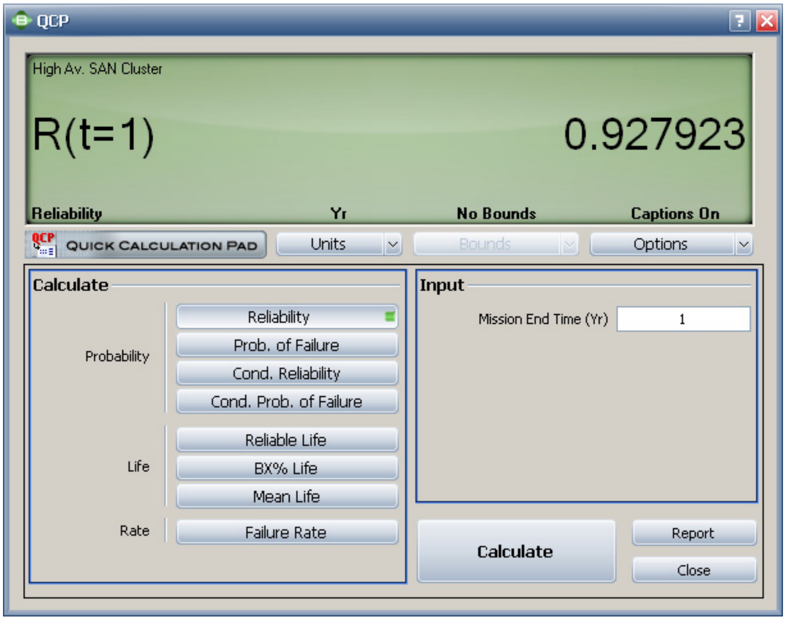


Finally, you choose Failure Rate Equation from the drop-down list to view the failure rate equation. Note that only part of the equation is shown next.



To determine the system reliability at one year of operation, you close the Equation Viewer and open the analytical QCP.

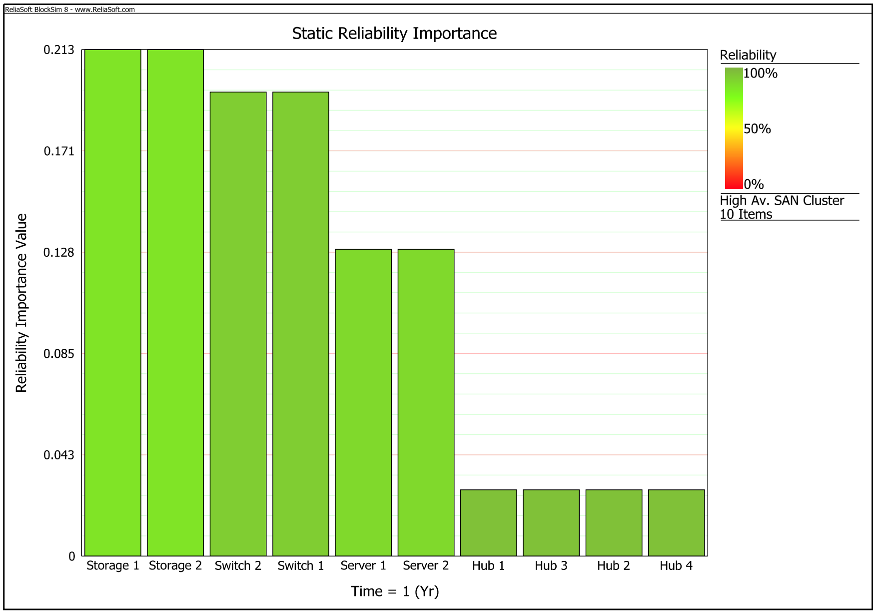
In the QCP, you select to calculate the **Reliability** and use the **Units** drop-down list to make sure the results will be returned in years. You enter a **Mission End Time** of **1** year. The system reliability at one year of operation is approximately 92.79%, as shown next.



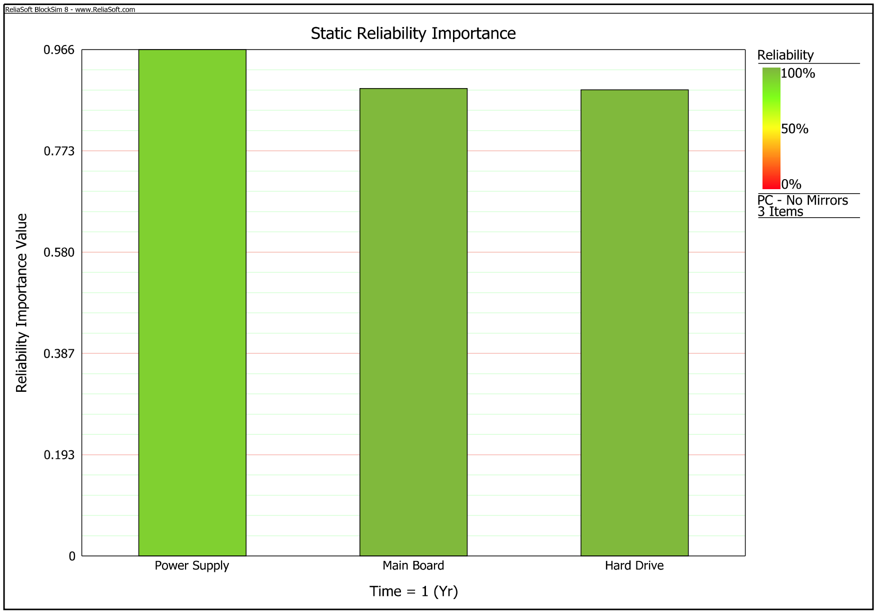
This does not meet the reliability requirement of 95% at one year of operation. Therefore, you need to identify what can and should be improved in order to achieve the requirement. As a starting point, you decide to look at the reliability importance of the components.

You click the **Plot** icon in the diagram control panel and then use the **Plot Type** drop-down list to choose the **Static RI** plot. In the **Units** field, you specify to use years in the plot, and enter **1** in the **At Time** field.

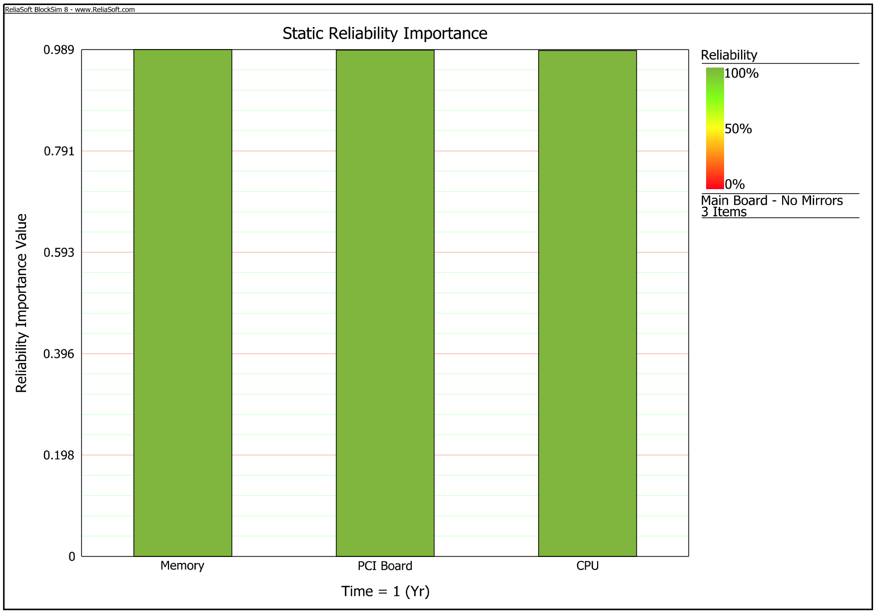
In the **Blocks** field, you specify to include all **10** blocks that have reliability information specified and then click the **Plot** icon again to redraw the plot.



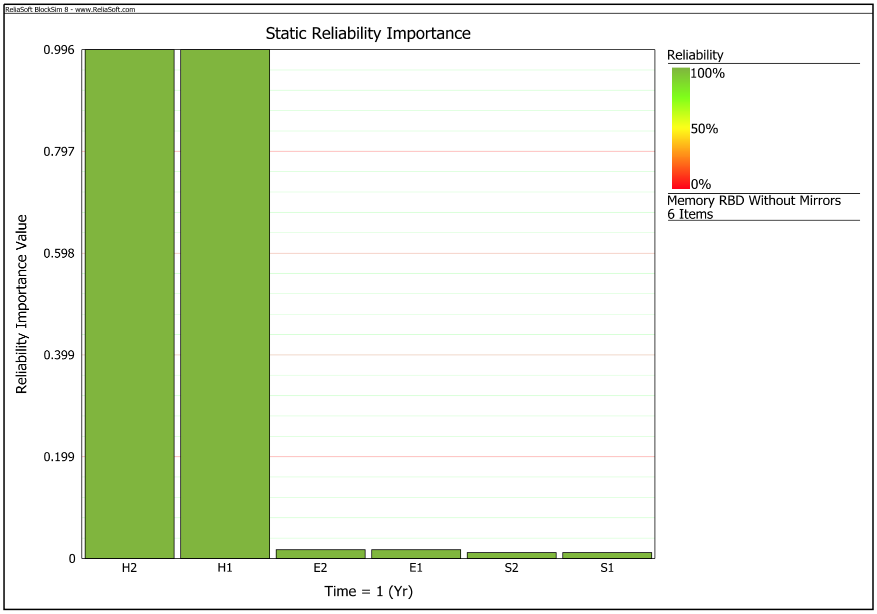
From this plot, you can see that the storage devices have the greatest effect on the system reliability, followed by the switches, then the servers and finally the hubs. You realize, however, that the servers are actually subsystems, so you also create a static reliability importance plot for the diagram that represents the servers (*"PC - No Mirrors" in the sample project*).



Following this logic, you also need to create a static reliability importance plot for the diagram that represents the main board in this diagram (*"Main Board - No Mirrors" in the sample project*).



In addition, you need to create a static reliability importance plot for the diagram that represents the memory chip in that diagram (*“Memory RBD Without Mirrors” in the sample project*).



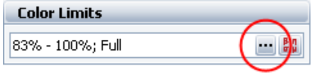
Using this many plots can make it difficult to get a sense of how the reliability importance values of all components relate. In lieu of the plots, you decide to create an analytical FRED report. Analytical FRED reports illustrate the reliability characteristics of a system's components in a flexible, color-coded format. This report can be used to easily identify the components most in need of improvement in order to achieve the desired system reliability.

You choose **Insert > Reports and Plots > FRED Report**.

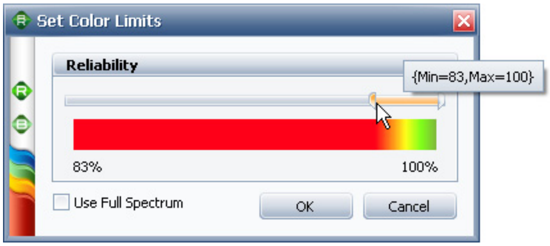
In the Select Diagram window that appears, you select your main diagram for the storage area network (*“High Av. SAN Cluster” in the sample project*).

When the FRED report is first created, it consists of a top level block that represents the system and, on the next level, a block for each of the 12 blocks in the diagram. Each standard block is marked with its reliability and reliability importance at the time specified in the control panel. In the **Reliability & Importance** area of the control panel, you change the FRED report to calculate at **1 Year** and then click **Recalculate** to recalculate the values for the blocks.

To make it easier to determine which blocks have room for improvement in their reliability, you click the **Adjust Color Limits** icon (**...**) in the Color Limits area of the control panel.



The blocks with the lowest reliability have a reliability of approximately 84%, so in the Set Color Limits window, you move the lower slider to 83%, as shown next.



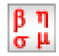
This changes the reliability range represented by the color scale, allowing you to easily see the relative reliabilities of the blocks.

To see the component of the subdiagrams, you right-click each subdiagram block (i.e., the server blocks) and choose **Append Level Below** on the shortcut menu. The complete FRED report allows you to see which blocks are most in need of improvement (i.e., which blocks have the lowest reliability) and which blocks will have the greatest effect on the reliability of the system if they are improved (i.e., which blocks have the highest reliability importance).

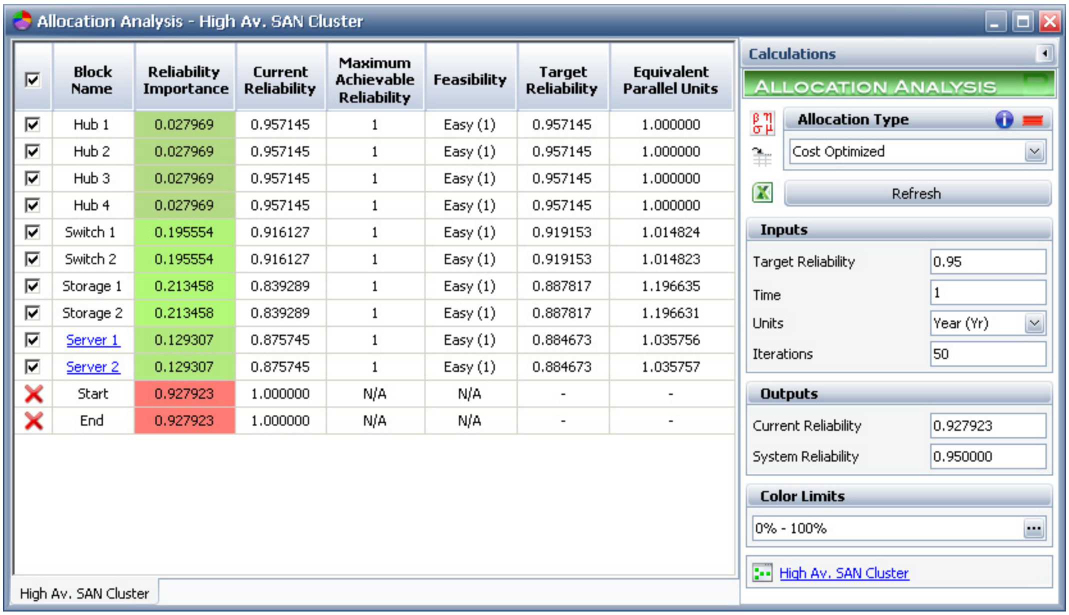
While these approaches have helped you to identify the reliability improvement areas, the next question is by how much should the reliability be increased. The Allocation Analysis utility is a powerful optimization utility that automatically determines which items need improvement and by how much.

You add an Allocation Analysis (*"Allocation Analysis - High Av. SAN Cluster" in the sample project*) to the project by choosing **Insert > Tools > Allocation Analysis**. In the Select Diagram window that appears, you select your main diagram for the storage area network (*"High Av. SAN Cluster" in the sample project*).

In the **Allocation Type** field on the control panel, you select to perform a **Cost Optimized** allocation. In the **Inputs** area, you specify a **Target Reliability** of **0.95** at a time of **1 Year**. The cost optimized allocation type allows you to meet the system's target reliability at the lowest possible cost. This allocation type determines the target reliability for each selected block based on the maximum reliability that can be achieved for the block and the relative difficulty or cost of increasing the block's reliability. In this example, you assume that all components can achieve 100% reliability (i.e., you enter **1** in the Maximum Achievable Reliability column in the data sheet). Additionally, you assume that all improvements are equally easy to make (i.e., you select **Easy (1)** in the Feasibility column). You then click the **Optimize** icon to perform the optimization.



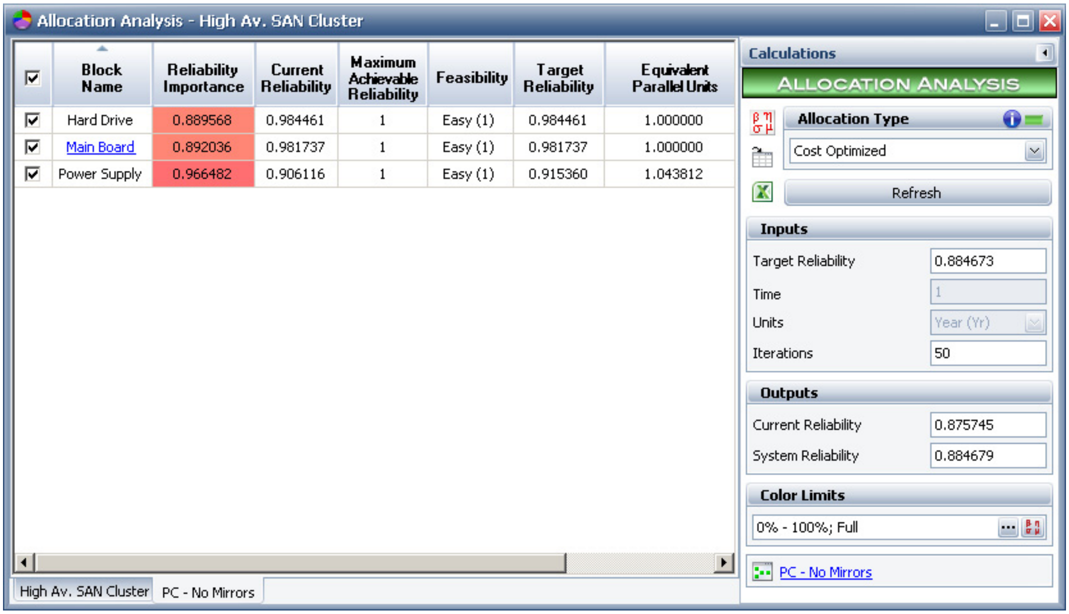
A target reliability value for each block is calculated, as well as a number of equivalent parallel units, which indicates the number of identical blocks that would be required in a parallel configuration for the particular block to meet the specified reliability goal, in lieu of increasing the component's reliability.



☑	Block Name	Reliability Importance	Current Reliability	Maximum Achievable Reliability	Feasibility	Target Reliability	Equivalent Parallel Units
☑	Hub 1	0.027969	0.957145	1	Easy (1)	0.957145	1.000000
☑	Hub 2	0.027969	0.957145	1	Easy (1)	0.957145	1.000000
☑	Hub 3	0.027969	0.957145	1	Easy (1)	0.957145	1.000000
☑	Hub 4	0.027969	0.957145	1	Easy (1)	0.957145	1.000000
☑	Switch 1	0.195554	0.916127	1	Easy (1)	0.919153	1.014824
☑	Switch 2	0.195554	0.916127	1	Easy (1)	0.919153	1.014823
☑	Storage 1	0.213458	0.839289	1	Easy (1)	0.887817	1.196635
☑	Storage 2	0.213458	0.839289	1	Easy (1)	0.887817	1.196631
☑	Server 1	0.129307	0.875745	1	Easy (1)	0.884673	1.035756
☑	Server 2	0.129307	0.875745	1	Easy (1)	0.884673	1.035757
✗	Start	0.927923	1.000000	N/A	N/A	-	-
✗	End	0.927923	1.000000	N/A	N/A	-	-

Because the Start and End blocks cannot fail, they are not included in the optimization. You notice that for each of the hub blocks, the target reliability value is the same as the current reliability value; these blocks do not need to be improved.

