

# **Systems Analysis Programs for Hands-On Integrated Reliability Evaluations (SAPHIRE) Version 8**

## **Volume 4 Tutorial**

C. L. Smith  
S. T. Beck  
S. T. Wood

March 2011



The INL is a U.S. Department of Energy National Laboratory  
operated by Battelle Energy Alliance

**INL/EXT-09-17012  
NUREG/CR-7039**

# **Systems Analysis Programs for Hands-On Integrated Reliability Evaluations (SAPHIRE) Version 8**

## **Volume 4 Tutorial**

**C. L. Smith  
S. T. Beck  
S. T. Wood**

**March 2011**

**Idaho National Laboratory  
Idaho Falls, Idaho 83415**

**<http://www.inl.gov>**

**Prepared for the  
Division of Risk Analysis  
Office of Nuclear Regulatory Research  
U.S. Nuclear Regulatory Commission  
Washington, D.C. 20555  
Job Code N6423**

## AVAILABILITY NOTICE

### Availability of Reference Materials Cited in NRC Publications

Most documents cited in NRC publications will be available from one of the following sources:

1. The NRC Public Document Room, Rockville Pike, Rockville, MD 20852 ([pdr@nrc.gov](mailto:pdr@nrc.gov))
2. The Superintendent of Documents, U. S. Government Printing Office (GPO), Mail Stop SSOP, Washington, DC 20402-9328
3. The National Technical Information Service, Springfield, VA 22161

Although the listing that follows represents the majority of documents cited in NRC publications, it is not intended to be exhaustive.

Referenced documents available for inspection and copying for a fee from the NRC Public Document Room include NRC correspondence and internal NRC memoranda; NRC bulletins, circulars, information notices, inspection and investigative notices; licensee event reports; vendor reports and correspondence; Commission papers; and applicant and licensee documents and correspondence.

The following documents in the NUREG series are available for purchase from the GPO Sales Program: formal NRC staff and contractor reports, NRC-sponsored conference proceedings, international agreement reports, grant publications, and NRC booklets and brochures. Also available are regulatory guides, NRC regulations in the *Code of Federal Regulations*, and *Nuclear Regulatory Commission Issuances*.

Documents available from the National Technical Information Service include NUREG-series reports and technical reports prepared by other Federal agencies and reports prepared by the Atomic Energy Commission, forerunner agency to the Nuclear Regulatory Commission.

Documents available from public and special technical libraries include all open literature items, such as books, journal articles, and transactions. *Federal Register* notices, Federal and State legislation, and congressional reports can usually be obtained from these libraries.

Documents such as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings are available for purchase from the organization sponsoring the publication cited.

Single copies of NRC draft reports are available free, to the extent of supply, upon written request to the Office of Administration, Distribution and Mail Services Section U. S. Nuclear Regulatory Commission, Washington, DC 20555-0001.

The public maintains copies of industry codes and standards used in a substantive manner in the NRC regulatory process at the NRC Library, Two White Flint North, 11545 Rockville Pike, Rockville, MD, 20852, for use. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from the American National Standards Institute, 1430 Broadway, New York, NY 10018.

### **DISCLAIMER NOTICE**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability of responsibility for any third party's use, or the results of such use, or any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.



## PREVIOUS REPORTS

S. T. Wood, C. L. Smith, K. J. Kvarfordt, S. T. Beck, *Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Vol. 1 Summary Manual*, NUREG/CR-6952, August 2008.

C. L. Smith, S. T. Wood, W. J. Galyean, J. A. Schroeder, S. T. Beck, M. B. Sattison, *Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Vol. 2 Technical Reference*, NUREG/CR-6952, August 2008.

K. J. Kvarfordt, S. T. Wood, C. L. Smith, *Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Vol. 3 Code Reference Manual*, NUREG/CR-6952, August 2008.

S. T. Beck, S. T. Wood, C. L. Smith, *Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Vol. 4 Tutorial*, NUREG/CR-6952, August 2008.

C. L. Smith, J. Schroeder, S. T. Beck, *Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Vol. 5 GEM Manual*, NUREG/CR-6952, August 2008.

C. L. Smith, R. Nims, K. J. Kvarfordt, C. Wharton, *Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Vol. 6 Quality Assurance Manual*, NUREG/CR-6952, August 2008.

K. J. Kvarfordt, S. T. Wood, C. L. Smith, *Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Vol. 7 Data Loading Manual*, NUREG/CR-6952, August 2008.

Smith, C. L., et al., *Testing, Verifying, and Validating SAPHIRE Versions 6.0 and 7.0*, NUREG/CR-6688, October 2000.

K. D. Russell, et al. *Systems Analysis Programs for Hands-on Reliability Evaluations (SAPHIRE) Version 6.0 - System Overview Manual*, NUREG/CR-6532, May 1999.

K. D. Russell et al., *Integrated Reliability and Risk Analysis System (IRRAS) Version 5.0, Volume 2 - Reference Manual*, NUREG/CR-6116, EGG-2716, July 1994.

K. D. Russell et al., *Verification and Validation (V&V), Volume 9 – Reference Manual*, NUREG/CR-6116, EGG-2716, July 1994.

K. D. Russell et al., *Integrated Reliability and Risk Analysis System (IRRAS) Version 4.0, Volume 1 - Reference Manual*, NUREG/CR-5813, EGG-2664, January 1992.

K. D. Russell et al., *Integrated Reliability and Risk Analysis System (IRRAS) Version 2.5 Reference Manual*, NUREG/CR-5300, EGG-2613, March 1991.

K. D. Russell, M. B. Sattison, D. M. Rasmuson, *Integrated Reliability and Risk Analysis System (IRRAS) - Version 2.0 User's Guide*, NUREG/CR-5111, EGG-2535, manuscript completed March 1989, published June 1990.

K. D. Russell, D. M. Snider, M. B. Sattison, H. D. Stewart, S.D. Matthews, K. L. Wagner, *Integrated Reliability and Risk Analysis System (IRRAS) User's Guide - Version 1.0 (DRAFT)*, NUREG/CR-4844, EGG-2495, June 1987.

## **ABSTRACT**

The Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) refers to a set of computer programs that were developed to create and analyze probabilistic risk assessment (PRAs). This volume is the tutorial manual for the SAPHIRE software Version 8. In this document, a series of lessons are provided that guide the user through basic steps common to most analyses performed with SAPHIRE. The tutorial is divided into two major sections covering both basic and advanced features. The section covering basic topics contains lessons that lead the reader through development of a hypothetical problem involving a potential shipping accident, highlighting the software's fundamental features. Then, the advanced features section contains additional lessons that expand on fundamental analysis features of SAPHIRE 8 and provide insights into more complex analysis techniques. Together, these two elements provide an overview into the operation and capabilities of the SAPHIRE software.





## FOREWORD

The U.S. Nuclear Regulatory Commission (NRC) has developed the Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) software that is used to perform probabilistic risk assessments (PRAs) on a personal computer. SAPHIRE enables users to supply basic event data, create and solve fault and event trees, perform uncertainty analyses, and generate reports. In that way, analysts can perform PRAs for any complex system, facility, or process.

For nuclear power plant PRAs, SAPHIRE can be used to model a plant's response to initiating events, quantify core damage frequencies, and identify important contributors to core damage (Level 1 PRA). The program also can be used to evaluate containment failure and release models for severe accident conditions given that core damage has occurred (Level 2 PRA). In so doing, the analyst could build the PRA model assuming that the reactor is initially at full power, low power, or shutdown. In addition, SAPHIRE can be used to analyze both internal and external events and, in a limited manner, to quantify the frequency of release consequences (Level 3 PRA). Because this software is a very detailed technical tool, users should be familiar with PRA concepts and methods used to perform such analyses.

SAPHIRE has evolved with advances in computer technology and users' needs. Starting with Version 5, SAPHIRE operated in the Microsoft Windows™ environment. Versions 6 and 7 included features and capabilities for developing and using larger, more complex models. SAPHIRE Version 8 includes significant new features and capabilities to meet user needs for NRC risk-informed programs. In general, these include:

Improved user interfaces supporting NRC's Significance Determination Process, event and condition assessments, and more detailed types of PRA analyses.

Development and use of NRC's Standardized Plant Analysis Risk models.

New and improved solving algorithms.

Support features for user-friendliness.

This NUREG-series report comprises seven volumes as outlined below and incorporates new features and capabilities of Version 8.

### Volume 1, "Overview and Summary"

Volume 1 provides an overview of the functions and features available in SAPHIRE Version 8 and presents general instructions for using the software.

### Volume 2, "Technical Reference"

Volume 2 summarizes the fundamental mathematical concepts of sets and logic, fault trees, and probability. It then describes the algorithms used to construct a fault tree and to obtain the minimal cut sets. This report presents the formulas used to obtain the probability of the top event from the minimal cut sets and the formulas for probabilities that apply for various assumptions concerning reparability and mission time. In addition, it defines the measures of basic event importance that SAPHIRE can calculate. This volume also gives an overview of uncertainty analysis using simple Monte Carlo sampling or Latin Hypercube sampling and states

the algorithms used by this program to generate random basic event probabilities from various distributions. Finally, this report discusses enhanced and new capabilities such as post-processing rules, integrated model solving using model types, and workspace analysis routines.

### Volume 3, “Users’ Guide”

Volume 3 provides a brief discussion of the purpose and history of the software as well as general information such as installation instructions, starting and stopping the program, and some pointers on how to get around inside the program. Next, it discusses database concepts and structure. The following nine sections (one for each of the menu options on the SAPHIRE main menu) furnish the purpose and general capabilities for each option. Finally, Volume 3 provides the capabilities and limitations of the software.

### Volume 4, “Tutorial”

Volume 4 provides a series of lessons that guide the user through basic steps common to most analyses performed with SAPHIRE.

### Volume 5, “Workspaces”

Volume 5 describes the functionality and process behind SAPHIRE Version 8 workspaces. Workspaces provide an area in which a PRA model can be analyzed to obtain risk insights for a given initiating event or condition. Workspaces replace the “Graphical Evaluation Module” in earlier SAPHIRE versions.

### Volume 6, “Quality Assurance”

Volume 6 is designed to describe how the SAPHIRE software quality assurance (QA) is performed for Version 8, what constitutes its parts, and the limitations of those processes. In addition, this report describes the Independent Verification and Validation that was conducted for Version 8 as part of an overall QA process.

### Volume 7, “Data Loading”

Volume 7 is designed to guide the user through the basic procedures necessary to enter PRA data into the SAPHIRE program using SAPHIRE’s MAR-D ASCII-text (or “flat file”) data formats. In addition, this manual covers loading data through the new Accident Sequence Matrix and discusses the Project Integrate interfaces with SAPHIRE.

---

Christiana H. Lui, Director  
Division of Risk Analysis  
Office of Nuclear Regulatory Research  
U.S. Nuclear Regulatory Commission

# CONTENTS

<u>Section</u>	<u>Page</u>
PREVIOUS REPORTS .....	ii
ABSTRACT .....	iii
FOREWORD.....	v
LIST OF FIGURES .....	viii
LIST OF TABLES.....	x
EXECUTIVE SUMMARY .....	xi
ACKNOWLEDGEMENTS .....	xiii
ACRONYMS .....	xv
1. INTRODUCTION.....	1
2. BASIC LESSONS .....	3
2.1 Lesson 1 – Adding and Selecting a Project .....	3
2.2 Lesson 2 – Building Event Trees .....	7
2.3 Lesson 3 – Building Fault Trees.....	15
2.3.1 Fault Tree Editor Symbols.....	16
2.3.2 Creating a Fault Tree in SAPHIRE .....	16
2.4 Lesson 4 – Basic Event Information.....	23
2.4.1 Entering Basic Event Information .....	24
2.5 Lesson 5 – Graphical Output .....	28
2.5.1 Printing Event Tree Graphics .....	28
2.5.2 Printing Fault Tree Graphics .....	30
2.6 Lesson 6 – Generating Fault Tree Cut sets .....	31
2.6.1 STEP 1: Generating Fault Tree Cut Sets .....	32
2.6.2 STEP 2: Displaying Fault Tree Cut Sets .....	33
2.6.3 STEP 3: Reporting Fault Tree Cut Sets Results .....	35
2.7 Lesson 7 – Fault Tree Uncertainty Analysis.....	36
2.8 Lesson 8 – Generating Event Tree Cut Sets .....	39
2.8.1 STEP 1: Linking Event Tree Sequences .....	40
2.8.2 STEP 2: Solve For Sequence Cut Sets.....	40
2.8.3 STEP 3: Performing Sequence Uncertainty Analysis.....	43
2.8.4 STEP 4: Reporting Sequence Cut Set Results .....	45
3. ADVANCED LESSONS .....	49
3.1 Lesson 10 – Creating Flag Sets.....	49
3.1.1 Creating a Flag Set .....	49
3.2 Lesson 11 – The “Linking” Rule Editor.....	54

3.2.1	STEP 1: Creating the New Substitution Fault Tree .....	56
3.2.2	STEP 2: Creating the Linking Rule.....	57
3.2.3	STEP 3: Re-Linking the Event Tree Sequence .....	59
3.2.4	STEP 4: Re-solve For Sequence Cut Sets .....	59
3.3	Lesson 12 – Post-processing Rules.....	60
3.3.1	STEP 1: Creating a Post-processing Rule .....	62
3.3.2	STEP 2: Re-solve For Fault Tree Cut Sets .....	64
3.4	Completion of the Tutorial .....	65

## LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
Figure 1. SAPHIRE 8 main screen (General Analysis area) upon startup.....	4
Figure 2. New project dialog .....	5
Figure 3. SAPHIRE open project dialog.....	6
Figure 4. The OBSTRUCT event tree .....	7
Figure 5. Event tree list.....	8
Figure 6. Event tree modify properties dialog .....	9
Figure 7. The SAPHIRE 8 graphical event tree editor with a new tree .....	9
Figure 8. Editing a top event dialog .....	10
Figure 9. Example of adding a branch point for an event tree sequence .....	11
Figure 10. Warning message when attempting to delete an event tree node.....	12
Figure 11. The OBSTRUCT event tree in the graphical editor.....	12
Figure 12. Editing an end state dialog .....	13
Figure 13. BRAKE-SYS fault tree .....	17
Figure 14. OP-MANEUVER fault tree .....	17
Figure 15. LOAD fault tree .....	18
Figure 16. Fault tree properties dialog .....	19
Figure 17. Graphical fault tree editor for a new fault tree.....	19

Figure 18. Gate editor dialog in the fault tree.....	20
Figure 19. Fault tree development steps and results.....	21
Figure 20. Gate modify dialog.....	22
Figure 21. Basic event listing screen.....	24
Figure 22. Modify basic event editing dialog.....	26
Figure 23. Event tree publish reports screen.....	29
Figure 24. Event tree graphics publish reports screen.....	30
Figure 25. Fault tree publish reports screen.....	31
Figure 26. Solve fault tree dialog.....	32
Figure 27. Cut set solve process dialog.....	33
Figure 28. Viewing fault tree cut sets screen.....	34
Figure 29. Fault tree cut sets report using the publish option.....	36
Figure 30. Fault tree uncertainty analysis dialog.....	38
Figure 31. Fault tree uncertainty analysis results.....	39
Figure 32. Event tree cut set solve dialog.....	41
Figure 33. Event tree cut set summary for solving dialog.....	42
Figure 34. Event tree cut set dialog.....	43
Figure 35. Event tree sequence list.....	44
Figure 36. Event tree sequence uncertainty analysis screen.....	44
Figure 37. Event tree sequence uncertainty analysis results.....	45
Figure 38. Event tree cut sets report using the publish option.....	47
Figure 39. Flag set option.....	50
Figure 40. Flag set editor.....	51
Figure 41. Adding a new event to the FLAG1 flag set.....	52
Figure 42. Modifying an event in the FLAG1 flag set.....	52
Figure 43. Modifying the flag set assigned to a fault tree.....	53

Figure 44. Fault tree cut sets with the FLAG1 flag set assigned to it.....	54
Figure 45. LOAD1 substitution fault tree.....	57
Figure 46. Event tree linkage rule editor (without a rule) .....	58
Figure 47. Event tree link creator.....	59
Figure 48. Event tree cut sets with a sequence flag set applied.....	60
Figure 49. Fault tree post-processing rules editor .....	63
Figure 50. Fault tree cut set solve dialog .....	64
Figure 51. Fault tree cut set viewer showing the post-processed cut sets.....	65

## LIST OF TABLES

<u>Table</u>	<u>Page</u>
Table 1. Basic event data values for the tutorial. ....	25
Table 2. SAPHIRE linking rule search criteria examples.....	56
Table 3. SAPHIRE post-processing rule search criteria examples.....	61

## EXECUTIVE SUMMARY

In this tutorial, lessons are provided that guides the user through the basic procedures necessary to perform analyses with the Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) software for Version 8. Additional specific instructions for individual commands and options available in SAPHIRE are provided in the SAPHIRE Code Reference Manual (Volume 3) that accompanies this report.

The Basic Lessons section contains an example of an accident scenario and allows the user to become familiar with the basic menu structure and features of SAPHIRE. The Basic Lessons section covers the following topics:

- Adding and Selecting a Project
- Building Event Trees
- Building Fault Trees
- Entering Basic Event Information
- Graphical Output
- Generating, Displaying, and Reporting Fault Tree Cut Sets
- Fault Tree Uncertainty Analysis
- Generating Event Tree Cut Sets

The Advanced Features section goes into detail of the SAPHIRE advanced features using the example database. Each lesson stands alone, but a lesson will generally build on the basic example. The Advanced Lessons section covers the following topics:

- Creating Change Sets and Flag Sets
- The “Linking” Rule Editor
- Post-processing Rules

For each topical area, systematic instructions are provided to guide the user through that procedure. Applicable screen shots accompany the textual instructions.

It is assumed that the SAPHIRE program has been successfully loaded onto a personal computer, that the user is familiar with the operations of Microsoft Windows™, a standard mouse, and that the user has a basic knowledge of event trees and fault trees. The user will have found reading Section 2.0 of the SAPHIRE Code Reference Manual (Volume 3) provided an overview of the SAPHIRE software, menu structure, and database concepts for the general knowledge of the program.





## **ACKNOWLEDGEMENTS**

We would like to specifically acknowledge Mr. Dan O'Neal of the U.S. Nuclear Regulatory Commission for his contribution to the development this report.



## ACRONYMS

NRC	Nuclear Regulatory Commission
PRA	probabilistic risk assessment
SAPHIRE	Systems Analysis Programs for Hand-on Integrated Reliability Evaluations



# Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Version 8

## Volume 4 Tutorial

### 1. INTRODUCTION

In this tutorial, a series of lessons is provided that guides the user through the basic procedures necessary to perform analyses with the Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) software package for Version 8. Specific instructions for individual commands and options available in SAPHIRE are provided in the SAPHIRE Reference Manual that accompanies this report.

This tutorial is separated into two major sections: (1) Basic Lessons and (2) Advanced Features. The Basic Lessons section contains a simple example of a trucking accident and allows the user to become familiar with the basic menu structure and features of SAPHIRE. The Advanced Features section goes into detail of the SAPHIRE advanced features using the example database. The Basic Lessons section contains Lesson 1 through Lesson 8, while the Advanced Features section covers Lesson 10 through Lesson 12. Each lesson stands alone, but a lesson will generally build on the basic example.

#### Basic Lessons

The Basic Lessons (Section 2) section covers the following topics:

##### **Lesson 1. Adding and Selecting a Project**

This lesson describes the process of adding and selecting a project called "PRACTICE." Adding and selecting a project is the first step in developing the example analysis.

##### **Lesson 2. Building Event Trees**

This lesson describes the specific process of creating the event tree called "OBSTRUCT" that will be used in the example analysis.

##### **Lesson 3. Building Fault Trees**

This lesson describes the specific process of creating a basic fault tree. In this lesson, five simple fault trees will be constructed and saved.

##### **Lesson 4. Entering Basic Event Information**

This lesson describes the process of entering basic event information including mean failure rate, probabilities, and uncertainty data.

##### **Lesson 5. Graphical Output**

This lesson describes the steps required to print event tree and fault trees to a printer or a file. The lesson also briefly describes how to export SAPHIRE graphic in the form of a metafile to be used by other software packages.

## **Lesson 6. Generating, Displaying, and Reporting Fault Tree Cut Sets**

This lesson describes the process of solving and quantifying fault tree cut sets. The lesson also includes information pertaining to displaying and printing cut sets. For this lesson, cut sets will be produced for the three fault trees developed in Lesson 3.

## **Lesson 7. Fault Tree Uncertainty Analysis**

This lesson describes the process of performing an uncertainty analysis for the fault trees created in Lesson 3.

## **Lesson 8. Generating Event Tree Cut Sets**

This lesson describes the process of linking an event tree and solving sequence cut sets. The lesson also includes information pertaining to displaying and printing sequence cut sets reports. For this lesson, cut sets will be produced for the event tree developed in Lesson 2.

### **Advanced Lessons**

The Advanced Lessons section (Section 3) covers the following topics:

## **Lesson 10. Creating a Flag Set**

This lesson describes the construction and use of flag sets and guides the user on how to implement them into SAPHIRE.

## **Lesson 11. The “Linking” Rule Editor**

Lesson 11 introduces the user to SAPHIRE “Link Event Tree” rule editor. The rule editor allows you create rules that affect the event-tree linkage process and sequence generation.

## **Lesson 12. Post-processing Rules**

Lesson 12 introduces the user to the “Post-processing Rule” editor of SAPHIRE. The post-processing rule editor allows you to create rules that affect existing cut sets in a “post-processing” manner.

The tutorial assumes that the user has performed or has general knowledge of the following:

- The SAPHIRE program has been successfully loaded onto a personal computer as described in Appendix A of the SAPHIRE Reference Manual. The user is also familiar with the operations of Microsoft Windows™ and a standard mouse.
- The user has a basic knowledge of event trees and fault trees.
- The user has read Section 2.0 of the SAPHIRE Reference Manual (Volume 3 of this report) that provides an overview of the SAPHIRE software, menu structure, and database concepts.

## 2. BASIC LESSONS

The purpose of the Basic Lessons section is to familiarize the user with the fundamental features and operations of SAPHIRE software. In the following eight lessons, a simple example is developed that models the frequency of a (hypothetical) trucking incident that results in the loss of cargo. While creating this model, the user will construct an event tree and several fault trees; enter basic event data; solve fault tree cut sets; and link and solve sequence cut sets.

For this tutorial, failure/success logic was used to develop an event tree to calculate the frequency that a -truck hauling cargo encounters an obstruction in the roadway (downed tree, fallen rock, stalled automobile, etc.) that causes an accident that result in the load being separated from the truck. The time of interest is over a one-year period. The event tree model consists of the initiating event, IE-OB, and three top events (or “systems”) that could prevent the end-state (load separation) from occurring. The initiating event, IE-OB, is the number of occurrences in a period that initiate the sequence of events. For this case, it is assumed there are approximately three shipments a week over 52 weeks. In addition, for discussion of the example, it is assumed that experience has shown that for about one-half of the time, the driver reports that they encountered some sort of obstruction in the roadway during the route. These analysis boundary conditions will be embodied in the example model.

### 2.1 Lesson 1 – Adding and Selecting a Project

In Lesson 2.1, the user will add a new project called PRACTICE. This is the first step in developing the sample example.