You now have target reliability values for each block. In order to reach the target reliability for the subdiagrams, you will need to perform allocation analysis for them so that you have target reliability values for each of their components. To do this, you click the Server 1 link in the Block Name column. Another sheet is added to the allocation analysis, with all of the components of the diagram that the Server 1 block represents (*"PC - No Mirrors" in the sample project*). You see that the **Target Reliability** field on the control panel is automatically populated with the target reliability for the server, as determined in the system-wide allocation analysis, so you click **Optimize**. You see that the only block that needs to be improved in the subdiagram is the Power Supply block.



Dependencies

5

In the preceding examples, we assume that redundant items are independent. This is not, however, always the case. Often, a system will include redundant items where the behavior of each one depends on the behavior of the other(s).

In this chapter, you will work with two types of dependencies:

- In load sharing configurations, two or more redundant items share the responsibility for keeping the system running properly. If one of them fails, the other(s) can take on an increased "load" so the system can continue to operate. Therefore, load sharing components exhibit different failure characteristics when one or more fail.
- In standby configurations, the redundant items are set to have two states: an active state and a standby state. Each state has a failure distribution defined. When an item in an active state fails, an item in a standby state is activated to take its place.

In this chapter

- ✓ Reliability block diagrams (RBDs)
- ✓ Load sharing configurations
- ✓ Standby configurations

ReliaSoft.TV

Watch a video of this example at http://www.reliasoft.tv/blocksim/startguide/blocksim_qsg_5.html.

5.1 Load Sharing

Consider the case of a dual battery power supply for a computer system. Both batteries are identical and, if working alone, each will power the system for an average time of 3 hours with a standard deviation of 15 minutes (i.e., the failure distribution is normal with a mean of 3 hours and std of 0.25 hours).

Assume that when the batteries are used together in parallel, you are doubling the amp hours for each and if one fails the system will still work.

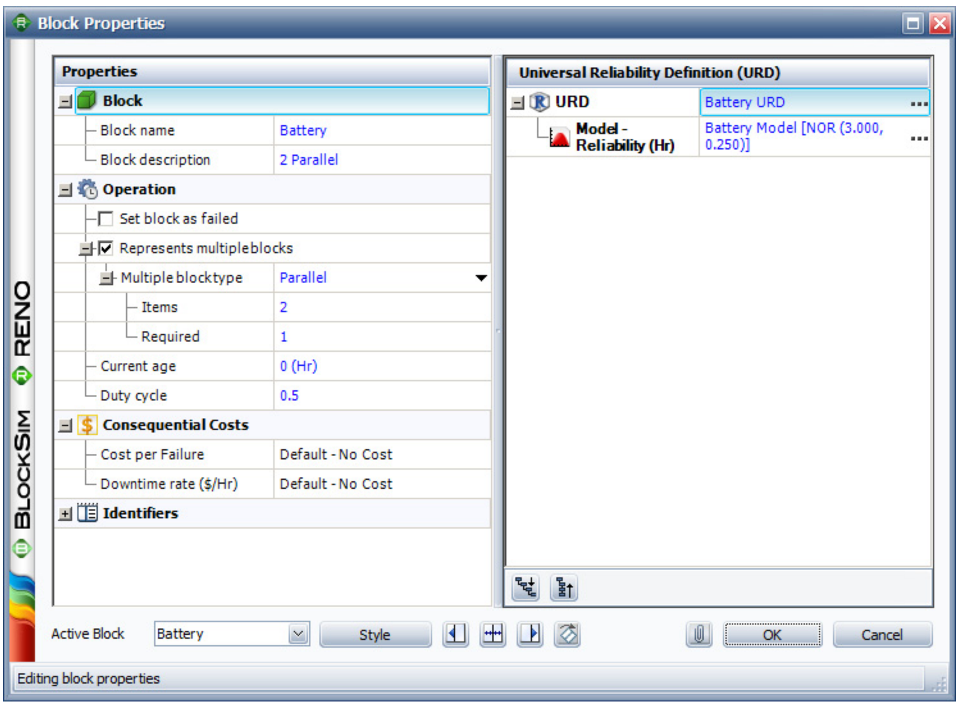
Objectives

- Create an RBD that accurately models this configuration.

Solution

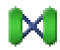
First, you create a new analytical RBD (*"Power Supply Parallel" in the sample project*). You add a standard block to the diagram. This block represents both batteries in a parallel configuration, so you configure it as follows:

- Universal Reliability Definition area:
  - Reliability: Use the Battery URD and the Battery Model (normal, mean time = 3 hours, std = 0.25 hours)
- Properties area:
  - Select **Represents multiple blocks**
    - Multiple block type: **Parallel**
      - Items: **2**
      - Required: **1**
  - Duty cycle: **0.5**



You analyze the diagram. However, you realize that there is a problem with this approach: The 50% duty cycle ensures that when both batteries are operating, they are doing so at half of their rated load and are therefore sharing the load equally. However, there is no change in the duty cycle if one of the batteries fails. Given the current settings, when one battery fails, the surviving battery will continue to operate at the defined 50% duty cycle. This does not account for the fact that the surviving battery would actually need to operate at a 100% duty cycle to handle the entire load. To address this issue of dependency, you need to use a load sharing container.

You create another new analytical RBD (*"Power Supply Load Sharing" in the sample project*) and add a load sharing container block to it by choosing **Diagram > Blocks > Add Load Sharing Container**.

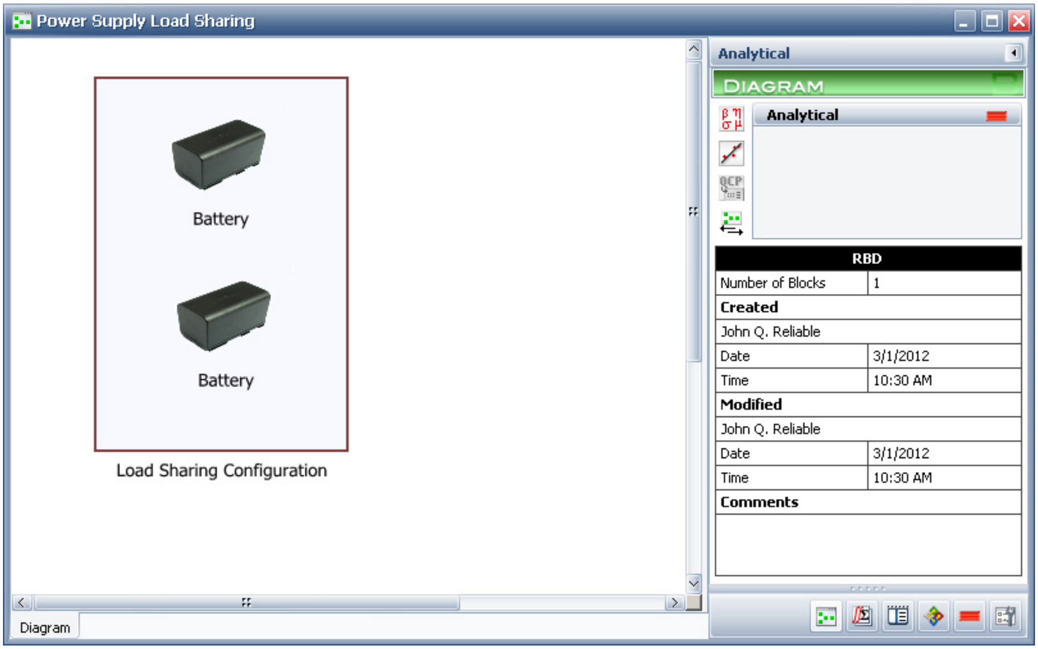


In the Block Properties window for the load sharing container block, you enter **1** in the **Number of paths required** field. This indicates that at least 1 block (or path) within the container must succeed in order for the container to succeed.

You then select the load sharing container and add two standard blocks to it, thereby making them contained blocks. These blocks represent the two batteries. You configure each of the contained blocks as follows:

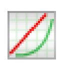
- Universal Reliability Definition area:
  - Reliability: Use the Battery URD and the Battery Model (normal, mean time = 3 hours, std = 0.25 hours)
- Properties area:
  - Weight proportionality factor: **1**
  - Duty cycle: **1**

The weight proportionality factor for each block is compared against the weight proportionality factor of each other block in the load sharing container to determine how the load is shared among the blocks. (For example, if a container holds two blocks and one has a factor of 3 while the other has a factor of 9, then the first block will receive 25% of the load (3/12) and the second block will receive 75% of the load (9/12).) In this case, because both contained blocks have the same weight proportionality factor, they will share the load evenly when both are operating. If either one fails, the other will then take over 100% of the load.



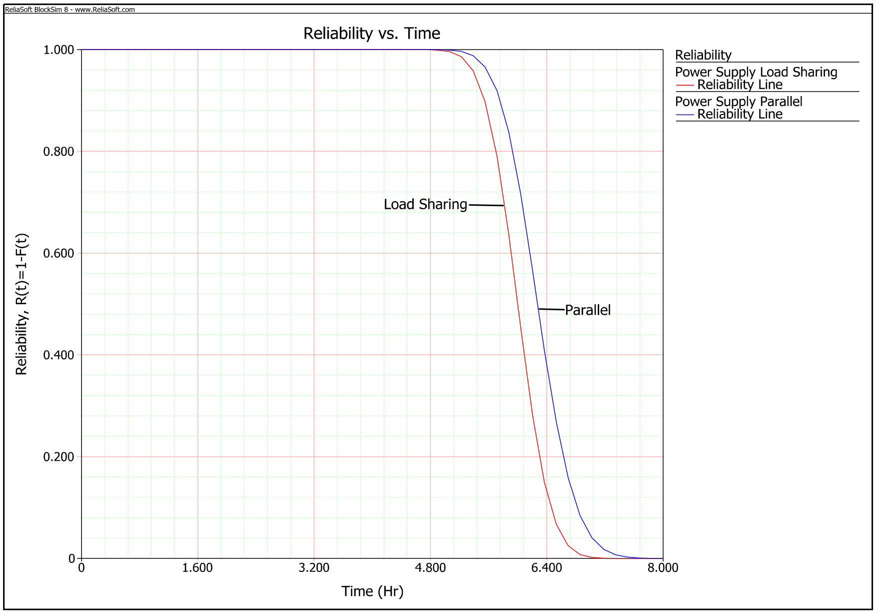
You analyze the diagram.

To understand how the diagrams differ in their results, you create an overlay plot by choosing **Insert > Reports and Plots > Overlay Plot**.



When prompted to select which diagrams to plot, you select the parallel configuration diagram and the load sharing configuration diagram (*"Power Supply Parallel" and "Power Supply Load Sharing" in the sample project*).

On the control panel of the plot sheet, you switch the plot type to a **Reliability vs. Time** plot. The plot (*"Overlay - Parallel and Load Sharing" in the sample project*) appears as shown next (with annotations added via RS Draw to make the plots easier to interpret).



You see that the parallel configuration yields more optimistic results, predicting a higher reliability for the power supply.

5.2 Standby

Another way to configure the power supply is to have only one battery operating at a time. In this case, one battery has the full load and when it fails the other battery continues with the full load. This is known as standby redundancy.

Objectives

- Create an RBD that accurately models this configuration.
- Compare the results to the parallel and load sharing configurations.

Solution

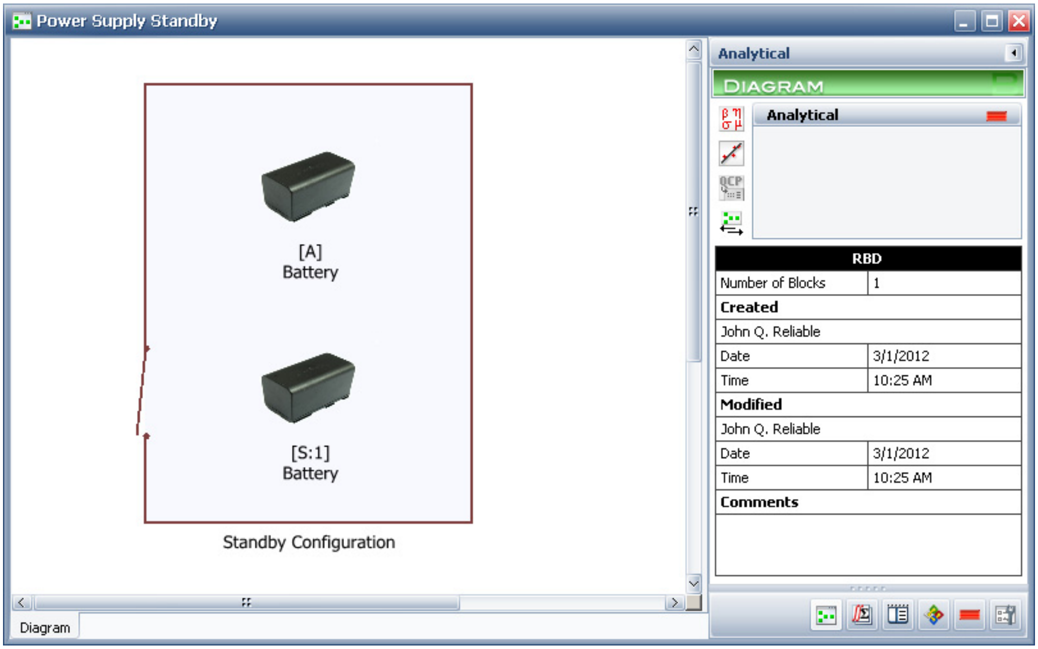
You create a new analytical RBD (*"Power Supply Standby" in the sample project*) and add a standby container block to it by choosing **Diagram > Blocks > Add Standby Container**.

In the Block Properties window for the standby container block, you enter **1** in the **Number of active paths required** field. This indicates that there must be 1 active block (or path) that succeeds within the container in order for the container to succeed.

You then select the standby container and add two standard blocks to it, thereby making them contained blocks. These blocks represent the two batteries. You configure one of the contained blocks as follows:

- Universal Reliability Definition area:
  - Reliability: Use the Battery URD and the Battery Model (normal, mean time = 3 hours, std = 0.25 hours)
- Properties area:
  - Duty cycle: **1**
  - Active/Standby: **Active**

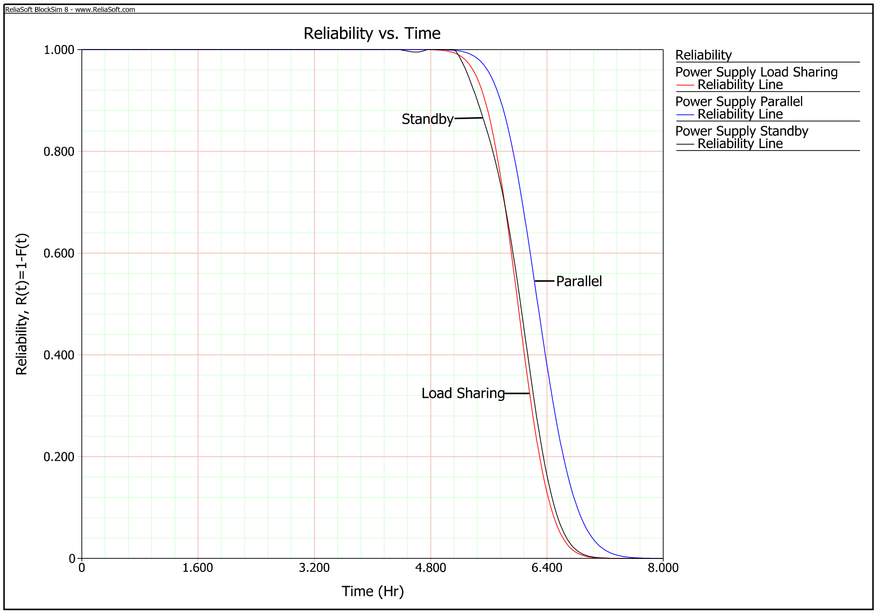
You configure the other contained block in the same way, except that its status is **Standby**. The diagram is shown next. Note that the block marked with [A] is in an active state and the block marked with [S:1] is the first priority block in the standby state (priority indicates the order in which contained standby blocks are used if there is more than one).



You then analyze the diagram.

You create another overlay plot. When prompted to select which diagrams to plot, you select the parallel configuration diagram, the load sharing configuration diagram and the standby configuration diagram (*"Power Supply Parallel," "Power Supply Load Sharing" and "Power Supply Standby" in the sample project*).

On the control panel of the plot sheet, you switch the plot type to a **Reliability vs. Time** plot. The plot (*"Overlay - Parallel, Load Sharing and Standby" in the sample project*) appears as shown next (with annotations added via RS Draw to make the plots easier to interpret).



You see that the results for the load sharing configuration and the standby configuration are nearly identical in this case.

Note: The minor differences between the load sharing configuration and the standby configuration are a result of varying numerical convergence settings. If you are creating this example rather than working with the provided repository, you may need to adjust the numerical convergence settings for the standby container to achieve a smooth plot. Instructions on adjusting numerical convergence settings are provided in the online help at: http://help.synthesis8.com/blocksim_reno9/index.htm#algorithm_setup_window.htm.

Simulation Diagrams 6

In contrast to analytical diagrams, simulation diagrams can take into account repair and restoration actions, including behaviors of crews, spare part pools, throughput, etc. These actions mean that the age of system components is no longer uniform, nor is the operation time of the system continuous. If one includes information on the repair and maintenance characteristics of the components and resources available in the system, other information can also be analyzed/obtained, such as system availability, throughput, spare parts utilization, life cycle costs, etc. This can be accomplished through discrete event simulation.

In this chapter

- ✓ Reliability block diagrams (RBDs)
- ✓ Simulation diagrams
- ✓ Publishing models
- ✓ Corrective tasks
- ✓ Crews
- ✓ Spare Part Pools
- ✓ Resource Manager
- ✓ Reports - Analysis Workbooks
- ✓ Function Wizard

ReliaSoft.TV Watch a video of this example at http://www.reliasoft.tv/blocksim/startguide/blocksim_qsg_6.html.

6.1 Maintenance, Availability and Costs

Looking again at the storage area network, let's assume that it will be deployed at the current reliability (i.e., the targets computed in Chapter 4 were not implemented).

When a component of the network fails, it needs to be repaired or replaced. In this example, the primary objective is the restoration of the network; thus, failed components are replaced with new ones and no troubleshooting is done. In other words, all components, including the server, will be looked at as line replaceable units (LRUs) that get replaced upon failure.

Your company has IT coverage 24 hours a day, 7 days a week, with a single IT person on-site at all times. The IT technician replaces the components as they fail, and can swap out any of the components in about an hour with a standard deviation of 10 minutes. The hourly rate for IT is $20/hour.