SAPHIRE 8 is a menu-driven application. The Main Screen consists of a title bar, menu bar, search bar, probabilistic risk assessment (PRA) object lists, and a working area – this working area is known as the **General Analysis** area. The main menu is shown in Figure 1. Note that the title bar appears at the top of the window, the menu bar appears directly below the title bar, the search bar appears below the menu bar, the PRA object lists appear on the left, while the working area appears on the right.

SAPHIRE allows the user various alternatives to set up the objects list structure (via the View menu option). To select a menu option, maneuver the mouse pointer over the desired option in the menu bar and click the left mouse button.

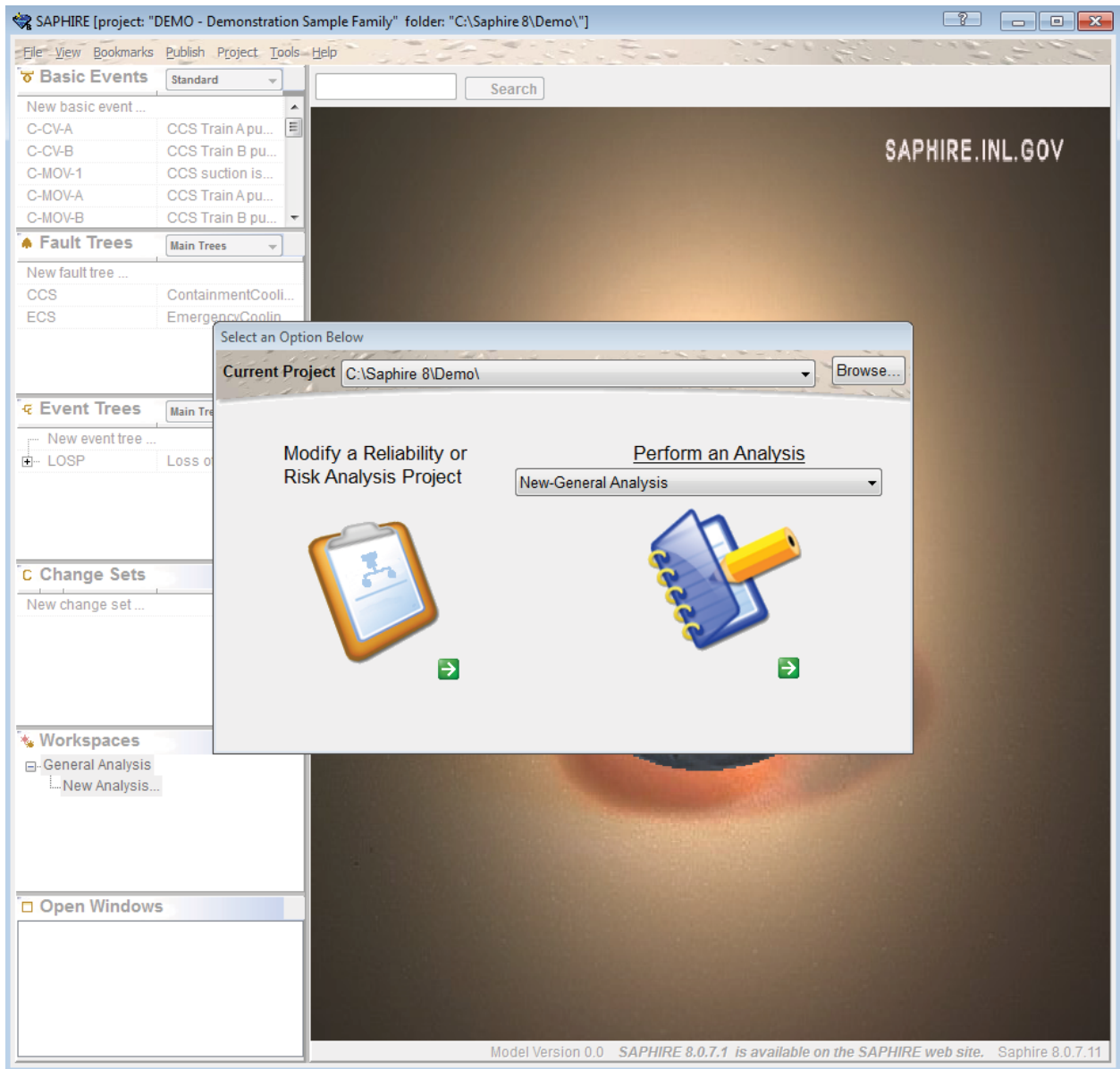


Figure 1. SAPHIRE 8 main screen (General Analysis area) upon startup



### STEP 1: Add a New Project

A SAPHIRE database (or “project”) structure is any grouping of fault trees, event trees, cut sets, and basic events with their respected reliability and uncertainty data. Each project is setup in a separate Windows™ based folder. Therefore, for any new analysis to be accomplished, the first step is to set up a new SAPHIRE database project.

In this step, the user will add a project called PRACTICE. This project will be used throughout the tutorial to store and develop the example problem database.

To add a project in SAPHIRE, perform the following steps:

Select File on the Menu Bar, then New → Project. SAPHIRE will display the screen to enter the new project information. The “New Project” dialog will appear (Figure 2). Type in the new project name (and optional description), PRACTICE and click on OK. SAPHIRE indicates where the new project will be located.

The default setup for SAPHIRE is to create project sub-folders in the SAPHIRE main directory (i.e. C:\SAPHIRE 8). However, the user has the ability to change both to the location of the project folder (via Parent Directory) to any location by clicking on the “Browse...” button and the folder name. Click on OK to create the new project folder and initial project files inside that folder.

SAPHIRE creates the new project folder and automatically places the user in the new project, PRACTICE.

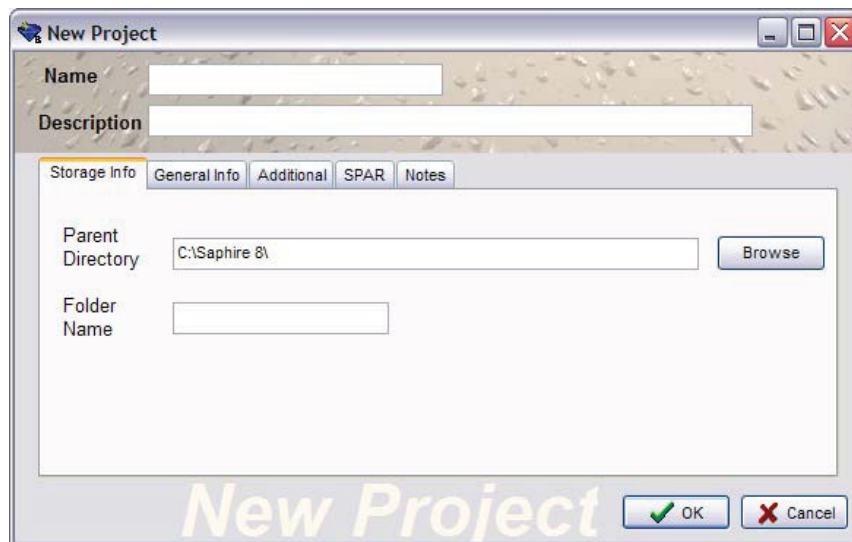


Figure 2. New project dialog

## STEP 2: Select an Existing Project (Optional)

Since the new project created in SAPHIRE is automatically selected after it has been created, this step is not necessary to perform in order to continue with the example problem. However, if the user wanted to go back to another previous project, this step would be necessary.

To select a project in SAPHIRE, the following steps are required:

Select File on the Menu Bar, then Open existing project. In the browse dialog, select the desired project folder using the drop down menu button or the “Up Folder” button as shown in Figure 3.

Select the project name “.SRA” file (i.e. <project name>.SRA ) and click on Open. The project is now selected and available for analysis. Note that older files in SAPHIRE were identified with a FAM.DAT file rather than the .SRA file. In addition, SAPHIRE 8 is able to open compressed (using the ZIP file format) files – consequently files with a “.ZIP” extension will appear in the browse dialog. Opening a .ZIP project will over-write the project (if any) that exists within that same folder.

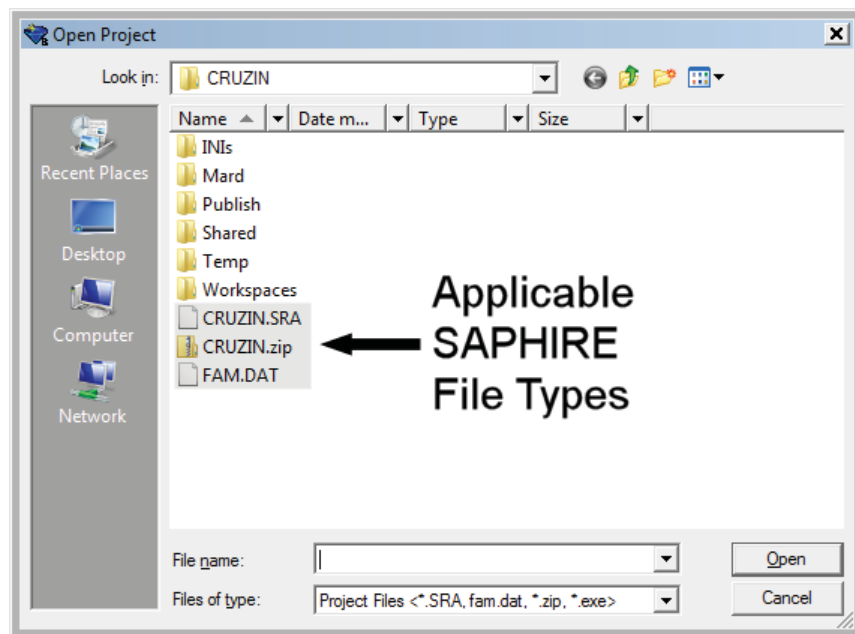


Figure 3. SAPHIRE open project dialog

## 2.2 Lesson 2 – Building Event Trees

In Lesson 2.2, the user will create the “OBSTRUCT” event tree shown below in Figure 4 (this will be the finished event tree). This event tree is used to generate the sequence logic for the example problem. The exercises in this lesson will guide the user through the process of creating an event tree in SAPHIRE.

Shown in Figure 4, the initiating event, IE-OB, is depicted by an initial horizontal line, with systems (top events) connected in a branching system with an “up” branch indicating success and the down branch indicating failure. The event tree top events are questioned from left to right. A top event can also be passed indicating that no success/failure questioned was posed and the next system top event is questioned. The unique combination of system failures and/or successes from the initiating event to the end-state defines the “sequence” for that particular path. An “end-state” is a group of accident sequences, which share certain characteristics that the analyst determines. For this example problem, there is only one end-state called “RELEASE”.

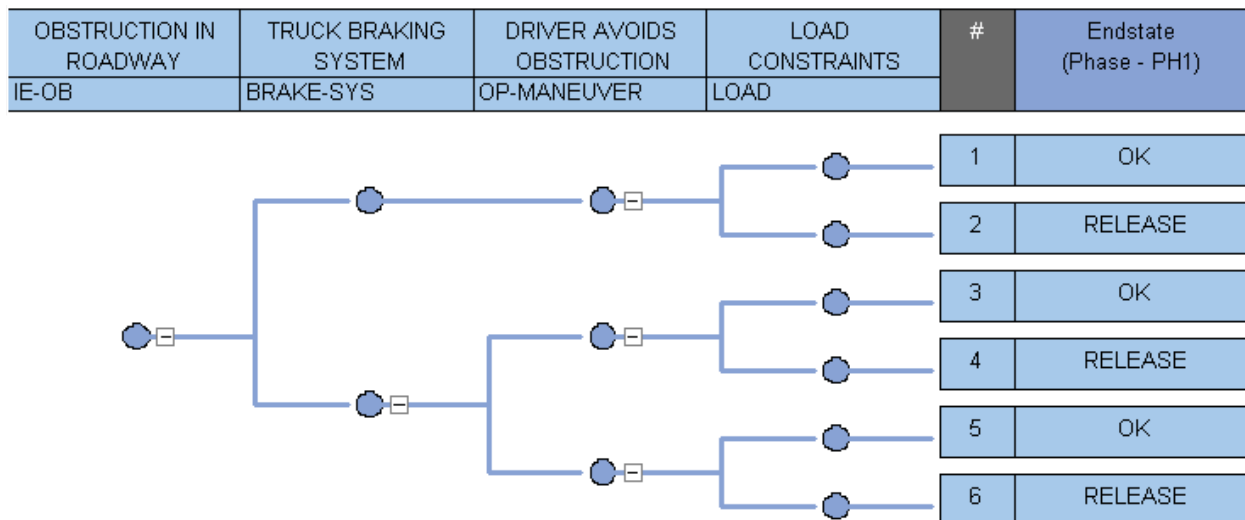


Figure 4. The OBSTRUCT event tree

### QUICK TIP

SAPHIRE has pop menus throughout its menu structure. To evoke a pop-up menu, click on the right mouse button <right click> and move the highlight to the desired function.

### STEP 1: Create a New Event Tree Using the Event Tree Using the Event Tree Graphics Feature

The following information guides the user on creating a new event tree in SAPHIRE. The event tree OBSTRUCT, shown in Figure 4, will be constructed using the detailed steps below.

To develop a new event tree, first access the event tree graphics feature by finding the Event Tree object list in the Main Screen (if this object list does not appear, it can be toggled back on via the View menu option). This object list will display the event trees in the database as shown in Figure 5. The current database for this tutorial has no event trees; therefore, the initial screen will be blank with no event tree names listed.

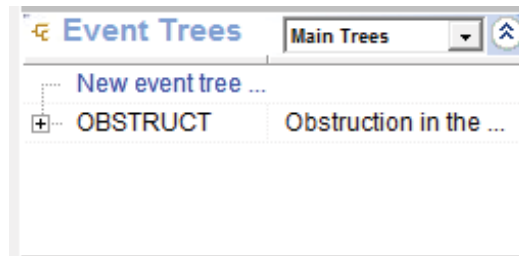


Figure 5. Event tree list

To build the OBSTRUCT event tree, perform the following steps:

In the Event Tree object list, double click the **New Event Tree** option with the left mouse button.

The “Event Tree Properties” dialog will be displayed (Figure 6). Type in the word “OBSTRUCT” for the Name of the event tree and any description desired. The “Group” options are addressed in Volume 3 of this NUREG document. Click on the OK button when completed. There now will be the event tree OBSTRUCT listed in the Event Tree object list.

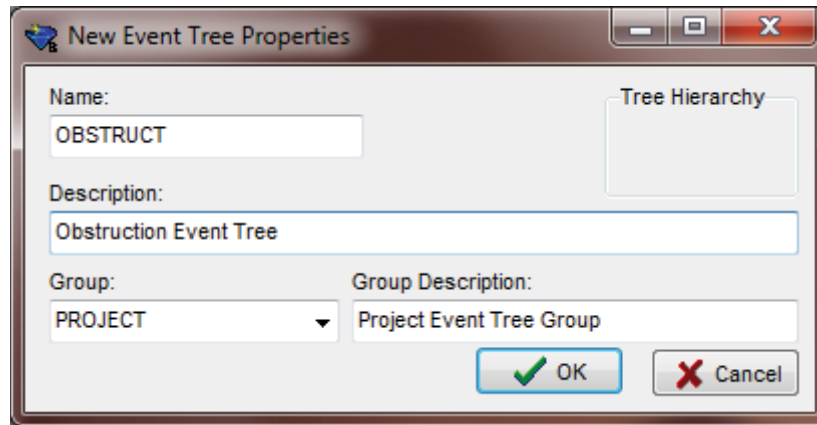


Figure 6. Event tree modify properties dialog

SAPHIRE will bring up the Event Tree graphical editor (Figure 7). Initially, SAPHIRE will place a temporary initiating event (called INIT-EV) and a temporary top event (with a name that varies). To edit either the initiating event or the top event, double click either one.

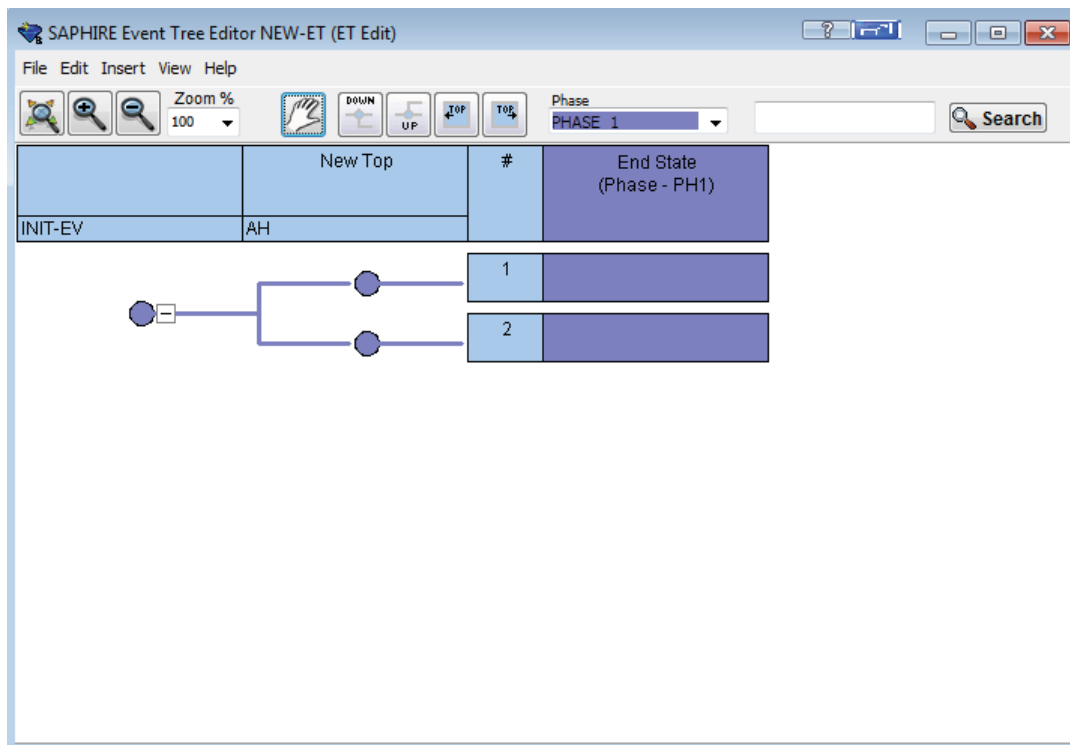


Figure 7. The SAPHIRE 8 graphical event tree editor with a new tree

## STEP 2: Add/Edit Branching to the New Event Tree Using the Event Tree Editor

The following information guides the user on adding branching to the new event tree in SAPHIRE. Typically, in an event tree, the up branch represents success and down branch represents failure. As event tree logic is developed, a top event can be passed (no success/failure questioned asked) or branched. The sequence logic resulting in an end-state is developed as the branches are added.

For the event tree OBSTRUCT shown in Figure 8, after the top event BRAKE-SYS is questioned, the success branch passes over OP-MANEUVER and questions the top event LOAD. The failure branch of BRAKE-SYS questions OP-MANEUVER and on both the success and failure of OP-MANEUVER, questions the top event, LOAD. These specific top events (OP-MANEUVER, and LOAD) will need to be added to the event tree. First though, the existing top event (automatically named AH by SAPHIRE) will need to be renamed to BRAKE-SYS

### Editing Top Events

At times, it may be necessary to edit, delete, or add a top event to the event tree graphic. To modify a top event, double click the top event. The **Edit Event** dialog box will be displayed (see Figure 8). From this box, the Name and Description of the event can be modified.

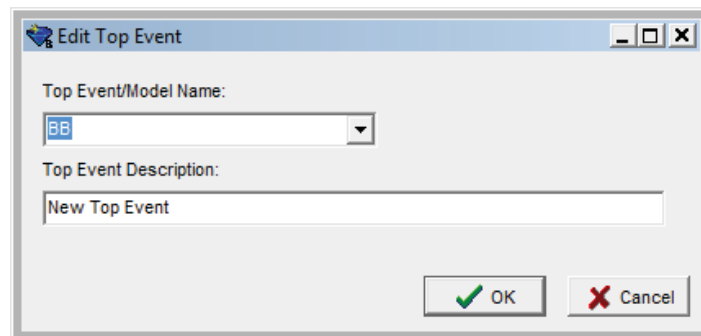





Figure 8. Editing a top event dialog

To delete a top event, select the top event using the Pick cursor. In the menu bar, select Edit → Delete to delete the top event or right click and select delete.

Double click the AH top event, and change the name to BRAKE-SYS. Click Ok.

Now, to add the other top events and branches to the new event tree, perform the following steps.

To add a top event, choose the “Top” object icon from the graphical tool bar  (note that the ← Top will add a top event to the left while the → Top will add a top event to the right). The cursor will change to the “Top” cursor. Move the cursor to the top event in the header that will follow the top event you are adding. Click the left mouse button to add the top event. The **Edit Event** dialog box will be displayed. To add a top after the rightmost top event, move the cursor to the right of the last top event and click the left mouse button.

To add a branch under the top event LOAD, click the “Add Branch” icon to either add a down or up branch . Then, where a branch point is desired, click to round node on the event tree path  under the specific top event. For example, click (left mouse button) on the round node on the success branch of BRAKE-SYSTEM directly under the LOAD top event. Once the round node is clicked, the branch will be automatically added as shown in Figure 9.

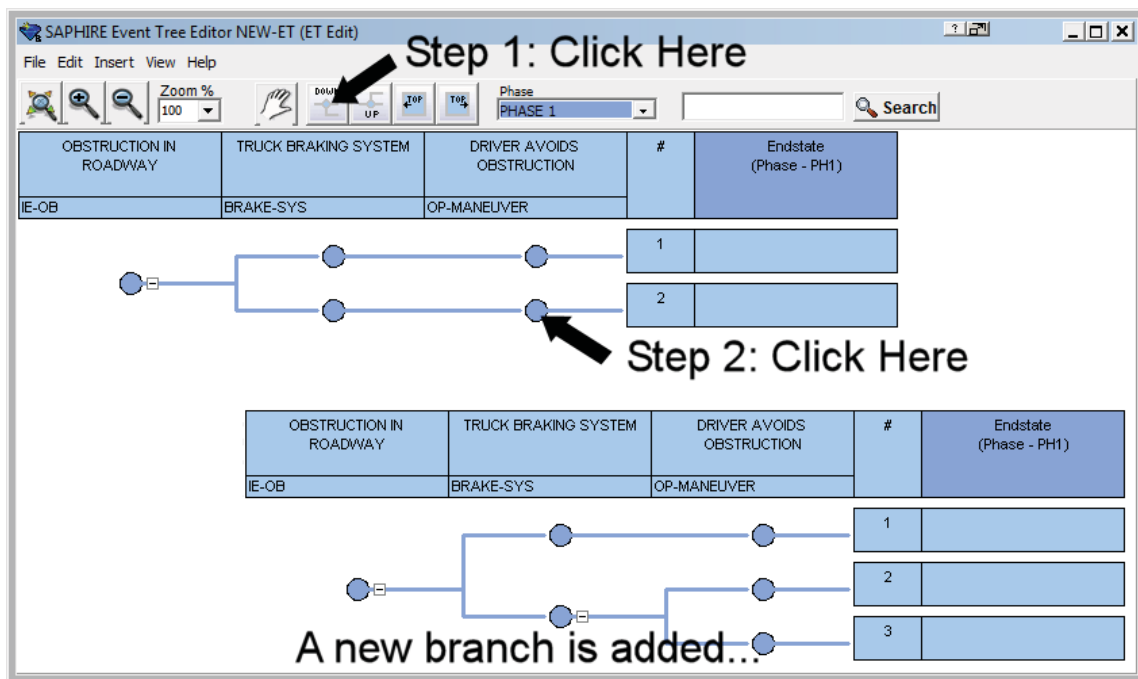



Figure 9. Example of adding a branch point for an event tree sequence

To delete an unwanted branch, use the point cursor  to select the unwanted branch node (turns yellow). Select Edit → Delete from the menu bar, use the pop-menu (right click) and select **Delete**, or just press the delete key. Attempting to delete a node will prompt SAPHIRE to ask for confirmation of the deletion, resulting in the message displayed in Figure 10.