Spare parts are not kept on hand, but must be ordered. It takes an average time of 6 days with a standard deviation of 2 days for a part to arrive. The costs for replacement parts, including shipping, are as follows:

- Server: $3,000
- Hub: $1,500
- Switch: $3,000
- Storage: $4,000

Objectives

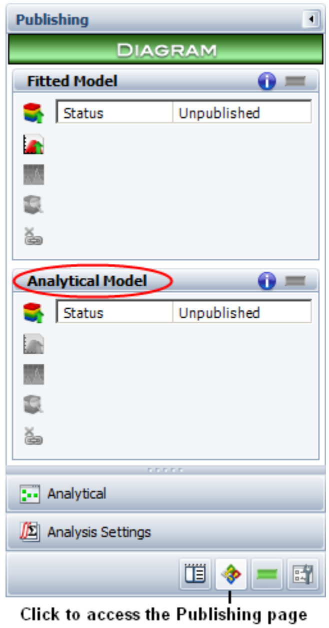
- Create an RBD that accurately models this configuration, including repair properties.
- Determine the costs associated with maintenance over a period of five years.

Solution

First, you need to create a model that represents the server, so that you can treat the server as an LRU. You want to do this by publishing the results from the server diagram as a model, to make use of the work that you have done. You consider two possible approaches to this:

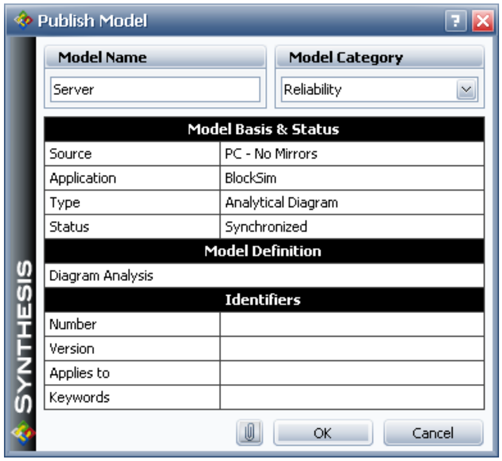
- You can create a model based on the analytical solution for the server diagram (*"PC - No Mirrors" in the sample project*). In effect, this creates a single reliability model object that references all items in the diagram, as well as their dependents. The benefit to this approach is that no approximation is needed and any change made to the models or dependencies of the blocks within it will be automatically incorporated in the model. The drawback is that such object models are complex and consume computer memory and resources. Each reference to a model like this is really a reference to the diagram and its subdiagrams, and uses resources accordingly.
- You can create a model based on the fitted distribution for the server diagram. This approach yields an approximation of the entire diagram (and its subdiagrams). This results in a significant reduction of complexity, as the entire diagram and its dependents no longer need to be used for any calculations. The drawback is that any changes that are subsequently made to the diagram and/or its dependents are not reflected unless the distribution is refitted and then republished.

Since this case is fairly simple, you decide to go ahead and publish the analytical model for the server diagram (*"PC - No Mirrors" in the sample project*). To do this, you analyze the diagram, then go to the Publishing page of the diagram's control panel and look at the **Analytical Model** section, as shown next.

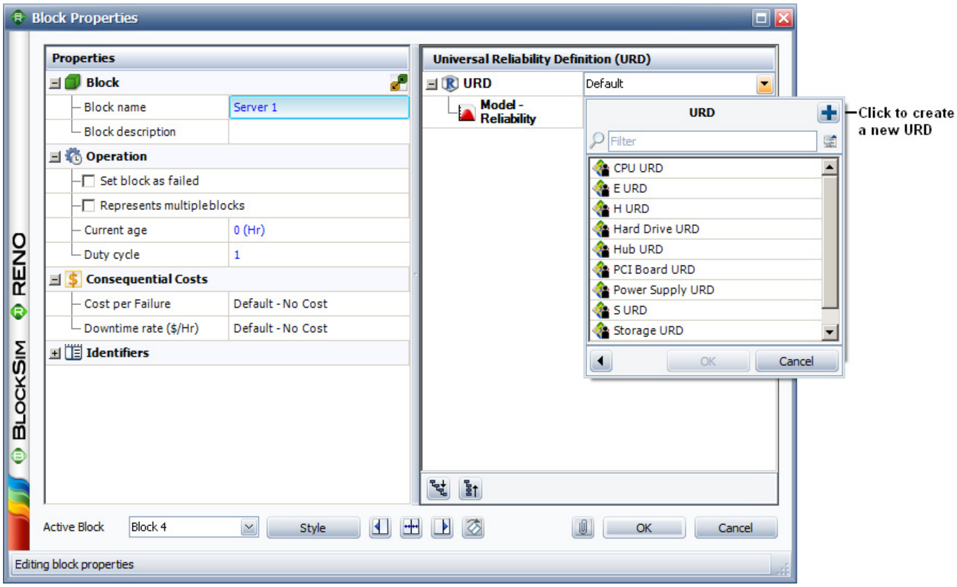


You then click the **Publish to Model** icon in the Analytical Model section.

In the Publish Model window that appears, you name the new model (*"Server" in the sample project*) and click **OK** to publish the model.



You then duplicate the existing storage area network diagram (*"High Av. SAN Cluster" in the sample project*). In the duplicate diagram (*"SAN with Maintenance" in the sample project*), you right-click the Server 1 block and choose **Convert Block To > Standard Block** on the shortcut menu. You open the Block Properties window for the Server 1 block and create a new universal reliability definition (URD) for the block by clicking the **URD** field, clicking the arrow that appears in the field and then clicking the **Create New** icon in the URD Wizard that appears.



You name the URD **Server URD** and assign the Server model to it.

You then convert the Server 2 block to a standard block and define its reliability using the Server URD.

If you were to analyze the diagram at this point, the results obtained in this model would be identical to the results obtained in the original model. However, you want to add maintenance information to the diagram, so you convert the diagram to a simulation RBD by clicking the **Change Diagram Type** icon on the control panel.

You can now set the repair properties for the components in the diagram. Note that you will need to edit the URD for each type of block (server, hub, switch and storage device) only once, because the changes that you make to a URD affect all blocks that use that URD.

You start by returning to the Block Properties window for the Server 1 block. You click the **URD** field, click the arrow that appears in the field and then click the **View/Edit** icon in the URD Wizard that appears.

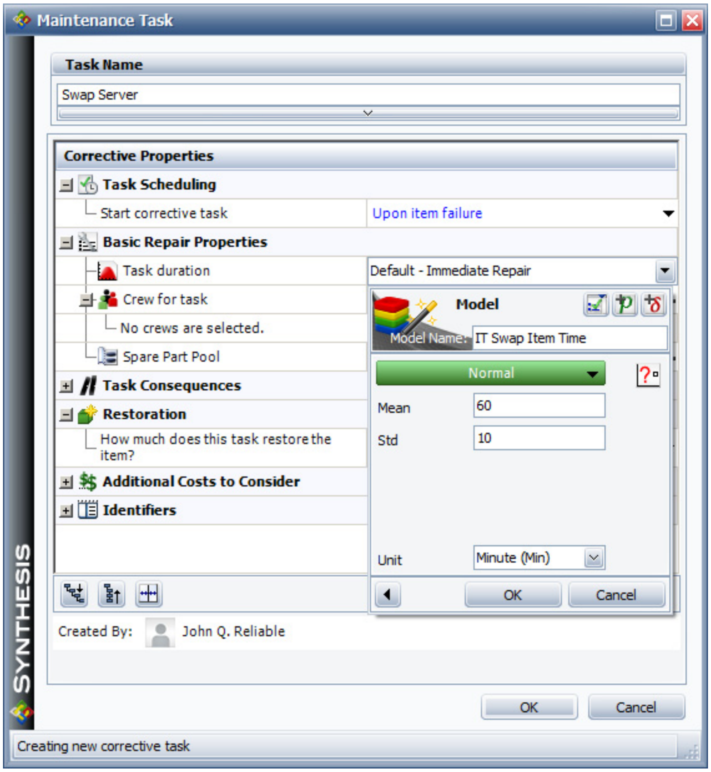
In the Universal Reliability Definition window, you click in the **Corrective Task** field and add a new task.

In the Maintenance Task window, you name the task **Swap Server** and set the following properties:

- Task Scheduling area - Start corrective task: **Upon item failure**
- Restoration area - How much does this task restore the item? **To as good as new condition**

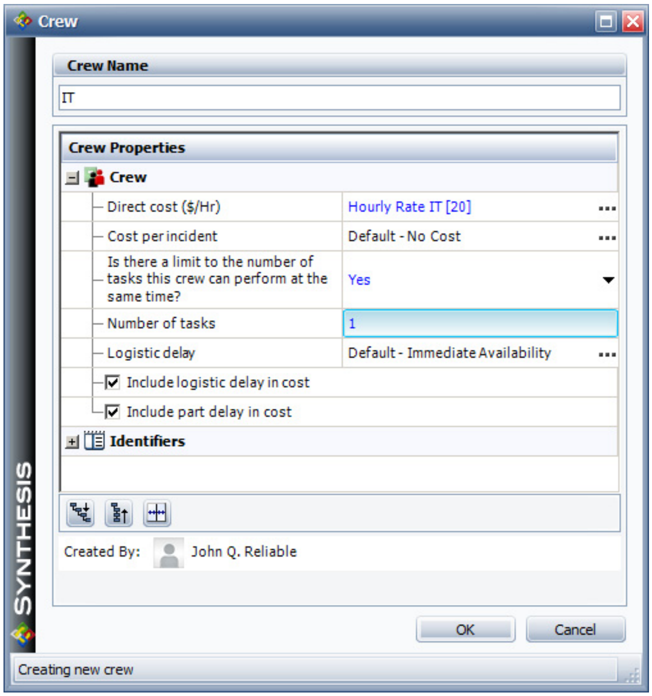
To assign a model for the task duration, you click in the **Task duration** field in the Basic Repair Properties area and create a new distribution model with the following properties:

- Model Name: **IT Swap Item Time**
- Distribution: **Normal**
- Mean: **60**
- Std: **10**
- Unit: **Minute**



To assign a crew to perform the task, you click in the **Crew for task** field in the Basic Repair Properties area of the Maintenance Task window and create a new crew with the following properties:

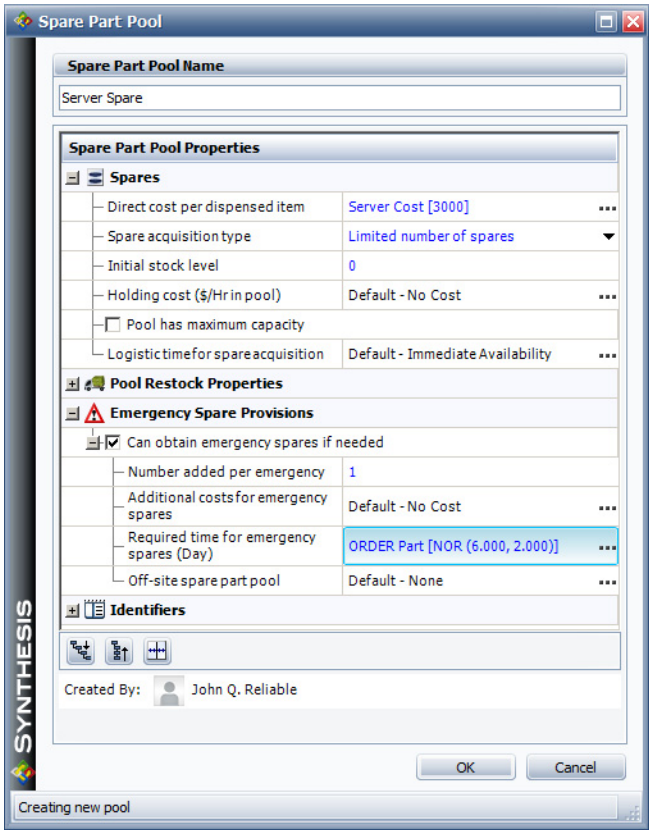
- Crew Name: **IT**
- Direct cost: Create a new constant model named **Hourly Rate IT**, using a Constant Cost of **20** and hours for the units (i.e., Currency per Unit).
- Number of tasks that the crew can perform at the same time limited to **1**



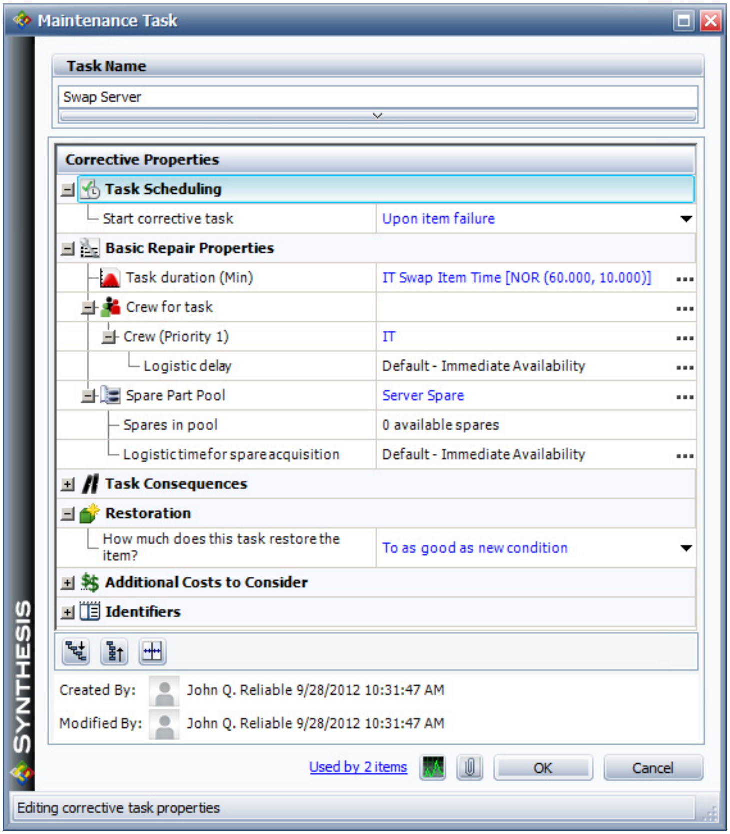
To assign a spare part pool to specify the times and costs surrounding spare parts for the task, you click in the **Spare part pool** field of the Maintenance Task window and create a new spare part pool with the following properties:

- Spare Part Pool Name: **Server Spare**
- Spares area:
    - Direct cost per dispensed item: Create a new constant model named **Server Cost**, using a Constant Cost of **3,000**.

- Spare acquisition type: **Limited number of spares**
  - Initial stock level: **0**
- Emergency Spare Provisions area:
  - Can obtain emergency spares if needed: **Yes**
    - Number added per emergency: **1**
    - Required time for emergency spares: Create a new distribution model named **ORDER Part,** using a normal distribution with Mean = 6 days and Std = 2 days.



The Swap Server task is now fully defined.



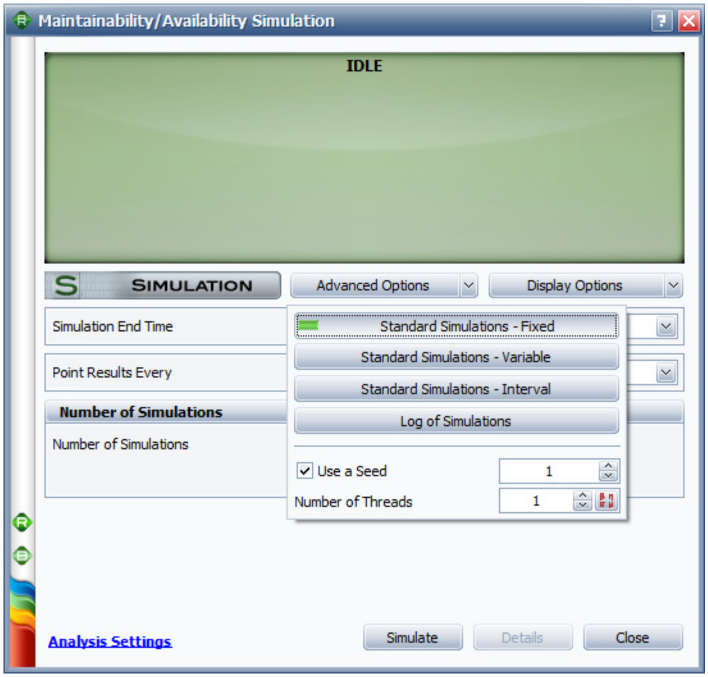
As mentioned earlier, the changes that you made affect the URD everywhere it is used, including in the original diagram. However, in this case, you added only maintenance properties. These are not used in analytical diagrams, so the changes do not affect the original diagram.

You follow similar steps to add maintenance information for the hubs, switches and nodes, adding the following to their URDs:

- For hubs, a corrective task called **Swap Hub**, using:
  - The **IT Swap Item Time** model for task duration
  - The **IT** crew
  - A spare part pool called **Hub Spare**, using:
    - A constant model (cost = $1,500) called **Hub Cost** for the direct cost per dispensed item
    - The **ORDER Part** model for the time required for emergency spares
- For switches, a corrective task called **Swap Switch**, using:
  - The **IT Swap Item Time** model for task duration
  - The **IT** crew
  - A spare part pool called **Switch Spare**, using:
    - A constant model (cost = $3,000) called **Switch Cost** for the direct cost per dispensed item
    - The **ORDER Part** model for the time required for emergency spares
- For storage devices, a corrective task called **Swap Storage**, using:
  - The **IT Swap Item Time** model for task duration
  - The **IT** crew
  - A spare part pool called **Storage Spare**, using:
    - A constant model (cost = $4,000) called **Storage Cost** for the direct cost per dispensed item
    - The **ORDER Part** model for the time required for emergency spares

Once you have defined all of the maintenance information for the diagram, you simulate the diagram by clicking the **Simulate** icon on the control panel.

In the Maintainability/Availability Simulation window, you click the **Advanced Options** drop-down and select **Standard Simulations - Fixed** to perform a fixed number of simulations. In addition, you specify to use a seed of 1, and a single thread.