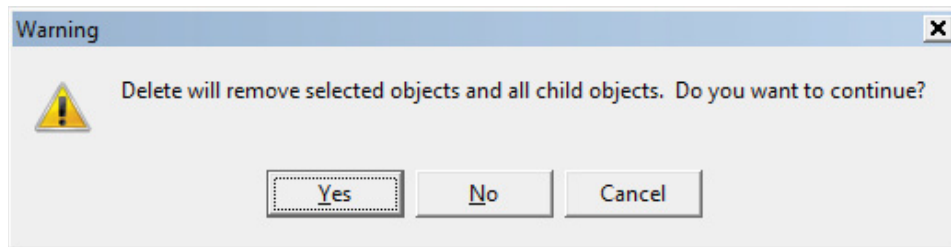


Figure 10. Warning message when attempting to delete an event tree node

Continue to add the branches and top events as shown in Figure 11.

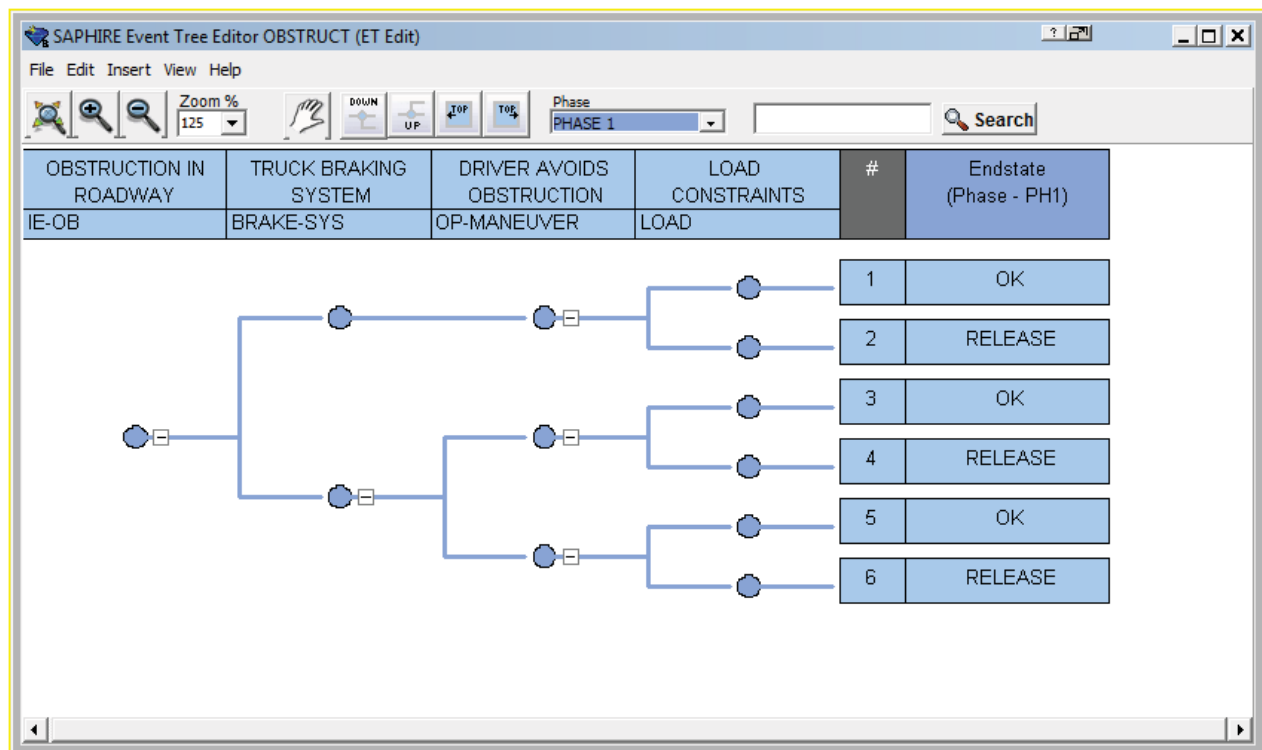


Figure 11. The OBSTRUCT event tree in the graphical editor

### STEP 3: Add/Edit End-States to the New Event Tree Using the Event Tree Editor

This step guides the user on how to add and edit end states on each of the sequences in SAPHIRE. End states are required in order to tell SAPHIRE where to group (or bin) sequence cut sets.



On the far right of the event tree are the end-state headers. There are four header columns available for each event tree and a fifth column reserved for the sequence number. The default for SAPHIRE is to display three of the header columns (end states, frequency, and extra information). However, the user can adjust the number of sequence header columns displayed via the View menu option.

**QUICK TIP**

To change the width of a sequence header, select the **View →Box Size** menu option.

Each sequence in the event tree needs to terminate in some kind of end state. On success branches, this end-state is usually labeled with an OK— however, it does not have to be. In SAPHIRE, if the end-state column entry is OK, begins with the “@” symbol, or is the word “Success,” then logic for that sequence will not be developed. For an end-state with any other 24 character or less entry, sequence logic will be developed. End states are assigned to event tree branches by double clicking the end state “box,” which will bring up the **Edit Sequence** dialog box (Figure 12).

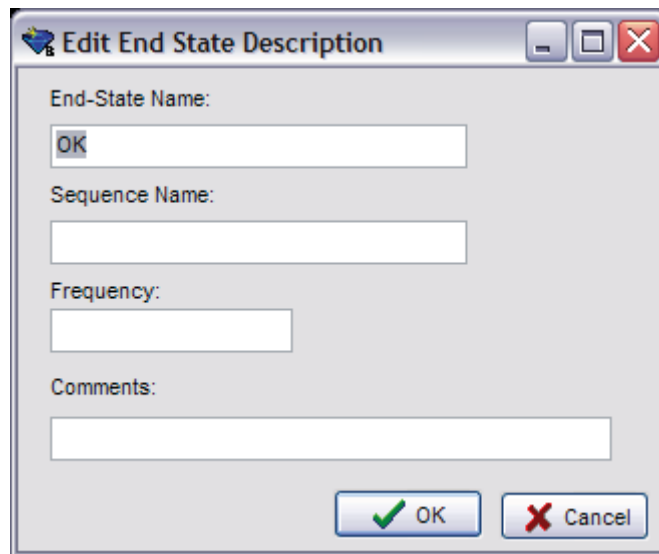
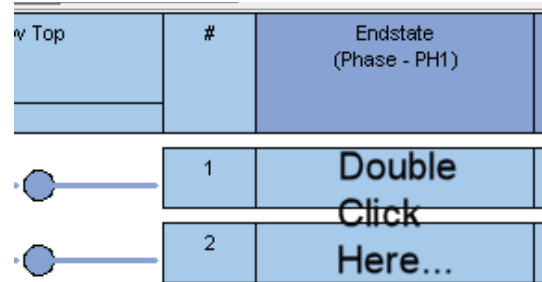


Figure 12. Editing an end state dialog

To enter the end-state names, complete the following steps:


Use the pick cursor to select sequence number 1 by moving the cursor over the first box to the right of number “1” and **double** click the box (with the left mouse button). At the top of Edit Sequence dialog, there are four available sequence information boxes. The fields “End-State Name,” “Sequence Name,” “Frequency,” and “Comments” are used to display text about the sequence. The “End-State Name” field is the name of the sequence end-state. This field needs to be filled in order for SAPHIRE to generate sequences. For success sequences (or sequences to not be evaluated), this field can be either “OK” or “Success.” Alternatively, the sequence can be disabled by right-clicking the end stat and selecting the “Disable End State” option (if this is selected, SAPHIRE will place an “@”<sup>1</sup> at the beginning of the end state name). To turn the sequence back on, right-click it and select “Enable End State.”



When editing sequence number 1, since this is a success branch, type in the word “OK” in the End State Name field. Repeat the editing step for the remaining 2, 3, 4, 5, and 6 sequence numbers. Make sure that all the sequences that end with an “up” branch are labeled OK while the sequences that end with a “down” branch have an end-state labeled RELEASE.

#### **STEP 4: Save the Event Tree and Exit from the Event Tree Editor**


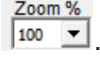
Step 4 of this section saves the event tree graphics in SAPHIRE database and exits the user from the event tree editor. To save the event tree graphics and exit from the event tree editor, perform the following steps:

1. When all the sequence end states are labeled, click on the on File → Save in the menu bar. This will save all modifications under the original event tree name.
2. If a new file name is desired, click on File → Save As and type in the new file name in the File Name box. The event tree name and the file name do not need to be the same.
3. To exit the event tree editor, click on File → Exit or click the close button on top of the dialog window . This will return the user to the main SAPHIRE window.

---

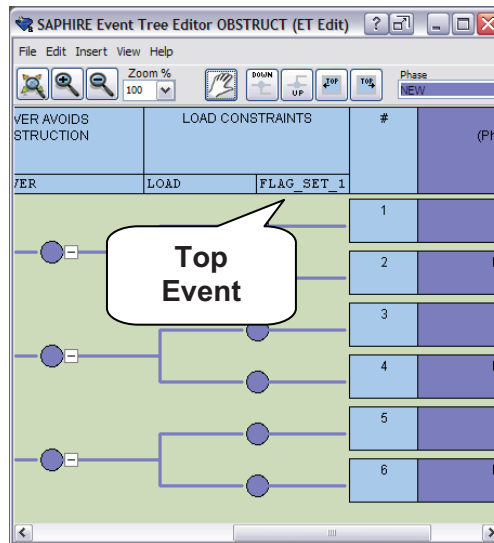
<sup>1</sup> Unlike SAPHIRE version 7, the “@” cannot be typed into the end stat name field.

### QUICK TIP

For modifying larger event trees, the zoom feature can be accessed. To access zoom, click on one of the zoom icons  to zoom in or out, respectively. Alternatively, select the zoom magnification level directly by selecting the zoom % option .

To redraw the event tree to the original full screen view, click on  icon.

Note that if flag sets (which are discussed in Section 3.1) are assigned to a top event (via the Fault Tree properties), the flag set name will appear on the event tree graphic, as shown below.



## 2.3 Lesson 3 – Building Fault Trees

In Lesson 2.3, the user is introduced to fault tree development, SAPHIRE fault tree symbols, and SAPHIRE fault tree modeling conventions. The user will create the basic fault trees used as top events in the “OBSTRUCT” event tree. For this lesson, it is necessary to create three fault trees (BRAKE-SYS, OP-MANEUVER, and LOAD) and save them in the SAPHIRE database.

The top event that were entered in the OBSTRUCT event tree are further developed as fault trees in this section. Fault tree construction allows a logical structure to be developed that is used to analyze possible failures and to calculate the probability of failure.

The following two sections below reviews the fault tree symbols used in building fault trees in SAPHIRE.

### 2.3.1 Fault Tree Editor Symbols

This section reviews the basic event and gate symbols in SAPHIRE used in fault tree construction. All the symbols below are buttons and are located in the graphical tool bar of the Fault Tree Graphical Editor.



**BE** This symbol represents a basic event such as a hardware failure, human error or an adverse condition.



**AND** Represents an AND gate. The logic operation for this gate requires all inputs into the AND gate must occur for failure to occur.



**OR** Represents an OR gate. The logic operation for this gate requires only one of the total number of inputs into the OR gate to occur for failure to occur.