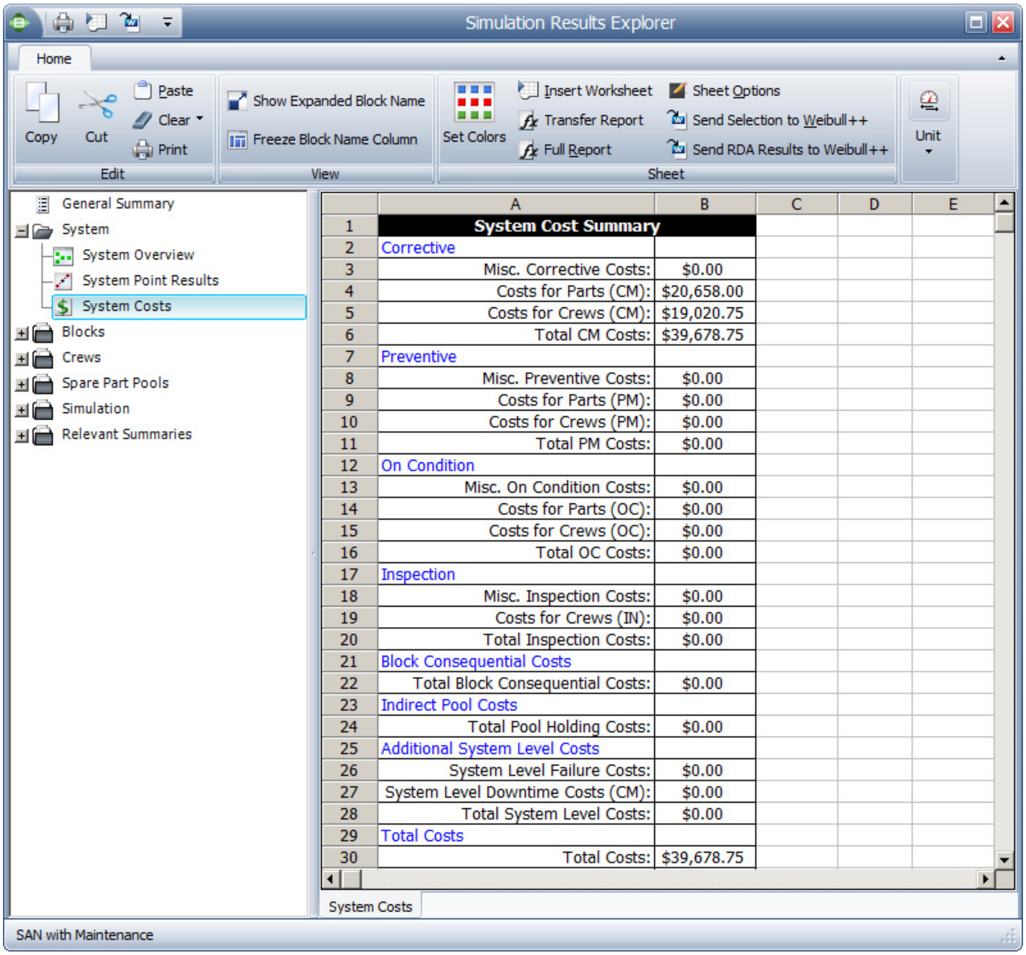


Seeds: Results obtained through repeated simulations of the same diagram will exhibit some variability because they depend on the randomly generated numbers that are selected for any given simulation run. A seed is a starting point from which the random numbers used in simulation will be generated; this forces the software to use the same sequence of random numbers in each simulation, resulting in repeatability.

Threads: Running simulations in multiple threads may improve performance and save time when simulating complex diagrams. The number of threads that should be used depends upon your computer's specifications. The number of threads you use affects repeatability of results. In order for your results to be repeatable, you must use the same number of threads and the same seed each time you simulate the diagram.

You then enter a **Simulation End Time** of **5** years and specify in the **Number of Simulations** field to run **1,000** simulations, then click **Simulate**.

When simulation is finished, you click the **Details** button to view the results in the Simulation Results Explorer. Because you are interested in exploring the costs, you select the System Costs worksheet in the navigation panel on the left.



You see that over the simulated 5 years of operation, the parts to maintain the storage area network cost $20,658.00 and crew costs amounted to $19,020.75, resulting in a total cost of $39,678.75. However, you also realize that the crew costs shown here reflect only those times when IT was actually engaged in fixing the storage area network. Since there is an IT technician on duty at all times, the actual labor cost (ignoring benefits, vacation, etc.) of this 24x7 staffing is $20/hour x 24 hours/day x 365 days/year x 5 years = $876,000. Adding that to the parts cost, you get a total cost of $896,658.00 over the 5 years.

Note: If you created this diagram on your own rather than using the sample project, your simulation results may vary slightly from those shown here. Using a seed ensures repeatability of results for a given diagram by generating the same stream of random numbers. However, the order of the utilizations of the stream may differ depending on the way the diagram was constructed. For example, two people may have projects containing identical diagrams using blocks A and B. If in one diagram, A was created before B, A will take the first random value from the stream and B will take the second. If the other person (in the other project) created the same diagram but created B before A, B will take the first random value from the stream and A will take the second, thus yielding potentially different answers, even though both analysts may have used the same seed in simulation.

6.2 Cost Comparison

Given the high labor cost, your company decides to outsource maintenance of the storage area network. The cost for this will be $1,000 per call plus $250 per hour. The response time for the outsource crew is 10 days with a standard deviation of 1 day, which includes the time that it takes to order the part.

Objectives

- Determine the costs associated with maintenance over a period of five years.
- Compare these costs to those associated with using in-house IT support.

Solution

Rather than go through all the work of defining maintenance properties again, you decide to copy the diagram that you have been working on. Knowing that you will need to make changes to some of the resources, however, you create copies of the resources at the same time so that your changes do not affect the original diagram. To do this, you choose the diagram to be copied (*"SAN with Maintenance" in the sample project*) in the current project explorer and choose **Project > Current Item > Duplicate > Duplicate With Resources**.

You then begin to make the necessary changes to the resources used by the current diagram (*"SAN - Outsourced Maintenance" in the sample project*). First, you open the Block Properties window for the Server 1 block and then open the associated URD. In this window:

- You decide to change the name of each resource that you edit to use "OUT" (for "outsourced") instead of "_1" so that it is easier to tell where the resources are being used. Therefore, you rename the URD to **Server URD OUT**.
- You click in the **Model** field, click the arrow that appears in the field and then choose the Server model instead of the Server_1 model. The reason for this is that only one model can be linked to

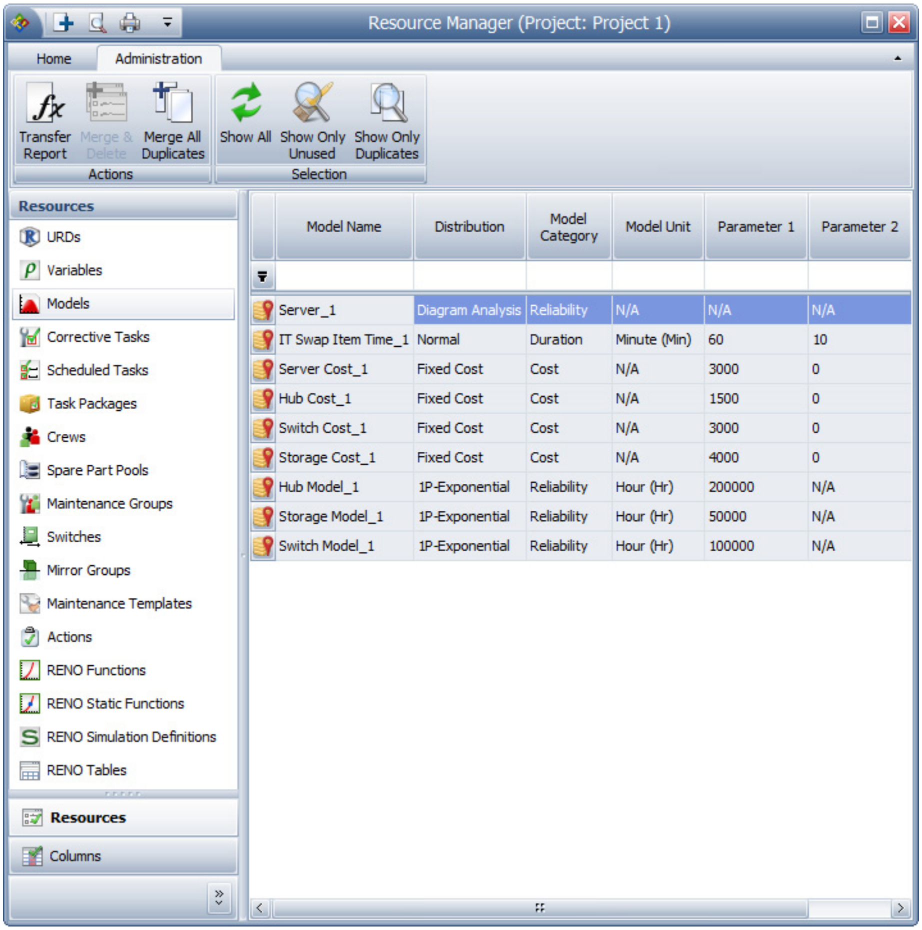
the original data source. Copies of published models are not linked, so any changes that are made to the data source will not affect the copy. You want this diagram to reflect any changes to the server diagram.

- You open the associated corrective task. In this window, you do the following:
    - Rename the task to **Swap Server OUT**.
    - Change the **Task duration** field to use the original IT Swap Item Time model instead of the copy, since they are the same, and make a mental note to delete the unnecessary model. This will ensure that any changes you make to the model will apply to both the original scenario and the outsourcing scenario.
    - Open the associated crew. In this window, you do the following:
        - Rename the crew to **Outsource Crew**.
        - **Direct cost**: You rename the constant model to **Hourly Rate OUT**, using a **Constant Cost** of 250 and hours for the units.
        - **Cost per incident**: You create a new constant model named **OUT Cost per Incident**, using a Constant Cost of 1,000.
        - You remove the limit to the number of tasks that the crew can perform at the same time.
        - **Logistic delay**: You create a new distribution model named **OUT Delay** using a normal distribution with mean = 10 days and std = 1 day.
    - Open the associated spare part pool. In this window, you do the following:
        - Rename the spare part pool to **Server Spare OUT**.
        - Change the **Direct cost per dispensed item** field to use the original Server Cost model instead of the copy, and make a mental note to delete the unnecessary model.
        - Change the **Spare acquisition type** field to use Unlimited spares.

You follow similar steps to update the maintenance information for the hubs, switches and nodes.

You then choose **Project > Synthesis > Resource Manage**r to use the Resource Manager to delete the unused models from the project.

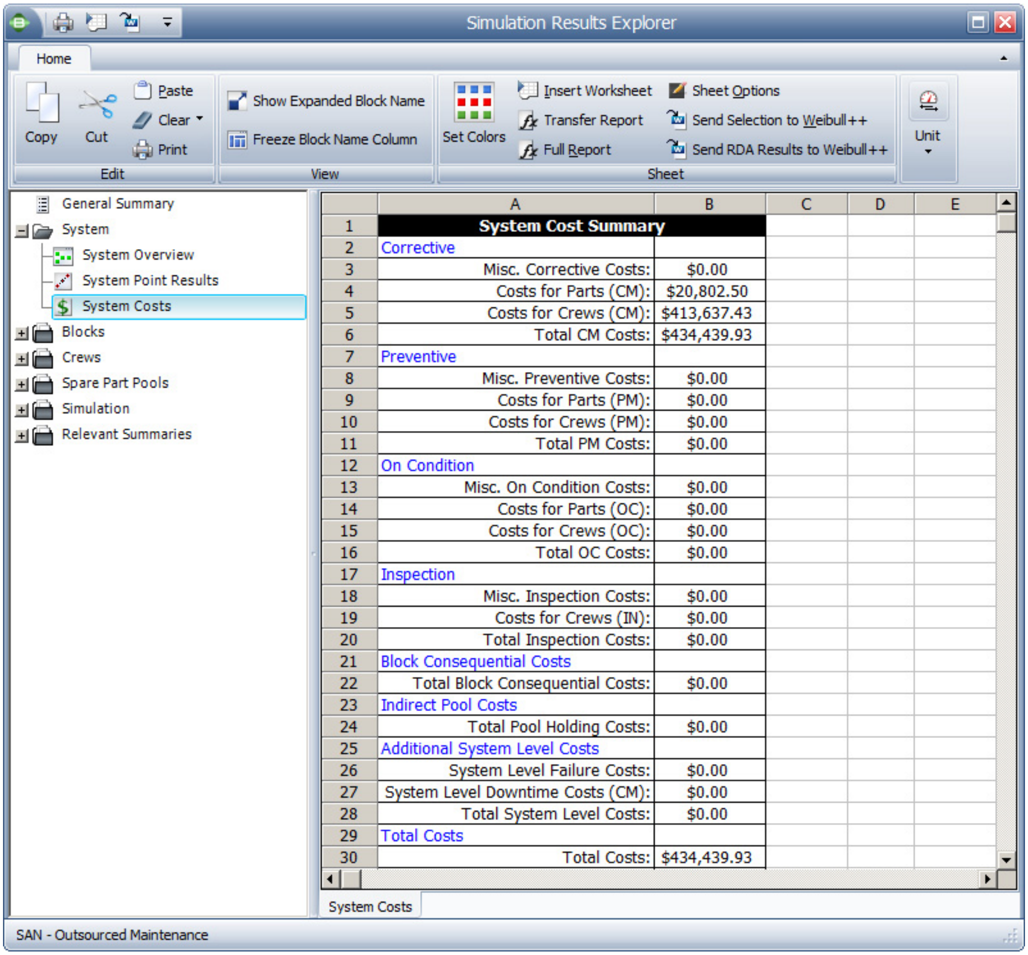
You click **Models** in the navigation panel on the left and then choose **Administration > Selection > Show Only Unused** to see only the models that are not used by any other resources.



You select all of the models in the list by selecting the first one, holding down the **SHIFT** key and selecting the last one, then choose **Home > Edit > Delete**.

After closing the Resource Manager, you simulate the current diagram using the same simulation settings that you used to simulate the original simulation RBD (as given on page 75).

When simulation is finished, you view the System Costs worksheet in the Simulation Results Explorer.



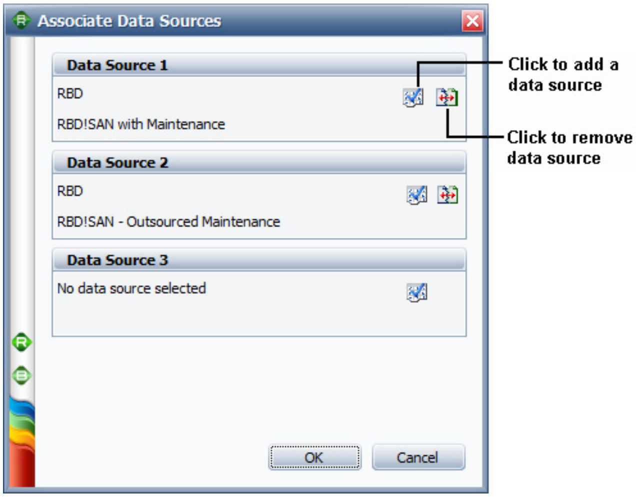
You see that over 5 years' worth of simulated operation, parts to maintain the storage area network cost \$20,802.50 and crew costs amounted to \$413,637.43, resulting in a total cost of \$434,439.93. Clearly, this is considerably less than the \$896,658.00 costs for using in-house IT support over 5 years.

To create an analysis workbook that will give you a better sense of how the costs associated with these scenarios compare, you choose **Insert > Reports and Plots > Analysis Workbook**.

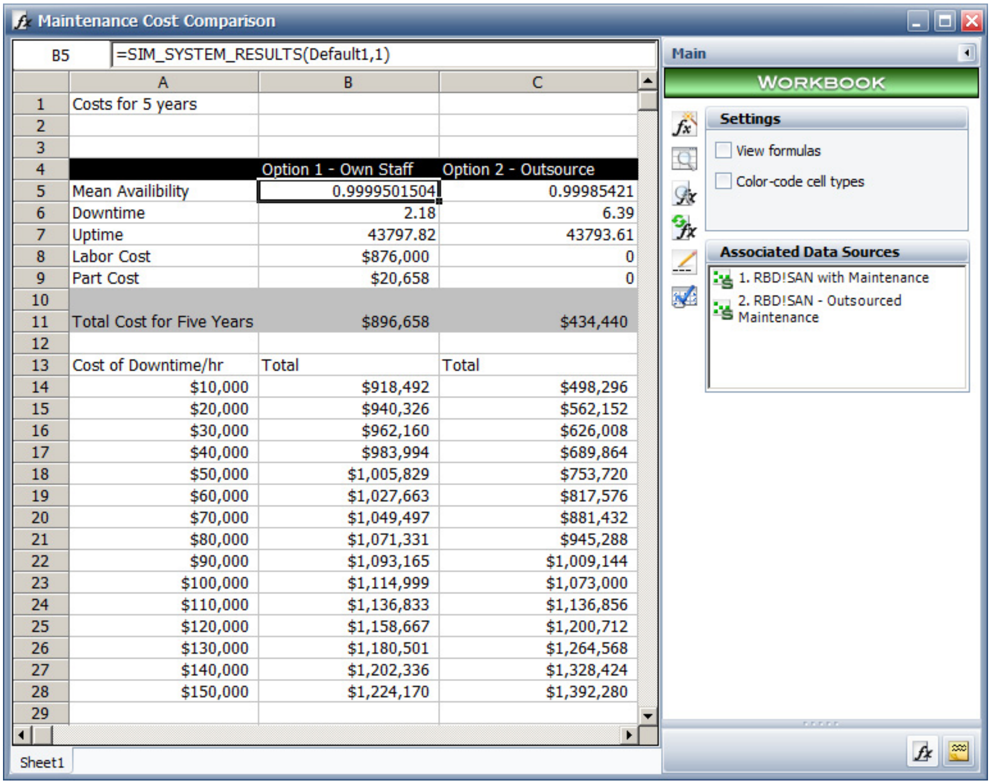
In the Report Wizard, you click **Select** to specify a default data source and choose the diagram of the in-house scenario (*"SAN with Maintenance" in the sample project*), then click **OK** to create a blank workbook.

On the control panel, you click the **Associate Data Sources** icon to add the diagram of the outsourced scenario as another default data source.

The Associate Data Sources window is shown next, with the outsourced scenario (*"SAN - Outsourced Maintenance" in the sample project*) already added as the second data source.



You build a worksheet like the one shown next (*"Maintenance Cost Comparison" in the sample project*) by using the Function Wizard to build functions that automatically insert calculated results based on the specified data source.

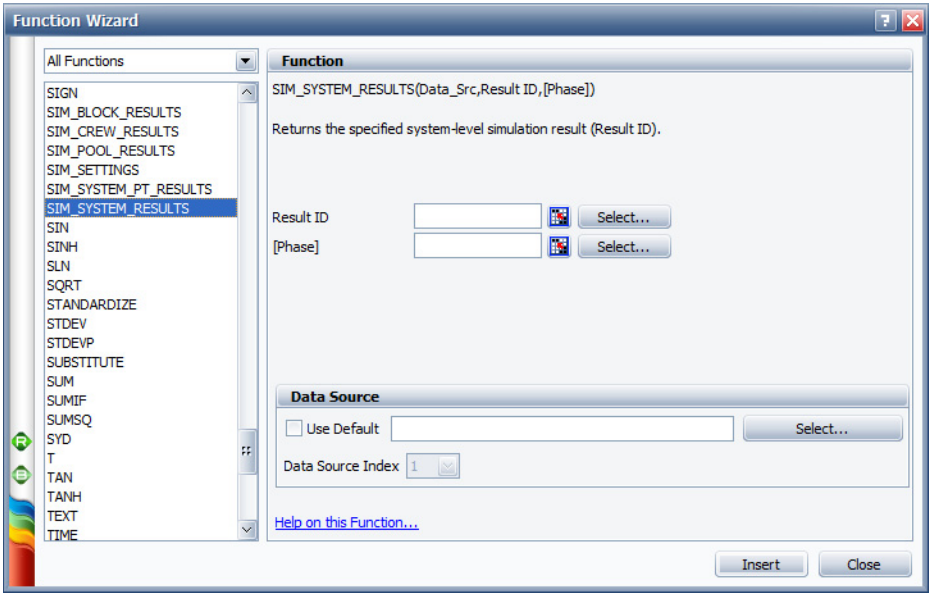


	A	B	C
1	Costs for 5 years		
2			
3			
4		Option 1 - Own Staff	Option 2 - Outsource
5	Mean Availibility	0.9999501504	0.99985421
6	Downtime	2.18	6.39
7	Uptime	43797.82	43793.61
8	Labor Cost	$876,000	0
9	Part Cost	$20,658	0
10			
11	Total Cost for Five Years	$896,658	$434,440
12			
13	Cost of Downtime/hr	Total	Total
14	$10,000	$918,492	$498,296
15	$20,000	$940,326	$562,152
16	$30,000	$962,160	$626,008
17	$40,000	$983,994	$689,864
18	$50,000	$1,005,829	$753,720
19	$60,000	$1,027,663	$817,576
20	$70,000	$1,049,497	$881,432
21	$80,000	$1,071,331	$945,288
22	$90,000	$1,093,165	$1,009,144
23	$100,000	$1,114,999	$1,073,000
24	$110,000	$1,136,833	$1,136,856
25	$120,000	$1,158,667	$1,200,712
26	$130,000	$1,180,501	$1,264,568
27	$140,000	$1,202,336	$1,328,424
28	$150,000	$1,224,170	$1,392,280
29			

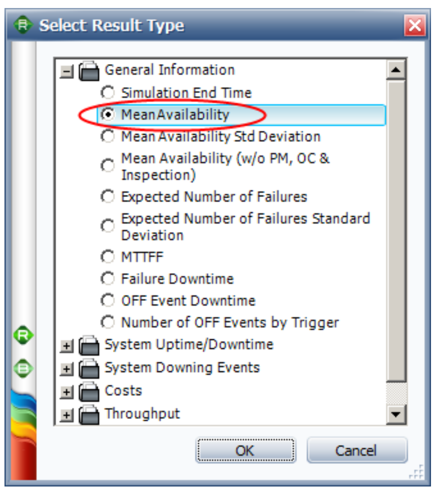
The lower portion of the report builds on the results from the diagrams to determine how the 5-year costs would be affected by increasing costs of downtime per hour. The outsourcing scenario results in increased amounts of downtime. However, you can see that unless downtime costs are over $110,000 per hour, it is still more economical to outsource the maintenance of the storage area network than to provide in-house IT support for it.

To build the function that is highlighted in the picture above, you open the Function Wizard by clicking the icon on the control panel.

You select the **SIM_SYSTEM_RESULTS** function from the function list, as shown next.



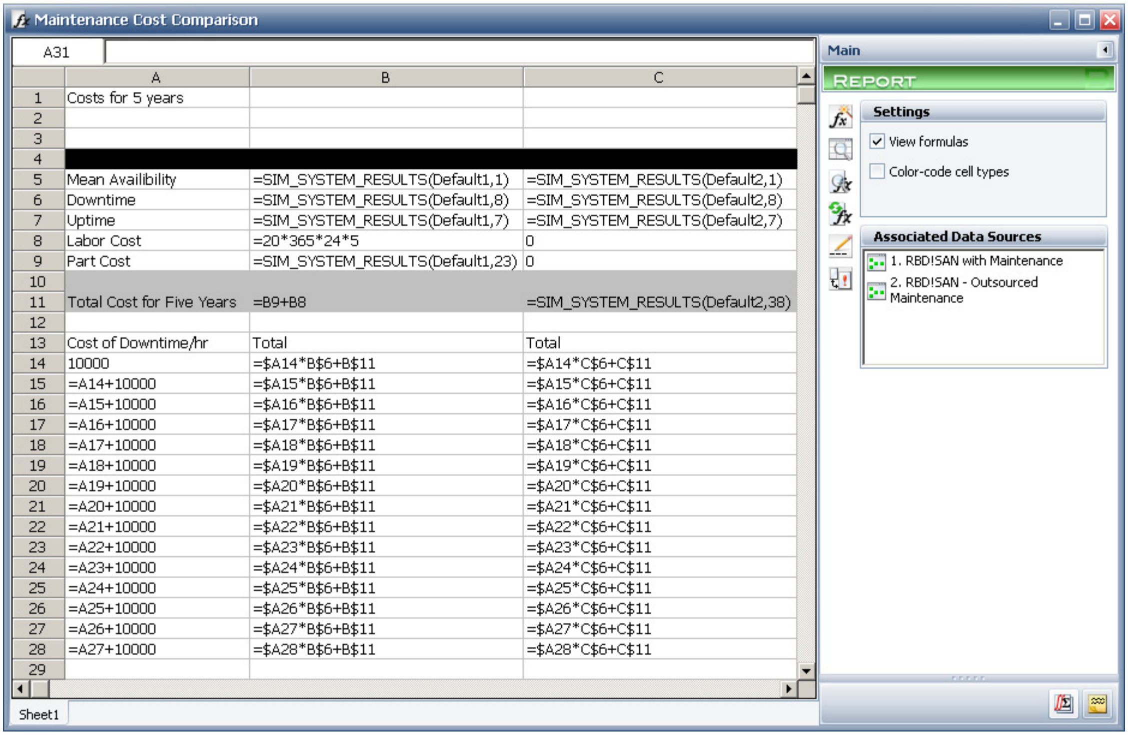
For the Result ID input, you click the **Select** button and select the **General Information > Mean Availability** result, as shown next.



The Phase input is optional (as indicated by the brackets that enclose the label) and not applicable in this example.

You specify the data source for the function by selecting the **Use Default** check box and selecting the number **1** (i.e., the first default data source, which you selected when you created the workbook), then you click **Insert**. The mean availability for the specified diagram will be automatically computed and displayed in the workbook.

You continue to use the Function Wizard, along with some simple formulas entered directly in the cell, to build the rest of the report. All of the formulas used in the report are shown next; this is done by selecting the **View formulas** check box on the control panel.



	A	B	C
1	Costs for 5 years		
2			
3			
4			
5	Mean Availibility	=SIM_SYSTEM_RESULTS(Default1,1)	=SIM_SYSTEM_RESULTS(Default2,1)
6	Downtime	=SIM_SYSTEM_RESULTS(Default1,8)	=SIM_SYSTEM_RESULTS(Default2,8)
7	Uptime	=SIM_SYSTEM_RESULTS(Default1,7)	=SIM_SYSTEM_RESULTS(Default2,7)
8	Labor Cost	=20*365*24*5	0
9	Part Cost	=SIM_SYSTEM_RESULTS(Default1,23)	0
10			
11	Total Cost for Five Years	=B9+B8	=SIM_SYSTEM_RESULTS(Default2,38)
12			
13	Cost of Downtime/hr	Total	Total
14	10000	=$A14*B$6+B$11	=$A14*C$6+C$11
15	=A14+10000	=$A15*B$6+B$11	=$A15*C$6+C$11
16	=A15+10000	=$A16*B$6+B$11	=$A16*C$6+C$11
17	=A16+10000	=$A17*B$6+B$11	=$A17*C$6+C$11
18	=A17+10000	=$A18*B$6+B$11	=$A18*C$6+C$11
19	=A18+10000	=$A19*B$6+B$11	=$A19*C$6+C$11
20	=A19+10000	=$A20*B$6+B$11	=$A20*C$6+C$11
21	=A20+10000	=$A21*B$6+B$11	=$A21*C$6+C$11
22	=A21+10000	=$A22*B$6+B$11	=$A22*C$6+C$11
23	=A22+10000	=$A23*B$6+B$11	=$A23*C$6+C$11
24	=A23+10000	=$A24*B$6+B$11	=$A24*C$6+C$11
25	=A24+10000	=$A25*B$6+B$11	=$A25*C$6+C$11
26	=A25+10000	=$A26*B$6+B$11	=$A26*C$6+C$11
27	=A26+10000	=$A27*B$6+B$11	=$A27*C$6+C$11
28	=A27+10000	=$A28*B$6+B$11	=$A28*C$6+C$11
29			

Note that the inputs for the SIM_SYSTEM_RESULTS are all entered by selecting the desired results in the Function Wizard, as follows:

- Mean Availability: General Information > Mean Availability, as shown above
- Downtime: System Uptime/Downtime > Downtime
- Uptime: System Uptime/Downtime > Uptime
- Part Cost: Costs > CM Pool Cost
- Total Costs for Five Years: Costs > Total Cost

Events and Conditions 7

Often, it is necessary for an event to trigger another event that affects an unrelated block. Maintenance groups allow you to model such configurations. A maintenance group is a set of blocks where some event within the group can trigger either maintenance or state changes for one or more blocks, either within the group or outside of it. You can use a maintenance group:

- To trigger a scheduled task (i.e., a preventive task, inspection or on condition task). You can set the task to be performed based on events such as block failure, maintenance or block restoration in one or more maintenance groups.
- To turn a block on or off. State change triggers are used to activate or deactivate a block when items in one or more specified maintenance groups go down or are restored.

In this chapter

- ✓ Reliability block diagrams (RBDs)
- ✓ Simulation diagrams
- ✓ Scheduled tasks
- ✓ Maintenance groups
- ✓ State change triggers

Watch a video of this example at http://www.reliasoft.tv/blocksim/startguide/blocksim_qsg_7.html.

7.1 Using Events to Trigger Tasks

In addition to normal failures, the storage area network is susceptible to viruses. If a virus attacks, both servers are infected and the servers need to be replaced.

Objectives

- Create an RBD that accurately models this configuration, where the failure and subsequent replacement of the server blocks are triggered by the failure of another block that is unrelated to the server.

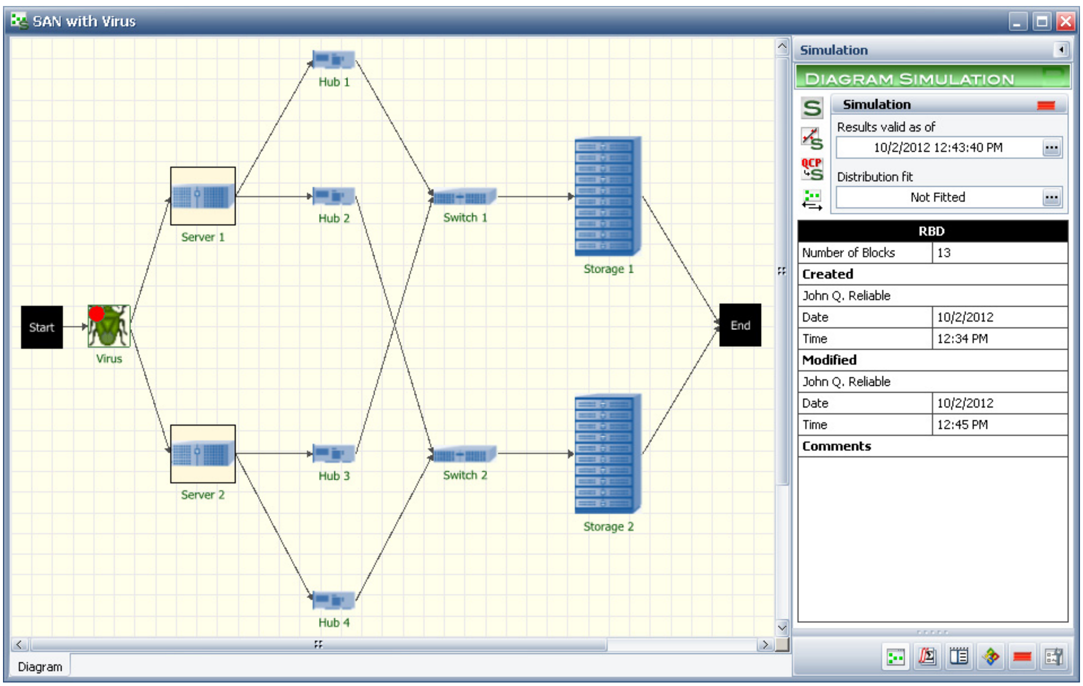
Solution

First, you duplicate the "SAN with Maintenance" diagram, without duplicating the resources, to create a new diagram to work on (*"SAN with Virus" in the sample project*).

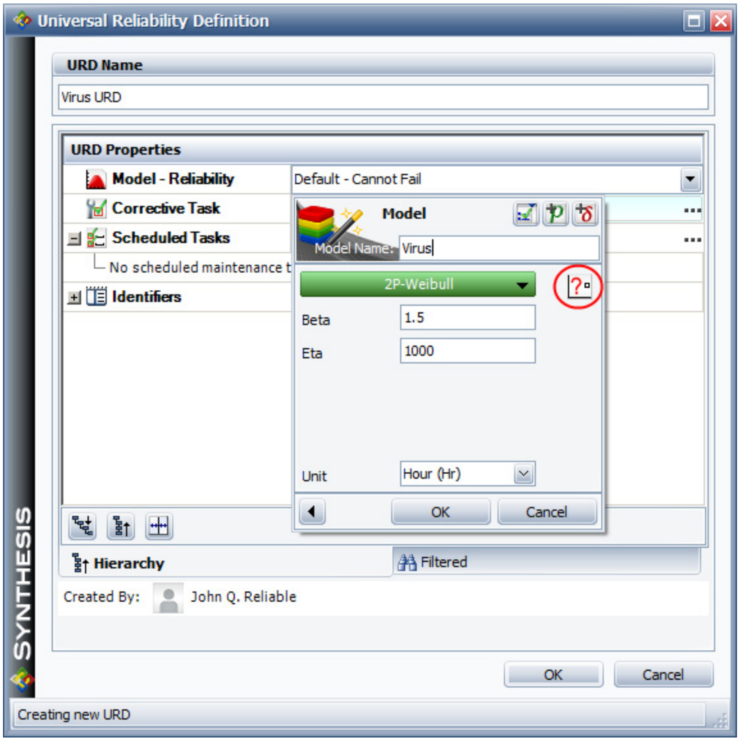
To model this configuration, you will need to add another task to the URD that the Server blocks use. Obviously, you don't want to add this task to the URD that is being used in the existing diagrams, so you need to create a copy of the URD to work with. You open the Resource Manager and, on the URDs page, select the Server URD and choose **Home > Edit > Duplicate**. You rename the copy of the URD (*"Server URD with Scheduled Task" in the sample project*) and then close the Resource Manager.

In the diagram, you open the Block Properties window for the Server 1 block and click in the **URD** field. You click the arrow in the field and then choose the new URD and click **OK** to replace the original URD with the new URD. You then do the same for the Server 2 block.

You then add a standard block named **Virus** to the diagram before the servers and change the connectors accordingly, as shown next.



In the Block Properties window for the Virus block, you create a Virus URD. From the Universal Reliability Definition window, you select to create a new distribution model and, in the Model Wizard, click the **Quick Parameter Estimator** icon, as shown next.

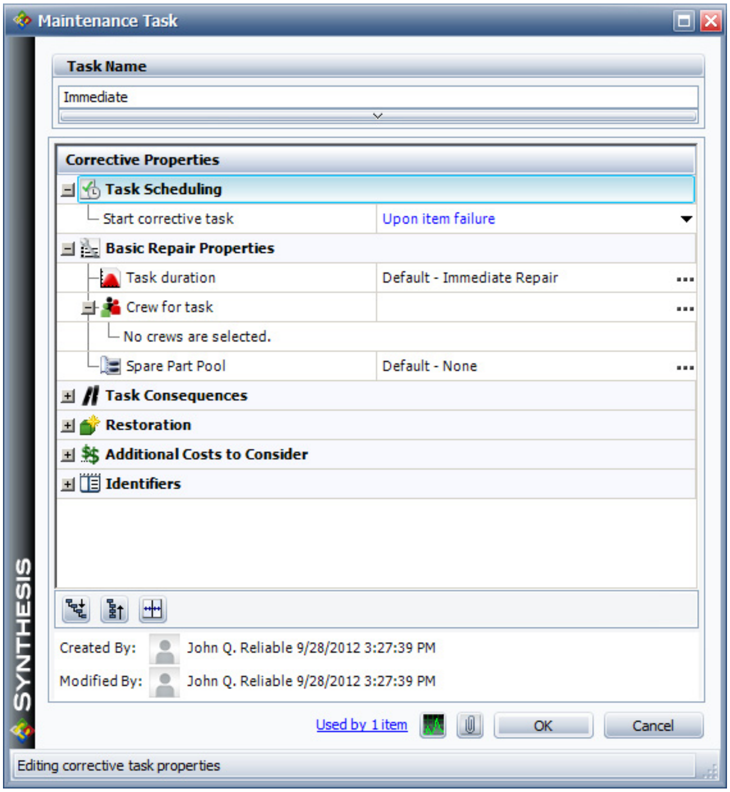


In the Quick Parameter Estimator (QPE), you click the **Use Wizard** button and answer the questions as follows:

- What kind of model are you building? **Event occurrence model**
- Is this event equally likely to happen regardless of the age of the item? **Yes**
- How often would you expect to observe this event on a single item?
  - The worst-case (highest) estimate: **1 in 100 days**
  - The most likely estimate: **1 in 1,000 days**
  - The best-case (lowest) estimate: **1 in 5,000 days**

The QPE calculates a 1-parameter exponential distribution with mean time = 1,516.1666 days. You click **Update** and then **OK** to use the model.

The repair properties are not relevant in this case, but since the Virus block is in series with the rest of the blocks in the diagram, you want to restore it instantly so that the diagram's availability results are not affected by the Virus block. Therefore, you add to the Virus URD a corrective task called Immediate, which starts upon item failure and has a task duration set to the default value (i.e., no model is used), which indicates immediate repair.



You have now set up the Virus block with a probability of occurrence and with immediate repair, which will allow you to use the block to initiate an event without the block affecting the simulation results in terms of availability. To use the block as a trigger, you select the **Belongs to a maintenance group** check box in the Block Properties window and add a maintenance group called Virus in the **Maintenance Group** field.