**N/M** Represents an “N-OUT-OF-M” gate (or N/M gate). The logic operation for this gate requires that N of the M inputs into the gate must occur for failure to occur. For example, for a 2/4 gate, any combination of 2 of the 4 input events must occur for a failure to occur.



**TRAN** This symbol represents a TRANSFER gate. The transfer gate indicates that logic is continued from some other location. The location could be a new separate fault tree (top event) or a gate from some other location in the same fault tree.

### 2.3.2 Creating a Fault Tree in SAPHIRE

Fault trees may be created with the Fault Tree Graphical Editor. This method will be discussed in detail for the example problem and will be used for the construction of the fault trees. Note that Figure 13, Figure 14 and Figure 15 illustrate three fault trees, BRAKE-SYS, OP-MANUVER, and LOAD, respectively. These fault trees will be created and will be used as the three system top events in the event tree created in the previous section. The following three fault trees will be created in this section:

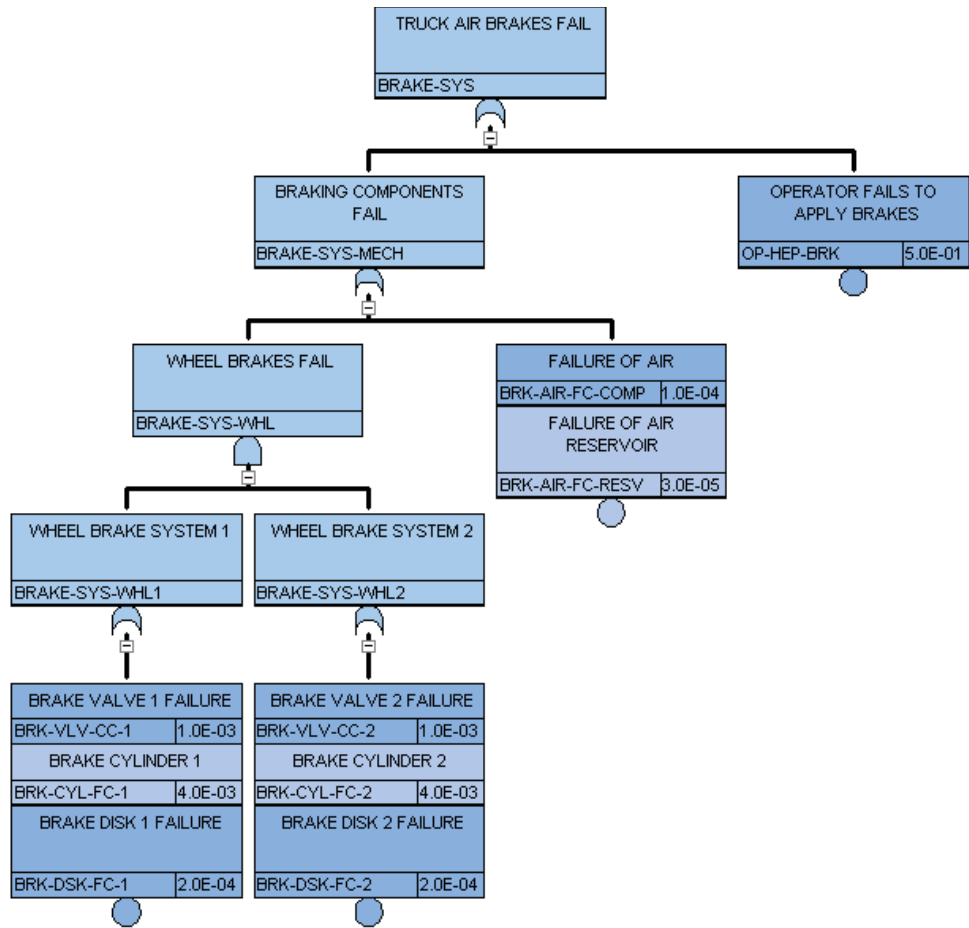


Figure 13. BRAKE-SYS fault tree

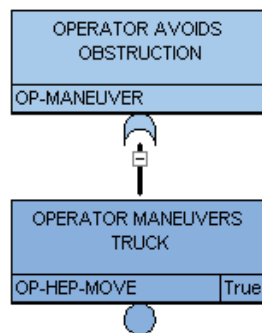


Figure 14. OP-MANEUVER fault tree

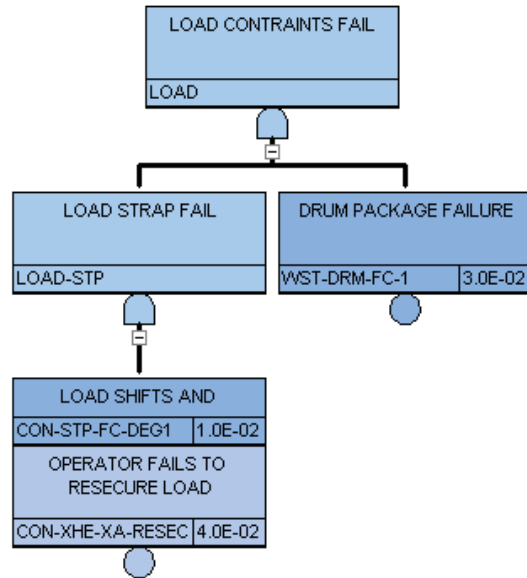


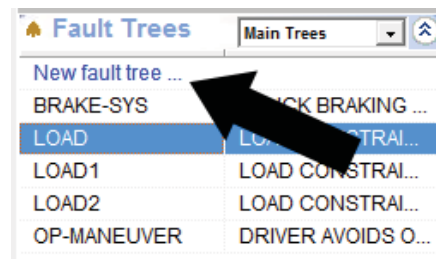
Figure 15. LOAD fault tree

The following steps guide the user on creating a new fault tree in SAPIRE. The three fault trees shown above in Figure 13 through Figure 15 will be constructed using the detailed steps below.

**STEP 1: Access the Fault Tree Graphical Editor**

The fault tree BRAKE-SYS will be created first. To access the graphical fault tree editor in SAPHIRE, perform the following steps:

Access the fault tree graphical editor by double clicking the New fault tree... option on the main fault tree object list (or, alternatively, click on File → New → Fault tree). This will bring up the fault trees property dialog editor (Figure 16), where the initial properties such as name and description can be entered. Enter the applicable name and description, then click the OK button when completed. Clicking OK will bring up a blank fault tree editor, with the top event name automatically entered as the fault tree top gate as shown in Figure 17. All the available graphical features needed to complete any fault tree are displayed at the top of the editor in the tool bar.



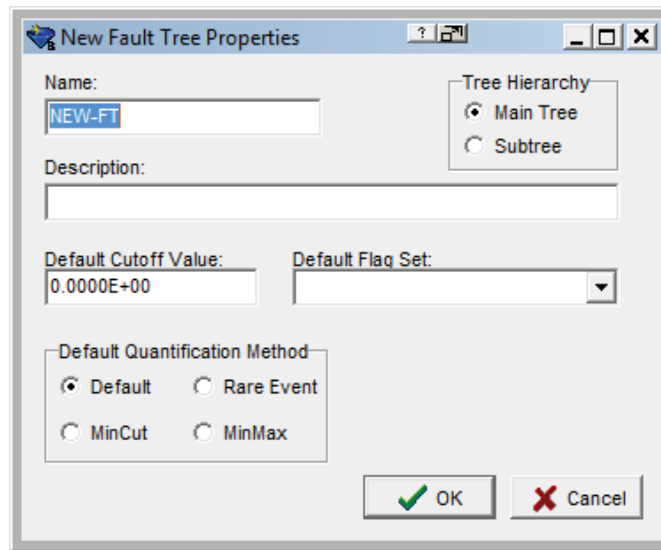


Figure 16. Fault tree properties dialog

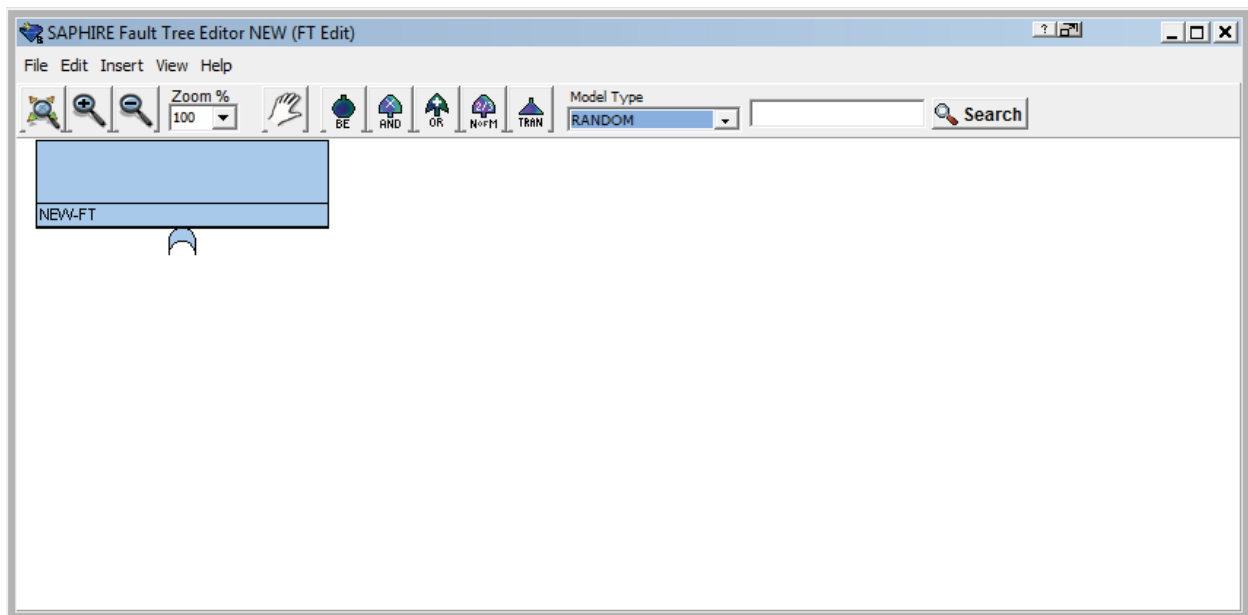


Figure 17. Graphical fault tree editor for a new fault tree

## STEP 2: Selecting and Arranging Fault Tree Logic Symbols

The symbols in fault tree BRAKE-SYS will be added. To add symbols to the fault tree, perform the following steps:

To begin building the logic for the fault tree, select the desired object button from the tool bar. For the fault tree BRAKE-SYS, the top gate is an “OR” gate. The default top gate is an OR gate and should be (already) called BRAKE-SYS. However, if one wanted to modify a gate, the gate object should be double clicked (with the left mouse button) which will bring up the gate edit dialog (Figure 18). If the top gate is being edited from the gate editor, only the type of gate (OR, AND, N/M) can be modified.

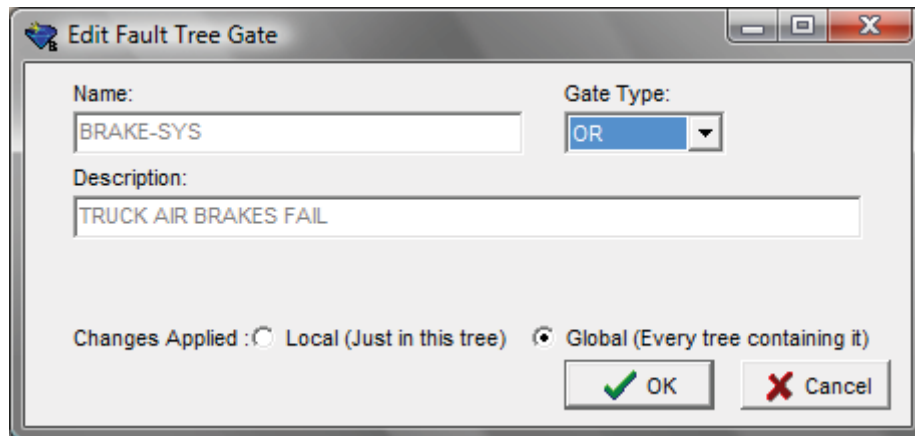


Figure 18. Gate editor dialog in the fault tree

The graphical editor in SAPHIRE 8 uses a drag-and-drop approach. Consequently, instead of having to draw manually fault tree logic, the fault tree structure is created by dropping logic gates or basic events directly onto other gates/events. For example, to create the BRAKE-SYS fault tree for this tutorial, the following steps would be applied:

1. Click the OR button icon.
2. Click the BRAKE-SYS top gate to drop the BRAKE-SYS-MECH gate onto it.
3. Click the basic event icon.
4. Click the BRAKE-SYS top gate to drop the OP-HEP-BRK basic event onto it.
5. Click the AND button icon.
6. Click the BRAKE-SYS-MECH gate to drop the BRAKE-SYS-WHL gate.

and so on. With the drag-and-drop approach, SAPHIRE will automatically align and connect gates/events. These steps above are illustrated in Figure 19.



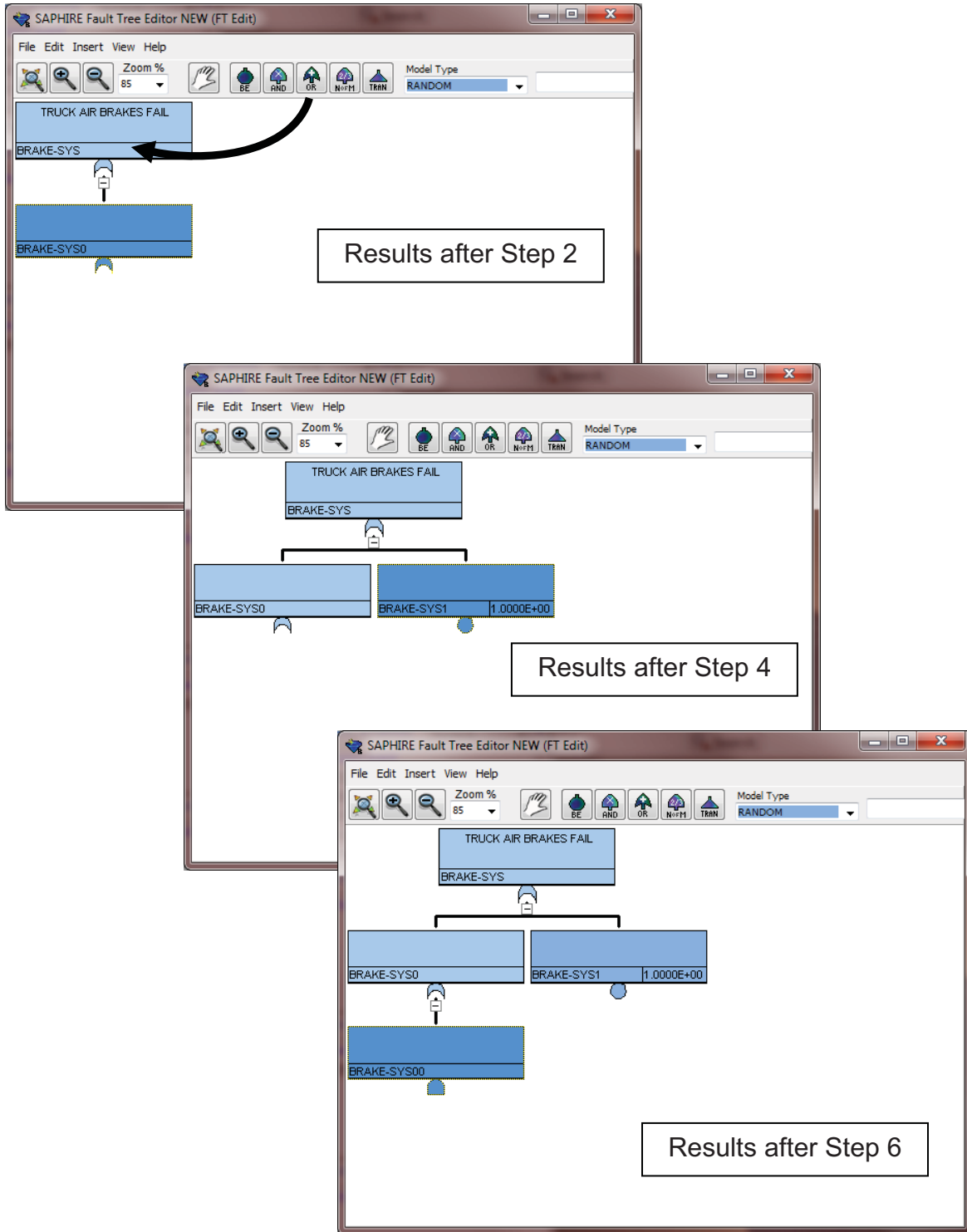



Figure 19. Fault tree development steps and results

### STEP 3: Modifying Gate and Basic Event Properties

After the various logic symbols in the fault tree have been added and connecting lines between the various symbols, the final step in created the fault tree is to label the gates and basic events with the appropriate names and descriptions. To add or edit symbol attributes to the fault tree BRAKE-SYS perform the following steps:

In the fault tree editor, select the desired shape to be modified first clicking the “pick” icon  then double clicking the applicable object. Double clicking a gate will display the “**Modify Property**” window appear as shown in Figure 20. From this window, the user can change the gate name, description, and type. In addition to these options, the nature of the change must be specified, where two options are available: (1) Local or (2) Global. Local changes will affect only the fault tree being edited, where global changes will affect all fault trees containing the gate (or event) being modified. For example, if gate BRAKE-SYS-MECH were renamed to BK-SYS-M and the global option checked, then any fault tree containing this gate will appear with the new renamed gate. If the local option were checked instead, then other fault trees will keep the BRAKE-SYS-MECH named gate, while the fault tree being edited will now contain the BK-SYS-M gate.

SAPHIRE automatically assigns default names to fault tree gates, basic events, and tables. These are usually re-named by the user. SAPHIRE does require that top gate in a fault tree must be identical to the name given to the fault tree. To change the gate name, enter the new gate name, BRAKE-SYS, in the Name entry box located at the top of screen. In addition, the description, “TRUCK AIR BRAKES FAIL,” can be entered in the Description entry box

Click on OK when finished to return to the *Fault Tree Editor* screen.

Repeat the naming procedures above for all the fault tree gates and basic events shown in Figure 13 through Figure 15.

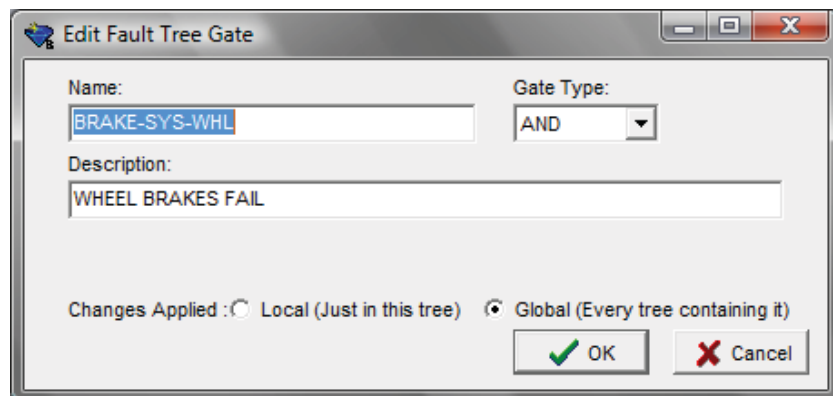


Figure 20. Gate modify dialog

#### **STEP 4: Ending the Fault Tree Editing Session**

When the fault tree is complete with the desired attributes, the fault tree can be saved and the fault tree editor exited with the following steps:

1. To save the fault tree graphic, select File → Save to save the fault tree with the same name as the top gate. If one wishes to save the fault tree with another name, select File → Save as...
2. For fault tree, BRAKE-SYS, save the file under the name “BRAKE-SYS”.
3. To exit without saving, in the Fault Tree Editor menu bar: Click on File → Exit. Choose No to quit without saving.

Repeat all the steps outlined in this section for the remaining fault trees OP-MANEUVER and LOAD. When completed, the user now has fault tree logic for all the top events for event tree, OBSTRUCT.

#### **QUICK TIP**

Fault tree logic can quickly be copied in two different ways.

1. In the fault tree editor, the File → Save As... option will save the current tree to a second tree with a new name (and new top gate name).
2. Any gate or basic event may be selected, then right-click and select Copy. Then a copy of that logic structure may be pasted elsewhere in the fault tree by select another gate, right-clicking, and selecting Paste.

## **2.4 Lesson 4 – Basic Event Information**

In this lesson, the user is introduced to basic event data entry into SAPHIRE. For the example problem, the lesson requires that the user input the values shown in Table 1.

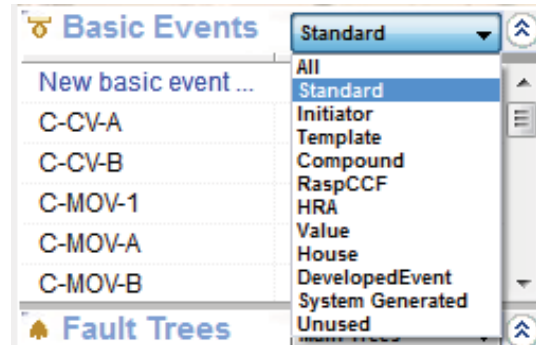
During the process of event tree and fault tree creation that was completed in Sections 2.2 and 2.3, the basic event names, gates, event tree top events (i.e. fault trees), and initiating events were added to the project. However, failure rates or probabilities and initiating event frequencies are not included and must be entered into SAPHIRE. In this lesson, failure data are entered into the project for use by SAPHIRE during an analysis. Note that new basic events may be added to the project by using the “Add basic event...” option from the main window or via the File → New → Basic Event option.

For the OBSTRUCT example, the data values will be added to SAPHIRE database to the basic events listed in Table 1. Note that the failure rates used are for demonstration purposes only and do not represent actual performance.

## 2.4.1 Entering Basic Event Information

Basic event data loaded or entered into SAPHIRE is placed into the project database. The basic event data in Table 1 will be entered into the SAPHIRE database. To access the graphical fault tree editor in SAPHIRE, perform the following steps:

To modify basic event data, access the list of basic events from the main window or double click a basic event from the fault tree graphical editor. The list of basic events appears (if enabled under View) on the main SAPHIRE window. The list of basic events can be filtered via the drop-down box, including showing all, standard (not top or system events), initiating, templates, compound, value, common-cause failure, human reliability, developed, system, house, and unused events.



On the main list of basic events, right clicking on an event will allow: editing the event, bookmarking the event, or deleting the event.

A listing of all the basic events in the database will be displayed as shown in Figure 21.

New basic event ...	
BRK-AIR-FC-COMP	FAILURE OF AIR CO...
BRK-AIR-FC-RESV	FAILURE OF AIR RE...
<u>BRK-CYL-FC-1</u>	<u>BRAKE CYLINDER 1...</u>
BRK-CYL-FC-2	BRAKE CYLINDER 2...
BRK-DSK-FC-1	BRAKE DISK 1 FAIL...
BRK-DSK-FC-2	BRAKE DISK 2 FAIL...
BRK-VLV-CC-1	BRAKE VALVE 1 FAIL...
BRK-VLV-CC-2	BRAKE VALVE 2 FAIL...
CON-STP-FC-BRK	LOAD CONSTRAINT...