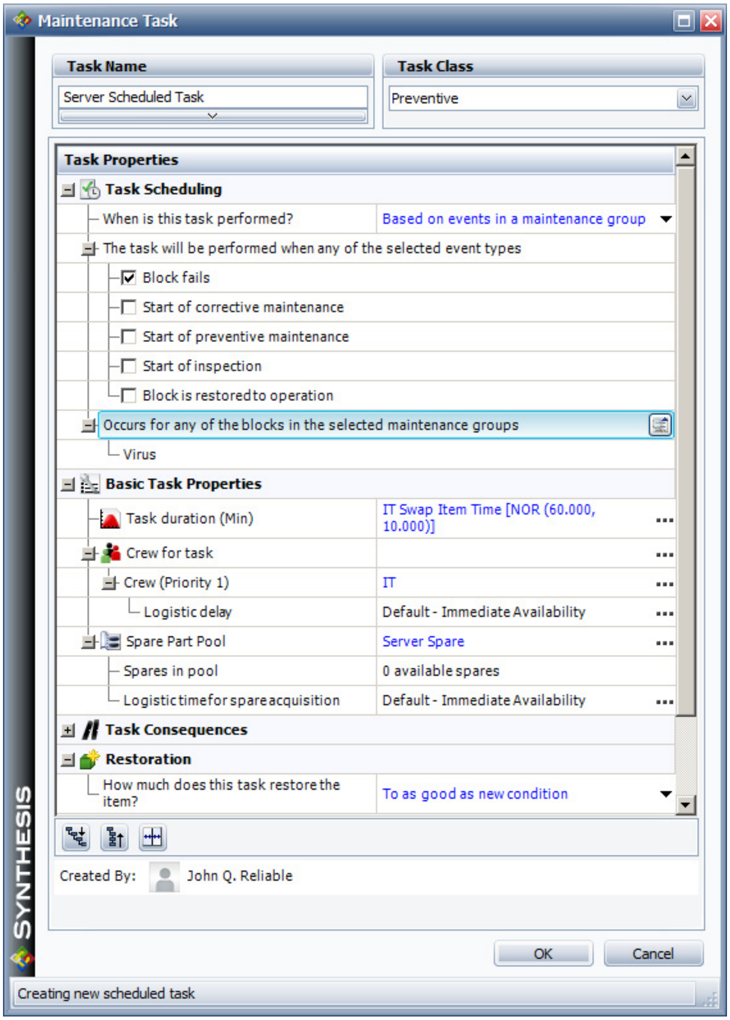
You then choose the Server 1 block from the **Active Block** drop-down list at the bottom of the Block Properties window. You add a scheduled task to the URD, with the following properties:

- Task Name: **Server Scheduled Task**
- Task Class: **Preventive** (Although this is not actually a preventive task, the properties of the preventive task class will accomplish what you need to accomplish here, which is to replace the server with a new one)
- Task Duration: **IT Swap Item Time**
- Crew: **IT**
- Spare Part Pool: **Server Spare**
- Restoration: **To as good as new condition**

In the Task Scheduling properties, you answer **When is this task performed?** by clicking the arrow in the field, clicking **Upon certain events** and then clicking **Based on events in a maintenance group**. When you click **OK**, the Task Scheduling properties allow you to specify the event type(s) that will cause the task to be performed. You select the **Block fails** check box and then, in the **Occurs for any of the blocks in the selected maintenance groups** field, you click the **Select Existing** icon.

In the Select Resource window, you select the Virus maintenance group and click **OK**. The task will be performed when any block in the Virus maintenance group (i.e., the Virus block) fails. In effect, this causes the servers to be shut down and replaced each time the virus event occurs.

The scheduled task is now fully defined, as shown next.



You return to the diagram and click the **Simulate** icon in the control panel.

Prior to simulating, BlockSim checks the diagram for inconsistencies. In this case, you see a warning in the Notifications page of the control panel that lets you know that the Virus block has an associated task (called Immediate) with no duration. You are already aware of this, so you click **Yes** in the message that appears asking if you want to continue to the simulation window.

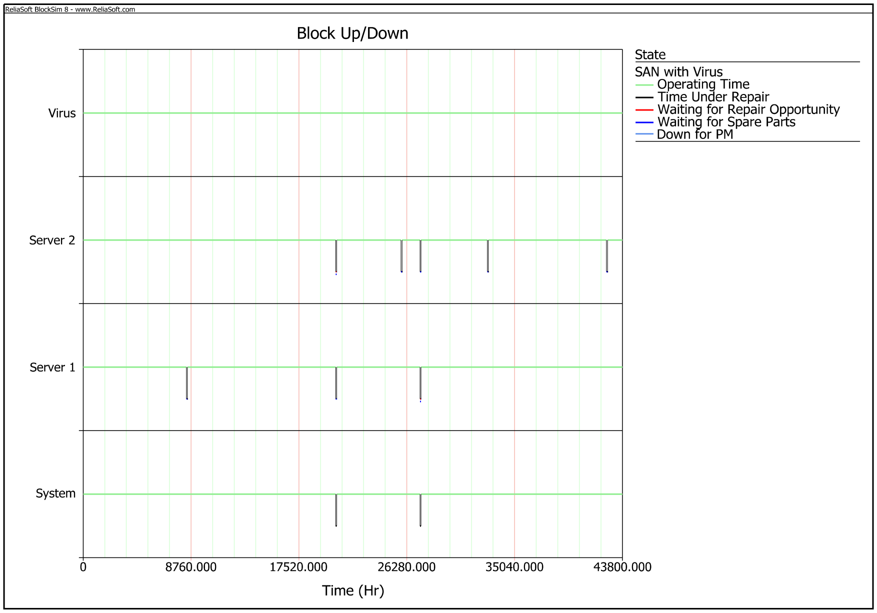
You simulate the diagram using the following settings:

- Seed: **1**
- Number of Threads: **1**
- Simulation End Time: **5 years**
- Number of Simulations: **1**

As you add complexity to the diagram, it becomes important for you to “debug” the diagram, or make sure that it is doing what you thought it should be doing. For debugging purposes, the Block Up/Down plot is your best source for figuring out what the system is doing. Since this plot displays only the last simulation run, there is no reason for you to run more than a single simulation. The plot allows you to trace through the execution of the run and verify that things happened as you planned.

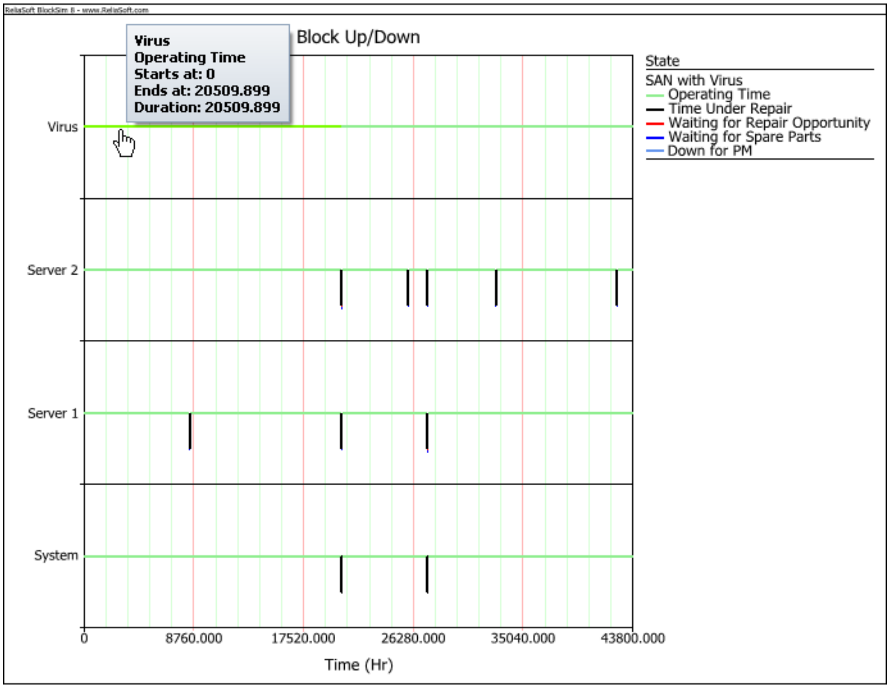
After simulating the diagram, you close the Maintainability/Availability Simulation window, return to the simulation page of the control panel and click the **Plot** icon.

You choose the **Block Up/Down** plot from the **Plot Type** drop-down list. In the **Other** area you select **Show System**, and in the **Available Blocks** area you select only the Server 1, Server 2 and Virus blocks. The Block Up/Down plot is shown next.

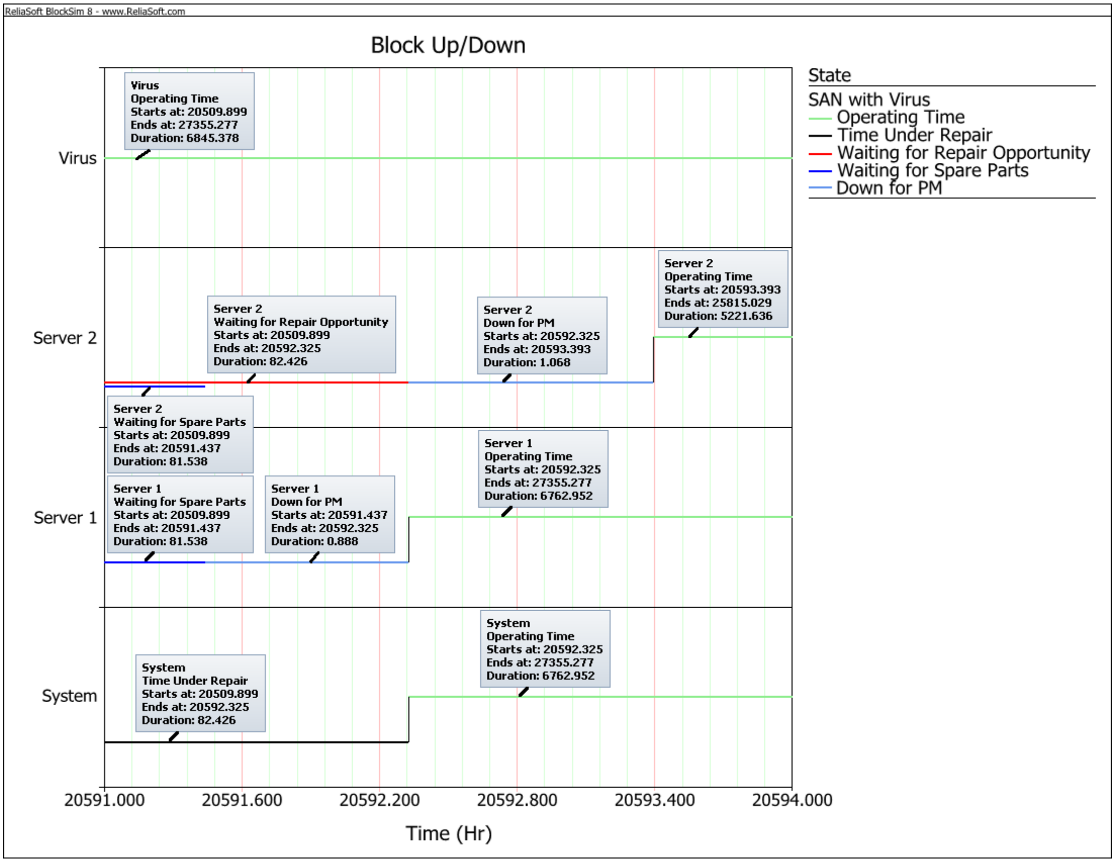


Note: As explained on page 77, your simulation results may vary from the ones shown here.

You see that Server 1 went down fairly early in the simulation, but Server 2 did not go down along with it, so you suspect that the failure was not associated with the Virus block. Indeed, when you point to the left end of the Virus block line, you see that the first operating time for the Virus block ended at 20,509.899 hours; when you point to the first down time for the Server 2 block, you see that it begins at this time.



Pointing to the uptime that follows, you see that Server 2 began operating at 20,593.393 hours, while both Server 1 and the system began operating at 20,592.325 hours. To take a closer look at what happened in this time period, you clear the check box in the **Scaling** area of the control panel and enter a start time of **20,591** hours and an end time of **20,594** hours for the x-axis, then redraw the plot. The plot shown next shows all of the events visible during this time (with annotations added to make the plot easier to interpret).



From this plot, you determine what happened:

- The Virus event occurred at 20,509.899 hours. This event was of zero duration.
- At that time, Server 1 and Server 2 went down and spare parts (new servers) were ordered.
- Because both servers went down, the system went down.
- At 20,591.437 hours, the parts arrived and repair began on Server 1. Server 2 continued to wait for repair, because the single IT technician was busy fixing Server 1.

- Once repairs on Server 1 were complete at 20,592.325 hours, repair on Server 2 began.
- The system began operating at 20,592.325 hours, when Server 1 returned to service.
- Repairs on Server 2 ended at 20,593.393 hours and the server began operating.

7.2 State Change Triggers

You decide that it may be a more efficient use of resources to use only one of the storage devices at a time and keep the other one in a standby mode. Rather than modeling this configuration using a standby container, you decide to use state change triggers (SCT) to represent it.

Note: State change triggers are a powerful new addition to the Synthesis version of BlockSim that provide a great deal of modeling flexibility. With power comes responsibility; it is important to use state change triggers wisely and carefully, as they can lead to cascade events that may produce results other than those you intended.

You will set up the diagram so that Storage 1 is active and Storage 2 is in standby. If Storage 1 fails or Switch 1 fails, then Storage 2 is activated. It will remain active until Switch 2 or Storage 2 fails, and so on. In other words:

Storage 1:

- Starts ON.
- When ON, is deactivated in either of the following cases:
  - Switch 1 fails.
  - Storage 1 itself fails.
- When OFF, is activated in either of the following cases:
  - Switch 2 fails.
  - Storage 2 fails.

Storage 2:

- Starts OFF.
- When OFF, is activated in either of the following cases:
    - Switch 1 fails.
    - Storage 1 fails.
- When ON, is deactivated in either of the following cases:
    - Switch 2 fails.
    - Storage 2 itself fails.

Objectives

- Create an RBD that accurately models this configuration using state change triggers.

Solution

First, you duplicate the "SAN with Virus" diagram, without duplicating the resources, to create a new diagram to work on (*"SAN with State Change Triggers" in the sample project*).

In the new diagram, you open the Block Properties window for the Switch 1 block and select the **Belongs to a maintenance group** check box. You create a maintenance group called Switch 1 in the **Maintenance Group** field. You follow these steps to assign Switch 2, Storage 1 and Storage 2 to new maintenance groups, each named corresponding to the name of the block.

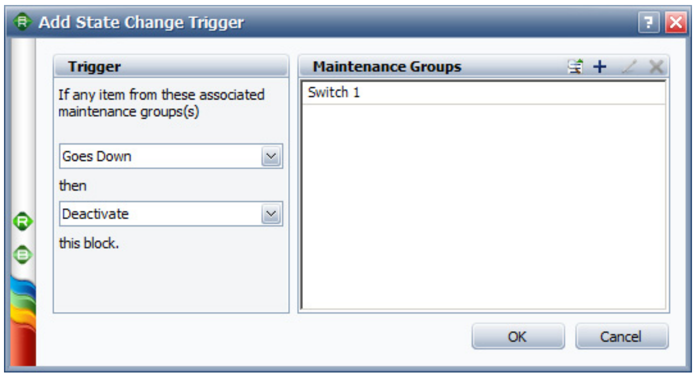
You then set up the state change triggers for the Storage 1 block. In the Block Properties window, you select the **Enable state change triggers (SCT)** check box. You then set the following properties in the fields that appear:

- Initial state: **ON**
- State upon repair: **Default OFF unless SCT overridden**

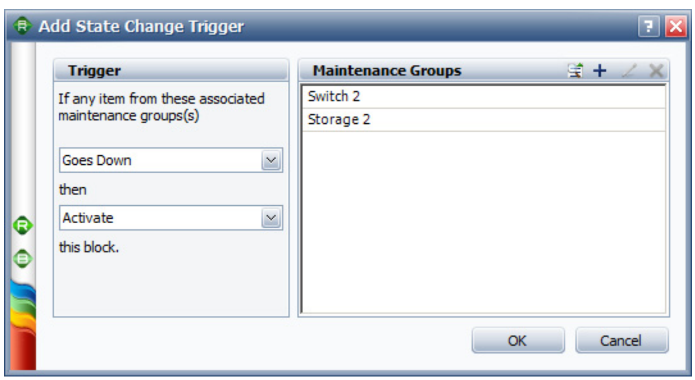
You click the **Add** icon in the **Add a state change trigger** field to open the State Change Trigger window.

On the left side of the window, you use the drop down lists to specify that if any item from the associated maintenance groups **Goes Down**, then BlockSim should **Deactivate** this block. On the right side, you click the **Select Existing** icon and then select the **Switch 1** maintenance group in the Select Resource window.

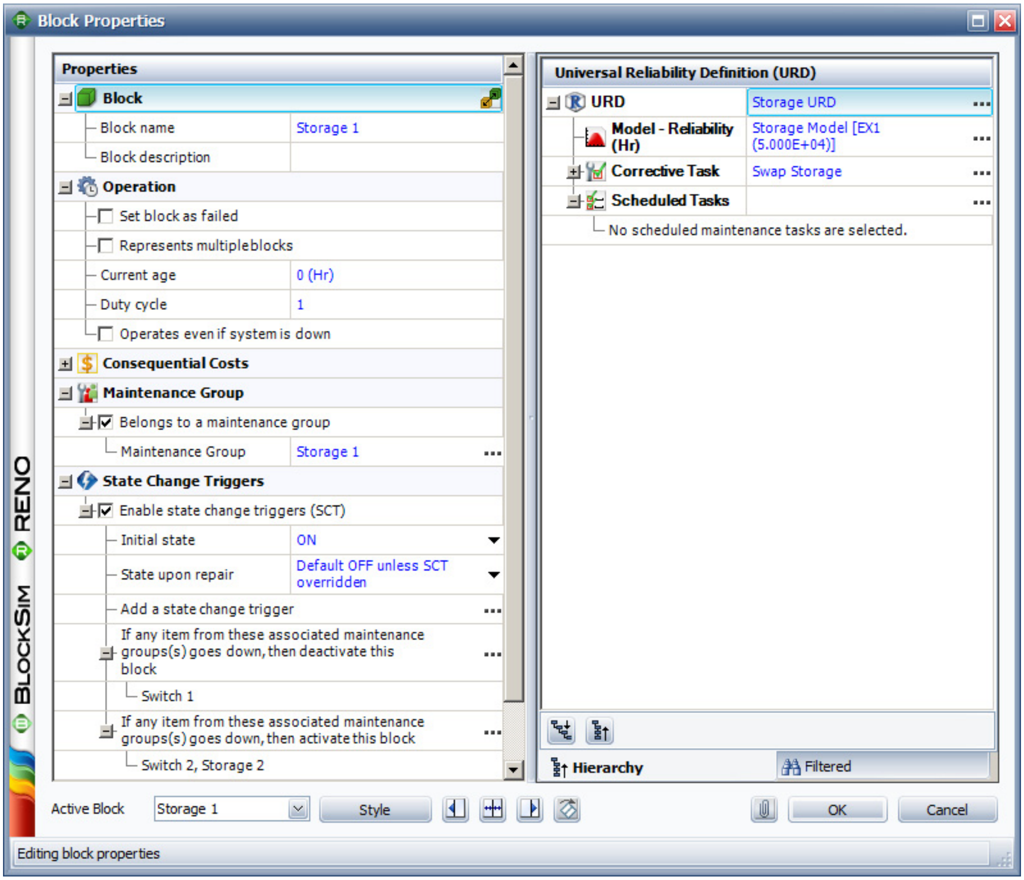
The State Change Trigger window looks like the one shown next.



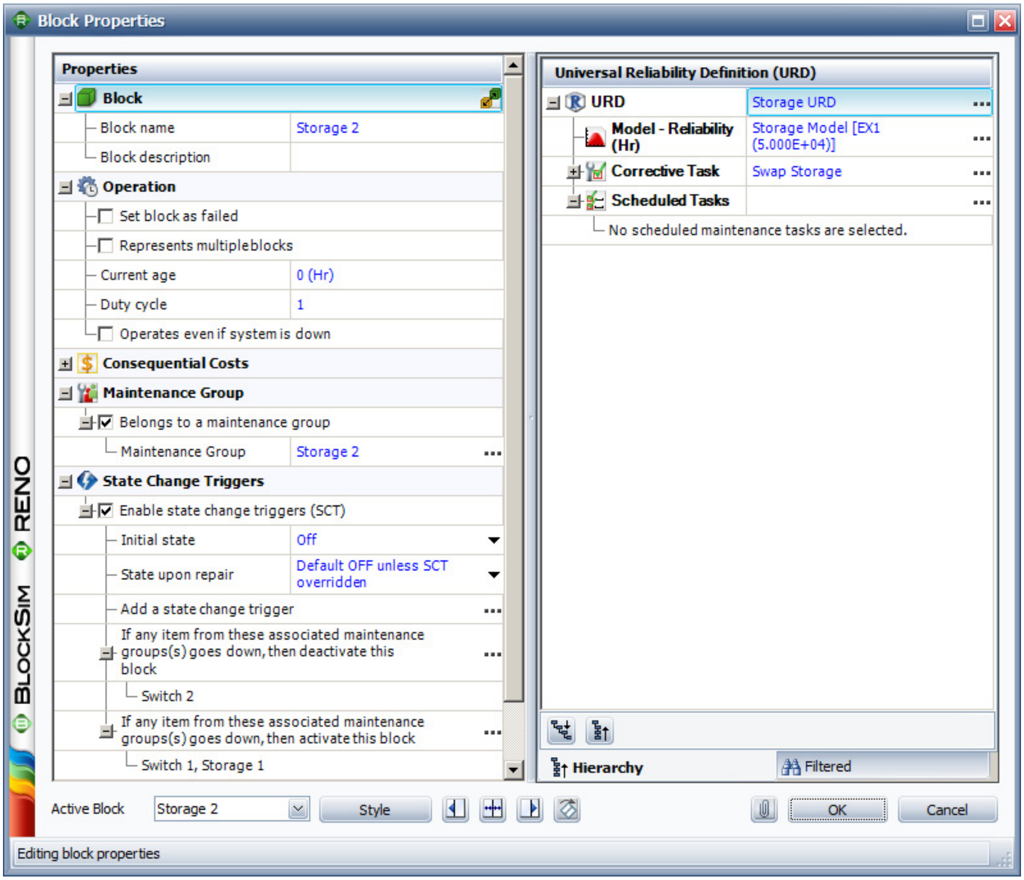
You follow similar steps to add a second state change trigger that specifies that if any item from the Switch 2 or Storage 2 maintenance groups goes down, then this block should be activated.



The complete block properties, including state change triggers, are shown next.

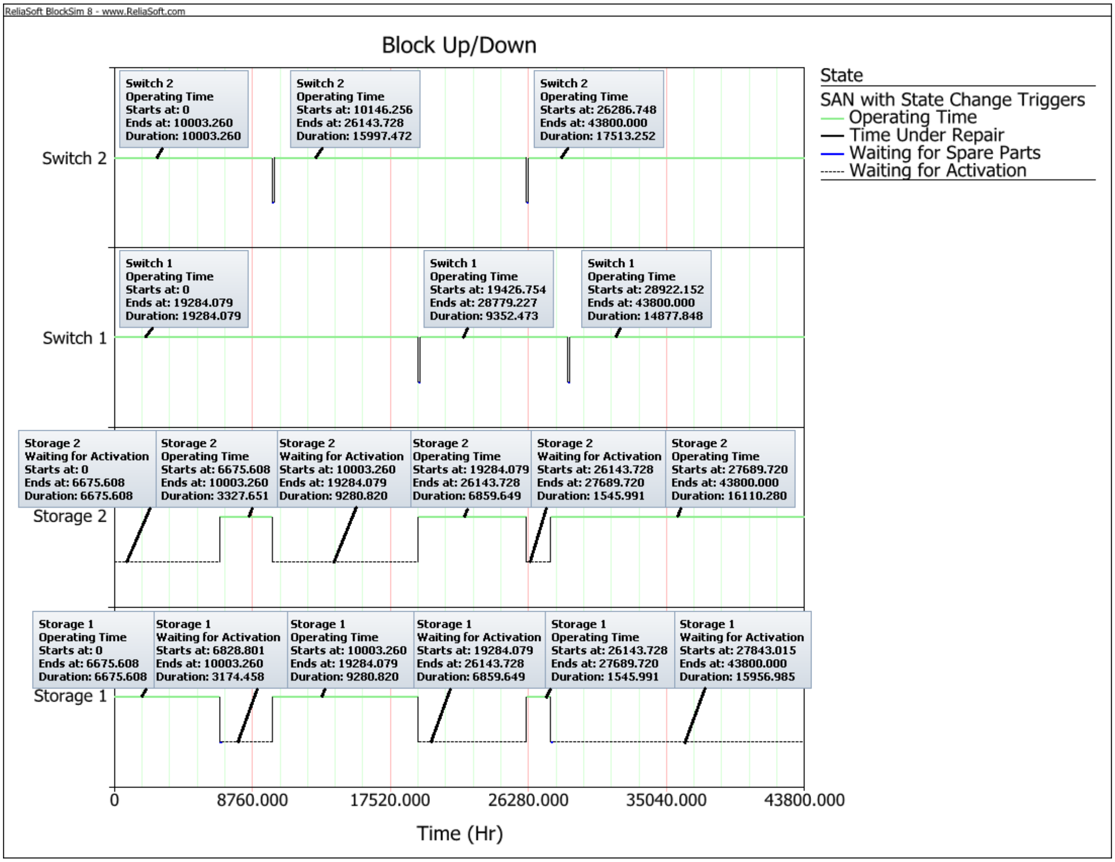


You then follow similar steps for the Storage 2 block. You set its initial state to **OFF** and its state upon repair to **Default OFF unless SCT overridden**. You create a state change trigger to deactivate the block upon item failure in the Switch 2 maintenance group and another state change trigger to activate the block upon item failure in the Switch 1 and Storage 1 maintenance groups, as shown next.



Note: Blocks that have state change triggers are, by default, shown in the diagram with a blue diamond at the lower right corner of the block.

You simulate the diagram, using the same settings given on page 91, except with a seed of **90**. You then create a Block Up/Down plot that shows the Switch 1, Switch 2, Storage 1 and Storage 2 blocks. The plot shown next shows all of the events that occurred (with annotations added to make the plot easier to interpret).



You see that Storage 1 started the simulation in the active state, with Storage 2 in standby. From there:

- At 6,675.608 hours, Storage 1 failed. This activated Storage 2. Storage 1 was repaired.
- At 10,003.260 hours, Switch 2 failed. This deactivated Storage 2 and activated Storage 1.
- At 19,284.079 hours, Switch 1 failed. This deactivated Storage 1 and activated Storage 2.
- At 26,143.728 hours, Switch 2 failed again, deactivating Storage 2 and activating Storage 1.
- At 27,689.720 hours, Storage 1 failed again, activating Storage 2.
- At 28,779.227 hours, Switch 1 failed again. Storage 1 was not currently active, so nothing happened to it. Storage 2 was active and was unaffected.

RELIASOFT CORPORATION END-USER LICENSE AGREEMENT

This ReliaSoft Corporation End-User License Agreement is a legal agreement between You and ReliaSoft Corporation for ReliaSoft Corporation's software products, which includes computer software and associated media and printed materials as well as on-line or electronic documentation and any software updates or upgrades, henceforth collectively called the "Application."

By installing, copying or otherwise using the Application, You agree to be bound by the terms of this End-User License Agreement and represent and warrant that You are authorized to do so. If You do not agree to the terms of this End-User License Agreement, do not install or use the Application and promptly return the Application to the vendor from which you obtained it within fourteen (14) calendar days from the date of purchase for a full refund.

This Agreement is effective from the earlier of the date You open the sealed software package, if applicable, or install the Application, or fourteen (14) calendar days from the date the Application was purchased or otherwise obtained from ReliaSoft Corporation, and continues until expressly terminated by ReliaSoft Corporation.

This Application is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. **This Application is licensed, not sold**.

1 DEFINITIONS

1.1 **Access**. Means connected to or using the Application in any manner, either directly or indirectly through any middle tier application or system.

1.2 **Activate**. Means the act of enabling the Application to operate on a specific computer by registering the Application license with the ReliaSoft License Server and activating the Application for operation.

1.3 **Location**. Means the postal address of the site where the Application shall be installed. Location shall mean a single physical location.

1.4 **Named User(s)**. Means the individual(s) permitted to Access the Application, and it refers to specific individuals.

1.5 **ReliaSoft License Server**. Means the ReliaSoft server(s) that provides license activation, registered end-user management, license management and license compliance enforcement services for the Application.

1.6 **You**. Means the licensee or the individual acquiring the license of this Application and, in either case, the affiliates, employees, contractors and agents of said licensee or individual.

2 LICENSE GRANT

2.1 **Grant**. Subject to the license types defined in Section 3 hereof, You are granted a personal, non-exclusive and limited license to install, Activate and Access the Application under the license type and in the quantities for which You have obtained a license. You may install and Activate the Application onto designated equipment at the specified Location(s), solely as necessary to use the Application in accordance with this Agreement. You may not distribute, sublicense, copy (except as expressly permitted in this Agreement), transfer (except as expressly permitted in this Agreement), nor give the Application to any third party.

2.2 **Ownership**. The Application and the information it contains, including any updates, upgrades, modifications and derivative works, and all copies of them, are the property of ReliaSoft Corporation (or its licensors), and title remains with ReliaSoft Corporation (or ReliaSoft Corporation's licensors). All applicable rights in patents, copyrights and trade secrets in the Application are and will remain with ReliaSoft Corporation (or ReliaSoft Corporation's licensors). No title to, or ownership of, the Application or the information contained therein is transferred to You. ReliaSoft Corporation reserves all rights not explicitly granted in this Agreement.

2.3 **Rights to Developments**. This Agreement also governs your use of any enhancements, data, or information provided by ReliaSoft Corporation while performing any technical, training or consulting services. Any ideas, know-how, techniques and software developed by ReliaSoft Corporation, including any enhancements or modifications made to the Application by any person, are the property of ReliaSoft Corporation.

3 INSTALLATION AND USE

ReliaSoft Corporation licenses any Application under this Agreement on a per user basis. In other words, ReliaSoft Corporation licenses an Application to each user and not to a particular computer. This Agreement grants You the following rights, depending on the license You have obtained from ReliaSoft Corporation:

3.1 **Single User License.** The Single User License is licensed to one individual Named User. This is a perpetual license. Each software package includes the installation media and one copy of the Application documentation per license, either in physical/hard copy format and/or electronic format. The Named User may install, Activate and Access the Application on up to three computers (i.e., multiple Locations), such as a work computer, a home computer and a portable computer, provided that the Named User is the only person who uses this Application, at any time, in any of these Locations. In the event that installation and activation of the Application is performed on three computers, the Named User will be required to deactivate at least one installation before being allowed to Activate the Application again. There is a limit of 6 activations allowed per year. The Single User License cannot be installed on multi-user capable environments such as Windows, Terminal or Citrix servers.

Use of the Application on any of the computers by any user other than the Named User who has licensed the Application is strictly prohibited. Such other users must obtain a separate license of the Application from ReliaSoft before any use.

3.2 **Standard Network License**. The Standard Network License is licensed to Named Users in an organization on a single network of computers at a specified Location. This is a perpetual license. The software package includes the installation media and one copy of the Application documentation per license, either in physical/hard copy format and/or electronic format. ***Note: This is not a Concurrent Network License****.*

This license provides for a limited number of Named Users, dependent on the quantity of seats licensed, to install, Activate and Access the Application, with a single activation allowed for each Named User. For example, if You purchase a ten (10) seat Standard Network License, only ten (10) specific Named Users are allowed Access. If an eleventh (11th) user wishes to have Access to the Application, an additional seat must be licensed. In the event that installation and activation of the Application is required on a new computer for a Named User, the Named User will be required to deactivate the active installation before being allowed to install and Activate the Application on the new computer. The Standard Network License allows for installation on multi-user capable environments such as Windows, Terminal or Citrix servers.

Under this license, no other copies can be installed on non-network computers, such as portable computers, home computers, etc.

3.3 **Concurrent Network License**. The Concurrent Network License is licensed to an organization with multiple regular and occasional Named Users on a single network of computers at a specified Location. This is a perpetual license. The software package includes the installation media and one copy of the Application documentation per license, either in physical/hard copy format and/or electronic format. Concurrent Network Licenses are specified by two attributes:

(i) Number of Named User Activations, which is the maximum number of computers on which a Named User can Activate the Application (i.e., make it available for operation). More than one activation is allowed for each Named User (e.g., to install the Application on both a work computer and a portable computer), but each activation is counted toward the maximum available under the license.

(ii) Number of Concurrent Access Licenses (or CALs), which is the maximum number of computers that can Access the Application at any given time.

For example, for a Concurrent Network License with thirty (30) Named User Activations and ten (10) CALs, the Application can be installed and activated on up to 30 computers (e.g., by 30 Named Users each activating the Application on only one computer, by 15 Named Users each activating the Application on two computers, etc.), and only ten (10) of those computers will be able to Access the Application at the same time. If a user attempts to Access the Application from an eleventh (11th) computer, the user must wait until one of the other ten users ends their session and releases the CAL. The Concurrent Network License allows for installation on multi-user capable environments such as Windows, Terminal or Citrix servers.

Under this license, Named Users are allowed to check out a CAL from the ReliaSoft License Server for a specified period of time, effectively being guaranteed Access to the Application while simultaneously reducing the number of available CALs for the rest of the user pool until the CAL is checked back into the ReliaSoft License Server or otherwise released. This ability to check out a CAL from the ReliaSoft License Server is only available for Applications designated as Major Version 8 and above (e.g., 8.0.1, 8.0.2, etc).

This type of license requires that you provide/allow internet access from the Application to the ReliaSoft License Server at https://Validate.ReliaSoft.org. Concurrency authorization and control is managed and provided by this server. You may need to modify and adjust your internet and/or firewall settings and/or set the aforementioned server as a trusted location in order to use the Application. This requirement for connectivity to the ReliaSoft License Server for concurrency authorization and control only applies for Applications designated as Major Version 8 and above (e.g., 8.0.1, 8.0.2, etc).

3.4 **Asset-based License (only applies to Orion eAPI software)**. The Asset-based License is licensed to all direct employees of a corporate entity within one or more specified Sites, depending on the license obtained. This license excludes subcontractors, suppliers or temporary employees, unless they are using the Application inside an eligible Site on a computer owned by that Site. This is a perpetual license.

Under this license, an unlimited number of Named Users from the licensing Site or Sites is allowed to Access the Application; however, the license restricts the number of physical assets (commonly referred to as Asset TAGs) that can be configured and managed within the Application. For this license, a Site is defined as a distinct physical address/site of a corporate entity, where the physical assets being managed reside (e.g., a refinery or a mine or a power station). The license specifically excludes broader geographical areas, such as cities or states/provinces (where multiple Sites might exist).

3.5 **Unlimited User License.** The Unlimited User License is licensed to all direct employees of a corporate entity within a specified Site, Division, or Corporation, depending on the type of license obtained. This license excludes subcontractors, suppliers or temporary employees, unless they are using the Application inside an eligible facility on a computer owned by that facility. This is a perpetual license.

Site-Wide License: Covers all direct employees working at a distinct physical address/site of a corporate entity. A Site-Wide License specifically excludes broader geographical areas, such as cities or states/provinces (where multiple sites might exist).

Division-Wide License: Covers all direct employees affiliated with a unique business unit of a corporate entity organized in multiple divisions/business units. Examples of divisions are the Aerospace Division of a large conglomerate or the Brakes Division of an automotive corporation. A Division-Wide license might additionally be restricted to a specific geographical area to facilitate licensee requirements (for example, the Aerospace Division of a conglomerate in North America).

Corporate-Wide License: Covers all direct employees of a corporate entity.

This license entitles the specified corporate entity at the specified Site/Division/Corporation to make and distribute copies of the Application in executable code in its entirety, as supplied by ReliaSoft Corporation, OR install and distribute the Application over a network. The software package includes the installation media and one copy of the Application documentation per license, either in physical/hard copy format and/or electronic format. The Unlimited User License allows for installation on multi-user capable environments such as Windows, Terminal or Citrix servers.

3.6 **Rental License.** Rental licenses are available for the Single User, Standard Network and Concurrent Network licenses. The license terms are the same as the corresponding license type being rented except that the license expires according to the term purchased. The rental fee will not be prorated or refunded if license rental is cancelled or not used in full. A standard non-expiring license may be purchased at any time at the current Application list price.

3.7 **Evaluation, Demo, No Save Demo, Beta, or Otherwise Expiring or Non-Expiring Licenses of any Type Provided for Evaluation Purposes.** These are Evaluation/Not for Sale licenses and, notwithstanding any other section of this Agreement, You may not use the Application for commercial purposes, nor sell or otherwise transfer it for value. "Commercial purposes" include the use of the Application in creation of publicly distributed computer software. You are acquiring only the limited right to use a single copy of the Application solely for evaluation purposes and for a limited time. ReliaSoft Corporation can terminate this Agreement at any time. You have no rights under any sections of this Agreement other than those specifically allowed for within this section (Section 3.7). THE APPLICATION IS LICENSED AS IS with no representations or warranties of any kind. Note: As the section title implies, users of any license provided by ReliaSoft Corporation for evaluation purposes (even if it is, for example, a Single User or Network License), shall abide by the rules set forth in this section.

3.8 **Complimentary License.** Complimentary Licenses have the same installation and use rights as their corresponding purchased licenses (e.g., a Complimentary Single User License provides the same installation and use rights as the Single User License). However, You have no rights under any other sections of this Agreement other than those specifically allowed for within this section (Section 3.8). THE APPLICATION IS LICENSED AS IS with no representations or warranties of any kind and license transfers are NOT allowed.

Regardless of the license type You have obtained from ReliaSoft Corporation, You may access shared data storage files or databases relating to the Application across a wide area network (WAN) or a virtual private network (VPN), instead of a local area network (LAN), as long as such access is permitted by your policies and authorized by You via appropriate Application and network configuration settings. Although such access is not prohibited by ReliaSoft Corporation, you acknowledge that WAN and VPN data transfer rates vary widely and could lead to decreased and unpredictable performance (as compared to a LAN) and therefore disclaim any expectation of Application performance should you choose to a) configure any part of the Application, b) use the Application, or c) access shared data storage files in a distributed environment over a WAN or VPN. Such disclaimer does not apply to any Application designated and licensed as a native Web-based Application by ReliaSoft Corporation.

4 SOFTWARE TRANSFERS

4.1 **Single User License.** You may **permanently** transfer all of your rights related to a Single User License under this Agreement, provided that You retain no copies, You transfer all of the Application (including all component parts, media, printed or electronic materials, any upgrades and this Agreement) and the transferee agrees to the terms of this Agreement. If the Application is an upgrade, any transfer must include all prior versions of the Application. Single User License transfers are only allowed between Named Users of the same organization (i.e., transfers are not allowed across different organizations). The Named User transferring the Application must deactivate any and all active installations of the Application before the transferee can Activate the Application. A Single User License may not be transferred more than two times in a year. This section also applies to Single User License Rentals.

4.2 Standard Network, Concurrent Network, Asset-based and Unlimited User Licenses. Standard Network, Concurrent Network, Asset-based and Unlimited User licenses are NOT transferable across organizations, and licensing rights CANNOT be split or transferred between multiple organizations in cases of corporate acquisitions or divestitures, except as allowed in items (a) and (b) of this section (Section 4.2), and with ReliaSoft Corporation's written consent.

In the case of Standard and Concurrent Network Licenses, Named User Activations can be deactivated and replaced by new users on a **permanent** basis. In other words, replacement of users is allowed to accommodate attrition and reassignment of user responsibilities in the ordinary course of doing business. However, replacing users on a daily, weekly, monthly or even quarterly basis and frequently reactivating users that were previously replaced is an indication that the number of Named Users (for Standard Network Licenses) or Named User Activations (for Concurrent Network Licenses) obtained by You is not adequate to serve the pool of users intending to Access the Application and is therefore prohibited. Each Standard Network License seat or Concurrent Network Named User Activation can be reassigned to a different Named User no more than two times in a year.

In cases of corporate acquisitions or divestitures: (a) Standard Network and Concurrent Network licenses remain with the organization that originally obtained the licenses, (e.g., if the organization that holds the license is acquired by another entity, the licensing rights will be transferred to the acquiring entity); (b) Unlimited Site-Wide licenses remain with the original licensing site and therefore pass on to the owner of the specific licensing site as long as the site continues to exist; (c) Unlimited Division-Wide licenses remain with the organization that originally obtained the licenses in cases where the Division is being partially divested, and are terminated when the Division is fully divested. As an example, in a partial divestiture, the portions of the Division divested will not have access to the Division-Wide license whereas the portions of the Division that remain will continue to have access to the Division-Wide license; and (d) Unlimited Corporate-Wide licenses remain with the organization that originally obtained the licenses as long as said organization divested a portion of its business or acquired other organizations, and are terminated in cases where said organization was acquired by another organization. As an example, in a partial divestiture, the portions of the corporation divested will not have access to the Corporate-Wide license whereas the portions of the corporation that remain will continue to have access to the Corporate-Wide license. As a further example, an organization with a Corporate-Wide license having acquired other entities (either in part or in whole) will continue to have the same licensing rights as before the acquisition and can therefore deploy the same licensing rights to the newly acquired entities.

This section also applies to Standard Network and Concurrent Network License Rentals.

5 DESCRIPTION OF OTHER RIGHTS, LIMITATIONS AND MISCELLANEOUS ITEMS

5.1 **Limitations on Reverse Engineering, Decompilation and Disassembly**. You may not reverse engineer, decompile or disassemble the Application. You shall not provide, disclose or otherwise make available the Application, in any form, to any person other than your employees and under your direction and control for purposes specifically related to your permitted use of the Application. You will not: (a) alter, modify or prepare derivative works of the Application; (b) transmit the Application electronically by any means; or (c) cause or permit the translation, reverse engineering, disassembly, or decompilation of the Application to determine any design structure, source code, concepts and methodology behind the Application, whether to incorporate it within any product of your own creation, create a derivative work, create any product that is competitive with the Application or other ReliaSoft Corporation products, or for any other purpose.

5.2 **Copyright**. All title and copyrights to the Application are owned by ReliaSoft Corporation (or its suppliers or licensors). The Application is protected by copyright laws and international treaty provisions. Therefore, You must treat the Application like any other copyrighted material except that You may make one copy of the media solely for backup or archival purposes.

5.3 **Proprietary Notices**. All title, trademarks, copyrights and intellectual property rights in and pertaining to the Application (including but not limited to any copies thereof, software structure and organization, source code, images and new releases) are valuable property of ReliaSoft Corporation and are owned or licensed by ReliaSoft Corporation. You may not intentionally remove, alter or destroy any form of copyright and trademark notices, proprietary markings or confidential legends placed upon or contained within the Application, including but not limited to any such notices contained in physical and/or electronic media or documentation, in the Application interface boxes, or in any of the runtime resources, code or other embodiments originally contained in or dynamically or otherwise created by the Application.

5.4 **Use of ReliaSoft Corporation's Marks**. You may not use the name, logos, trade names or trademarks of ReliaSoft Corporation or any of its affiliates in any manner including, without limitation, in your advertising, promotional literature or any other material, whether in written, electronic, or other form distributed to any third party, except in the form provided by ReliaSoft Corporation and then solely for the purposes of identifying your use of the ReliaSoft Corporation Application.

5.5 **Verification**. You will provide, on ReliaSoft Corporation's reasonable written request, written verification that the Application is being used according to the terms of this Agreement. Upon thirty days prior written notice, if ReliaSoft Corporation has reasonable grounds to believe that this Agreement has been breached, ReliaSoft may audit your use of the Application provided such audit is: (a) limited to records relating solely to the Application necessary to verify compliance with the terms of this Agreement; (b) performed by a reputable independent third party auditor acceptable to You (acting reasonably); (c) the third party auditor shall comply with your standard security policies; (d) the third party auditor shall execute your standard protective non-disclosure agreement; and (d) the cost of any requested audit will be solely borne by ReliaSoft Corporation if no breach is found as a result of the audit or will be solely borne by You if a breach is found. Such audit shall occur no more than once every twelve months and shall not unreasonably interfere with your normal business operations.

5.6 **Modification**. ReliaSoft Corporation reserves the right to modify or enhance the Application without obligation to notify You of such changes or to furnish them to You, unless otherwise agreed upon with a separate agreement (such as an annual maintenance agreement).

5.7 **Copying**. You may not, under any circumstances, copy the Application, in whole or in part, except as expressly provided under the Copyright section above.

5.8 **Separation of Components**. The Application is licensed as a single product. Its component parts may not be separated for use on more than one computer.

5.9 **Rental or Other Exploitation**. You may not publish, sub-license, re-license, assign, sell, distribute, license, transfer, rent, lease or lend the Application to any party, except transfer the Application as expressly provided under the Transfer section above. If you received any revenues from the unlawful distribution of the Application, such revenues will be forfeited to ReliaSoft Corporation.

5.10 **Fees**. You will pay ReliaSoft Corporation all fees or other amounts due under this Agreement, plus any and all applicable taxes, within the payment term due date specified on the respective invoice. In the event that the respective invoice is not paid on time, or at all, ReliaSoft Corporation reserves the right to terminate this Agreement and revoke the corresponding licenses issued to You within the scope of this Agreement.

5.11 **Termination**. Without prejudice to any other rights, ReliaSoft Corporation may terminate this Agreement if You fail to comply with the terms and conditions of this Agreement and such breach is not cured within thirty days of notice of such breach. In such event, You must destroy all copies of the Application and all of its component parts. Additionally, You may be held liable for any damage or loss of profit caused to ReliaSoft Corporation arising from unauthorized use or duplication of this Application.

5.12 **Supplemental Licenses**. If, after the effective date of this Agreement, You subsequently purchase additional licenses of the Application, these supplemental licenses will be included under this Agreement.

5.13 **Press Releases**. As part of this Agreement, You acknowledge that ReliaSoft Corporation may make reference to You as a customer of ReliaSoft Corporation in press releases, advertising and promotional materials, and You consent to any such reference. ReliaSoft Corporation will NOT disclose any further details beyond referring to You as a customer without prior written consent, not to be unreasonably withheld.

5.14 **Relationship**. You and ReliaSoft Corporation are independent contractors and neither is an agent, joint venture partner, partner or employee of the other, and ReliaSoft Corporation will not be obligated by any agreements or representations made by You to any person, nor with respect to any other action by You, nor will ReliaSoft Corporation be obligated for any damages to any person, whether caused by your actions, failure to act, negligence or willful conduct.

5.15 **Upgrades**. If the Application is an upgrade from another product, whether from ReliaSoft Corporation or another supplier, You may use or transfer the Application only in conjunction with that upgraded product, unless You destroy the upgraded product. If the Application is an upgrade of a ReliaSoft Corporation product, You may use the upgraded product only in accordance with this Agreement. If the Application is an upgrade of a component of a package of software programs that You licensed as a single product, the Application may be used and transferred only as part of that single product package and may not be separated for use on more than one computer.

5.16 **U.S. Government Restricted Rights**. The Application was developed at private expense. No portion of the Application was developed with government funds and the Application is a trade secret of ReliaSoft Corporation for all purposes of the Freedom of Information Act. The Application and documentation are provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 (or its successor) or subparagraphs (c)(1) and (2) of the Commercial Computer Software Restricted Rights at 48 CFR 52.227-19 (or its successor), as applicable. Manufacturer is ReliaSoft Corporation, 1450 S. Eastside Loop, Tucson, Arizona 85710, USA.

5.17 **Limited Warranty**. ReliaSoft Corporation warrants that: (a) the Application will perform substantially in accordance with the accompanying written materials, and on machines meeting the published minimum requirements, for a period of sixty (60) days from the date of receipt; and (b) any media on which the Application is furnished will be free from defects for a period of sixty (60) days from the date of receipt. Some states and jurisdictions do not allow limitations on duration of an implied warranty, so the above limitation may not apply to You; in such states and jurisdictions the liability of ReliaSoft Corporation shall be limited to the minimum extent permitted by law. To the extent allowed by applicable law, implied warranties on the Application and media, if any, are limited to sixty (60) days; and (c) at the time of installation, the Application will be free from any mechanism, feature or any other codes or instructions that: (i) cause the Application to remotely transmit information to ReliaSoft or any third party, except to communicate with one of ReliaSoft's servers to facilitate specific functions of the Application (such as to communicate with the ReliaSoft License Server, access the online help files, etc.); or (ii) may be used to permit Access to, or use of, the Application or computer system on which the Application is loaded, or to which the Application is linked, by ReliaSoft or any third party.

5.18 **Customer Remedies**. ReliaSoft Corporation's and its suppliers' or licensors' entire liability and Your exclusive remedy shall be, at ReliaSoft Corporation's option, either (a) return of the fee paid for the Application, or (b) repair or replacement of the Application or media that does not meet ReliaSoft Corporation's Limited Warranty and which is returned to ReliaSoft Corporation with a copy of your receipt or invoice. Any replacement Application or media will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer. None of these remedies nor any product support services offered by ReliaSoft Corporation are available without a valid License Certificate issued by ReliaSoft Corporation.

5.19 **Warranty Exclusions**. The Limited Warranty is void if the damage or defect has resulted from accident, abuse or misapplication. Any modification of the Application by any person other than ReliaSoft Corporation shall void this warranty. Any manipulation of the Application's data storage infrastructure or direct storage of data into the Application's data storage from outside the Application by any person other than ReliaSoft Corporation or ReliaSoft Corporation's authorized representative shall void this warranty. The warranties in this section extend only to You and are contingent upon proper use of the Application. The warranties will not apply to any failure caused by (a) accident, (b) unusual physical, electrical or electro-magnetic stress, (c) negligence, (d) misuse, (e) failure of electrical power, air conditioning or humidity control, (f) use of the Application with any equipment or software not reflected in ReliaSoft Corporation's specifications, (g) installation, alteration or repair of the Application by anyone other than ReliaSoft Corporation or ReliaSoft Corporation's authorized representative, or (h) or installation on equipment on which the original identification marks have been removed or altered.

5.20 **No Other Warranties**. No oral or written information or advice given by ReliaSoft Corporation, its suppliers, dealers, distributors or agents shall create a warranty or in any way increase the scope of the Limited Warranty, and You may not rely on any such information or advice as a warranty.

5.21 **Use of Results Provided By the Application Disclaimer**. You understand that the results provided by the Application cannot replace judgment required for important decisions. Use of the results provided is done completely at your own risk. ReliaSoft Corporation does not warrant that the functions of this Application will meet your requirements or be error free. You assume all risk of the use, quality and performance of the Application, and You are advised to use your own discretion and judgment regarding the use of the Application.

5.22 RELIASOFT CORPORATION, ON BEHALF OF ITSELF AND ITS LICENSORS, DISCLAIMS ALL OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH REGARD TO THE APPLICATION. IN NO EVENT SHALL RELIASOFT CORPORATION OR ITS SUPPLIERS OR LICENSORS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE APPLICATION, EVEN IF RELIASOFT CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. USE OF THIS APPLICATION IS DONE COMPLETELY AT YOUR OWN RISK, INCLUDING THE USE OF RESULTS PROVIDED BY THIS APPLICATION. RELIASOFT CORPORATION'S LIABILITY HEREUNDER SHALL BE LIMITED TO PHYSICAL DAMAGE DIRECTLY CAUSED BY THE SOLE NEGLIGENCE OF RELIASOFT CORPORATION AND SHALL NOT EXCEED THE PRICE

PAID FOR THE SOFTWARE PRODUCT. NOTWITHSTANDING ANYTHING HEREIN TO THE CONTRARY, THE FOREGOING DISCLAIMER AND LIMITATION OF LIABLITY SHALL NOT APPLY TO RELIASOFT CORPORATION'S OBLIGATIONS UNDER ANY CLAIM OF INFRINGEMENT AS DESCRIBED IN SECTION 5.28 OF THIS AGREEMENT.

5.23 **Venue**. Venue for any proceedings arising out of or relating to this Agreement shall be in Tucson, Pima County, Arizona. The parties disclaim the application of the United Nations Convention on Contracts for the International Sale of Goods. This Agreement is governed by the laws of the State of Arizona, USA, without reference to conflict to law principles. Each party to this Agreement submits to the exclusive jurisdiction of the state and federal courts in the State of Arizona for the purpose of resolving any disputes arising under or relating to this Agreement. Each party waives any jurisdictional, venue or inconvenient forum objections to such courts.

5.24 **Legal Expenses**. If legal action is taken by either party to enforce this Agreement, all costs and expenses (including reasonable attorney fees) incurred by the prevailing party in exercising any of its rights or remedies or in enforcing any of the terms, conditions, or provisions of this Agreement will be paid by the other party.

5.25 **Force Majeure**. ReliaSoft Corporation will not be responsible for delays or failures in its performance due, in whole or in part, to any cause beyond its reasonable control.

5.26 **Export Control**. Regardless of any disclosure made by You to ReliaSoft Corporation of an ultimate destination of the Application, You will not export, either directly or indirectly any Application without first obtaining all licenses required, from the U.S. Department of Commerce or any other agency or department of the United States Government, and complying with the applicable laws. Neither the Application nor any direct product thereof may be exported, directly or indirectly, in violation of applicable export laws, or may be used for any purpose prohibited by these laws including, without limitation, nuclear, chemical or biological weapons proliferation. ReliaSoft Corporation will provide You with all reasonable information requested by You in connection to exporting the Application, including providing You with the U.S. Export Controls Classification Number (ECCN) for the Application.

5.27 **Waiver**. The waiver by either party of any breach of this Agreement shall be in writing and shall not constitute a waiver of any other or subsequent breach. No waiver of any of the provisions of this Agreement will be deemed, or will constitute, a waiver of any other provision, whether or not similar, nor will any waiver constitute a continuing waiver. The failure by a party to enforce any provision of this Agreement will not be deemed a waiver of future enforcement of that or any other provision.

5.28 **Indemnification**. You will indemnify and hold ReliaSoft Corporation harmless against any and all claims, damages, losses, costs or other expenses (including reasonable attorney fees) that arise directly or indirectly from your breach of this Agreement. ReliaSoft Corporation shall defend, indemnify and hold harmless, at its own expense, You and your assigns, successors, directors, officers and employees (each an "Indemnified Party") against any and all claims incurred by or made against an Indemnified Party by a third party in connection with a claim, suit or action which is based on an allegation that the Application when used by You as authorized under this Agreement, misappropriates or infringes any third party patent, copyright, trade secret or other intellectual property right (each, a “Claim”) provided that ReliaSoft Corporation shall have received from the Indemnified Party: (i) notice of such Claim as soon as possible after You receive notice of the Claim; given that a failure to provide notice shall only relieve ReliaSoft Corporation of its indemnity obligation to the extent ReliaSoft Corporation was prejudiced by such failure; (ii) the exclusive right to control and direct the investigation, defense or settlement of such claim; and (iii) all reasonable necessary cooperation by You. If your use of any of the Application is, or in ReliaSoft Corporation’s opinion is likely to be, enjoined due to a Claim, ReliaSoft Corporation may, at its sole discretion: (a) modify the Application so that it becomes non-infringing, provided such modifications result in software with substantially similar functionality and performance; (b) procure for You the right to continue using the Application under substantially the same terms and conditions as provided for hereunder; or (c) if (a) and (b) are commercially impracticable, terminate the Agreement and refund to You the license fee paid by You for the Application which is the subject of the Claim as reduced to reflect a three-year straight-line depreciation from the applicable license purchase date. The foregoing indemnification obligation of ReliaSoft Corporation shall not apply: (1) if the Application is modified by any party other than ReliaSoft Corporation and such modification was not authorized in writing by ReliaSoft Corporation, but solely to the extent the alleged infringement is caused by such modification; or (2) to any release of the Application other than the most current release, provided that: (I) the most current release was either made available at no cost to You and (II) You had a commercially reasonable period of time (not to exceed 60 days) after availability of the current release to implement the current release so as to avoid the infringement claim. This section (Section 5.28) sets forth ReliaSoft Corporation’s sole liability and your sole and exclusive remedy with respect to any claim of infringement.

5.29 **Equitable Relief.** You acknowledge and agree that, due to the unique nature of the Application, there can be no adequate remedy at law for any breach of your obligations under this Agreement, that any such breach may allow You or third parties to unfairly compete with ReliaSoft Corporation resulting in irreparable harm and therefore that, upon any such breach or threat thereof, ReliaSoft Corporation shall be entitled to injunctive and other appropriate equitable relief in addition to whatever remedies it may have at law.

5.30 **Entire Agreement; Amendments.** This Agreement is the complete and exclusive statement of the agreement between the parties and supersedes all prior agreements and communications with respect to the subject matter, and there are no oral representations, understandings or agreements that are not fully expressed herein. Any terms appearing on any order or other form used by You which modify or conflict with the terms and conditions set forth herein are expressly rejected. Except for the purpose of negating implied warranties, no course of prior dealings between the parties and no usage of the trade shall be relevant to supplement or explain any term used in this Agreement. No ReliaSoft Corporation employee other than an officer of ReliaSoft Corporation (Vice President and above) shall have any actual or apparent authority to modify the terms of this Agreement in any way. All amendments shall be in writing and signed by the authorized representative of ReliaSoft Corporation.

5.31 **Severability.** If any one or more of the provisions of this Agreement shall for any reason be held to be invalid, illegal or unenforceable in any respect, any such provision shall be severable from this Agreement, in which event this Agreement shall be construed as if such provision had never been contained herein.

5.32 **Electronic Signatures.** ReliaSoft Corporation and You agree that this Agreement may be executed electronically and that electronic copies of this Agreement shall be binding upon the parties to the same extent as manually-executed copies.