Figure 21. Basic event listing screen

**Table 1. Basic event data values for the tutorial.**

Basic Event Name	Failure Probability	Uncertainty Distribution	Error Factor
BRK-AIR-FC-COMP	1.0E-04	Lognormal	5
BRK-AIR-FC-RESV	3.0E-05	Lognormal	3
BRK-CYL-FC-1	4.0E-03	Lognormal	5
BRK-CYL-FC-2	4.0E-03	Lognormal	5
BRK-DSK-FC-1	2.0E-04	Lognormal	10
BRK-DSK-FC-2	2.0E-04	Lognormal	10
BRK-VLV-CC-1	1.0E-03	Lognormal	5
BRK-VLV-CC-2	1.0E-03	Lognormal	5
CON-STP-FC-BRK	1.0E-01	Lognormal	10
CON-STP-FC-DEG1	1.0E-02	Lognormal	10
CON-XHE-XA-RESEC	4.0E-02	Lognormal	5
IE-OB	1.0	Lognormal	3
OP-HEP-BRK	5.0E-01	Lognormal	10
OP-HEP-MOVE	1.0	Lognormal	10
OP-MANEUVER	1.0	Lognormal	10
WST-DRM-FC-1	3.0E-02	Lognormal	5
WST-DRM-FC-2	1.0E-02	Lognormal	5

Note: These values in Table 1 are for demonstration purposes only and are not applicable for any actual application.

Highlight the basic event to be modified (For this first example, highlight the basic event BRK-AIR-FC-COMP) and then click the right mouse button. Select Edit Basic Event from the drop down menu. Alternatively, **Double Click** on the basic event to be modified.

The **Edit Basic Event** screen appears as shown in Figure 22. From this window, all basic event data including uncertainties can be entered. The **Modify Event** screen has “Tab” buttons across the top. The Failure Model tab is accessed automatically when the basic event modify dialog is called. This tab is used to input all failure and uncertainty data. The other tabs are used for adding other descriptive attributes, notes, and other advanced features. Specific details for these features can be found in the SAPHIRE Technical Reference (Volume 2 of this NUREG).

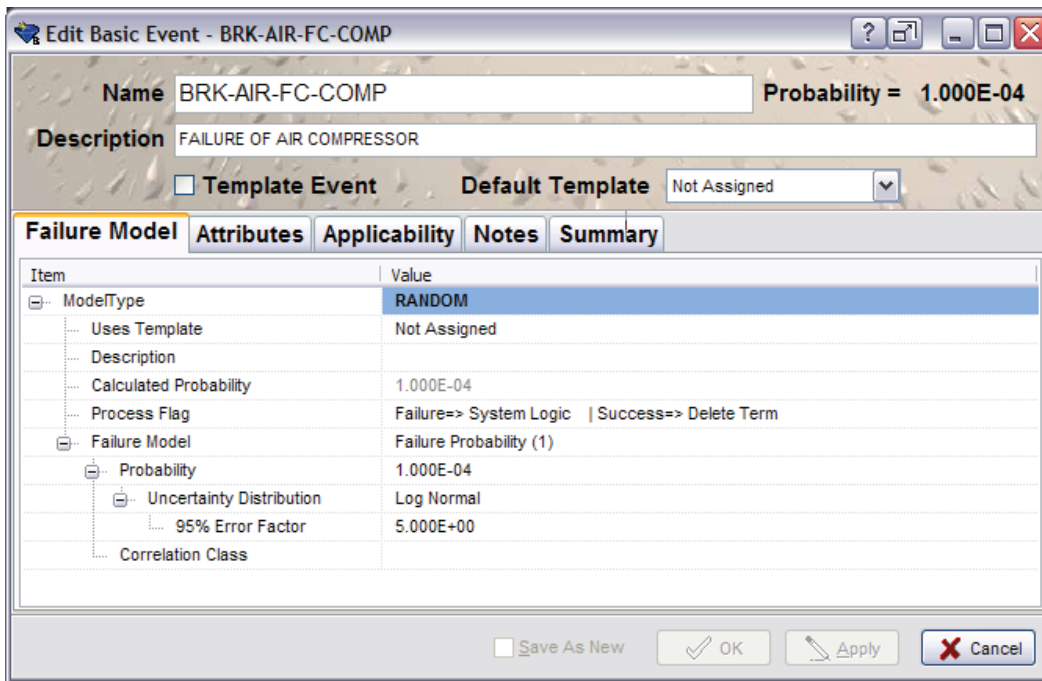


Figure 22. Modify basic event editing dialog

The left side of the Edit Basic Event dialog is where the failure information is recorded. The basic event name (up to 24 characters) and description (up to 120 characters) may also be modified. The Failure Model tab consists of a tree-view control listing all of the applicable options required to specify the event’s failure model. In SAPHIRE 8, a failure model is assigned to each event’s Model Type (if more than one, as assigned in the Applicability tab). Note that the Calculated Probability is not editable, it will be determined from the settings below it. The process flag is used if the basic event is used directly in the event tree logic to control how its failure (down branch) or success (up branch) is used. If the event is going to use a template,

select the template event from the list. The Failure Model controls how the event's probability is calculated.

For the basic event, BRK-AIR-FC-COMP, the Failure Model should be "Failure Probability (1)" (the default type). The data entry boxes below the Failure Model are use to enter the various failure data depending on the model being used. For the Failure Probability model, only the **Mean Failure Probability** value is required. From Table 1, enter the value, 1.0E-4 into the **Probability** box.

Below the failure probability, the Uncertainty Data is entered. First, the uncertainty distribution must be indicated. For the basic event, BKR-AIR-FC-COMP, the Uncertainty Distribution is "Log Normal." The entry fields below the uncertainty distribution are used to enter the various uncertainty parameters. From Table 1, enter the value "5" into the Error Factor field. A statistical Correlation Class number is not required for this example and should be left blank.

Note that as data is entered, SAPHIRE will perform data verification, consequently it is possible for "data warning" messages to be displayed if improper data is entered. Click on the OK button when all the data is entered.

Repeat all the steps outlined above for the remaining basic events listed in Table 1. When completed, the user now has failure and uncertainty data for all the basic events used by the fault trees in the OBSTRUCT project.

#### QUICK TIP

SAPHIRE has the ability to create "Template Events". With template events, basic event information, such as failure rate and uncertainty data, can be easily copied over to other basic events that require identical data.

To create a template event, click the **Template Events** check box (on the Failure Model tab). This flag tells SAPHIRE that this basic event is a template event and rate information can be transferred to other basic events.

By default, all of the template event characteristics are check. To check or uncheck an already checked box, click on the box next to the applicable field (note that the description check box is on the Attributes tab). Once all desired characteristics are check, click on OK.

To use the rate information of a template event for other basic event, select the desired template from the **Default Template** drop-down list. The rate information and attribute data shown, will then be applied (or copied over) to the basic event.

## 2.5 Lesson 5 – Graphical Output

Lesson 5 guides the user on how to print event trees and fault trees. In Lesson 5, the user will print event the event tree, OBSTRUCT, to a printer and print the three fault trees (BRAKE-SYS, OP-MANEUVER, and LOAD) to a printer. The lesson also briefly describes how to export SAPHIRE graphics to be used in other software packages.

Once the event tree and fault tree logic have been developed and saved, the graphical logic representation can be printed directly to a printer or exported to a graphical output file capable of being loaded into other software packages.

### 2.5.1 Printing Event Tree Graphics

Printing an event tree can be accomplished in two ways: (1) from the **event tree graphical editor** or (2) using the Publish feature of SAPHIRE (see Figure 23). To print an event tree from the **event tree graphical editor** (access the event tree editor by double clicking the event tree). Then select File → Print.

To report the event tree graphic, select (from the main menu) the Publish → Event Tree Report option (Graphics Report). The available reports and output format types are shown in Figure 23.



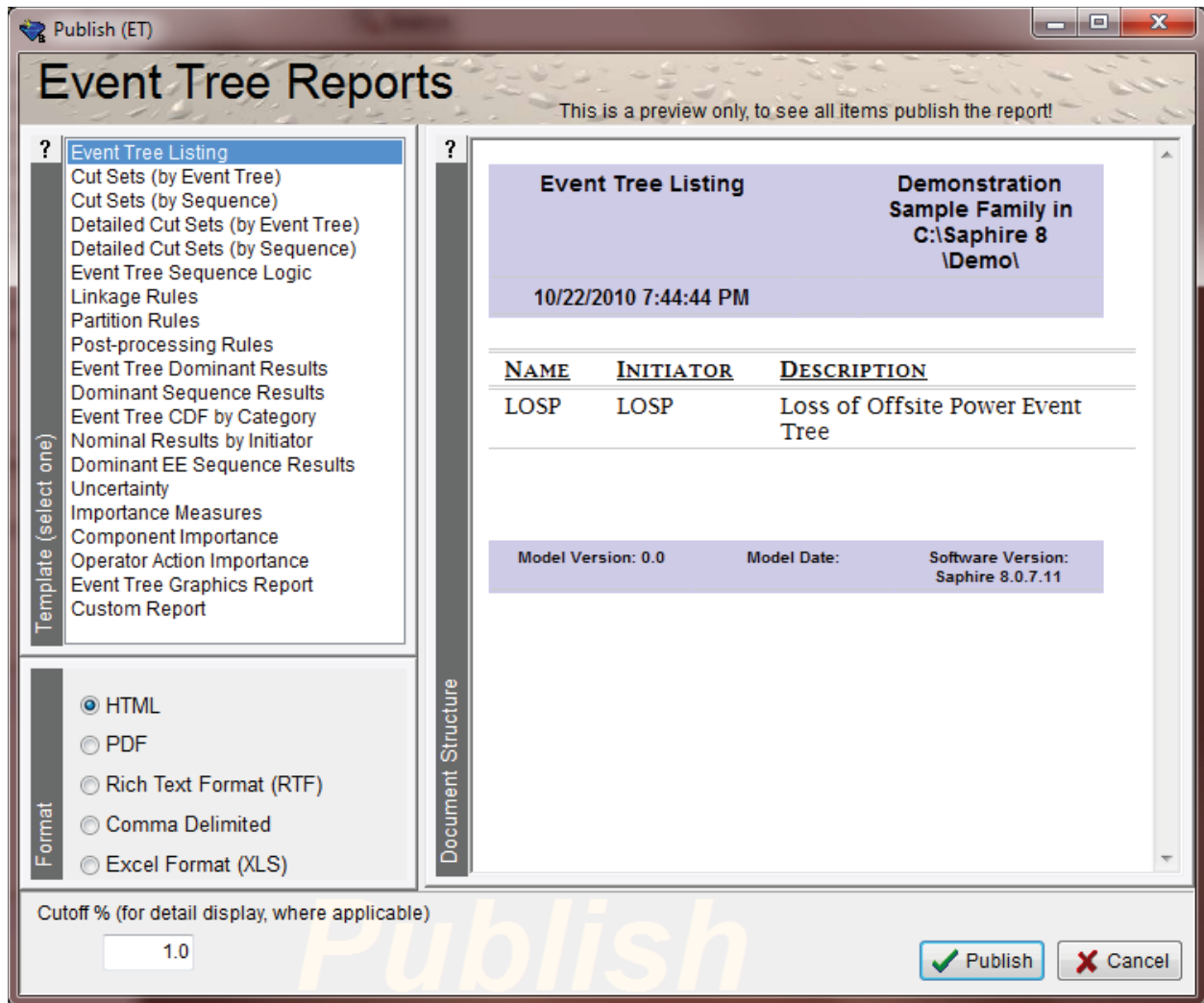


Figure 23. Event tree publish reports screen

To export the event tree graphics, either (2) enter the graphical editor (as described earlier) and select the File → Export Image option or (2) select the Event Tree Graphics Report from the Publish option (as shown in Figure 24). Note that from the graphical editor only two format options are available, either metafiles (.wmf) or bitmaps (.bmp). From the publish option, four output options are available, HTML, PDF, RTF, and XLS. Note that the HTML and PDF outputs bring up a “viewer” for the respective format, while RTF files are sent to Microsoft Word and XLS files are sent to Microsoft Excel.

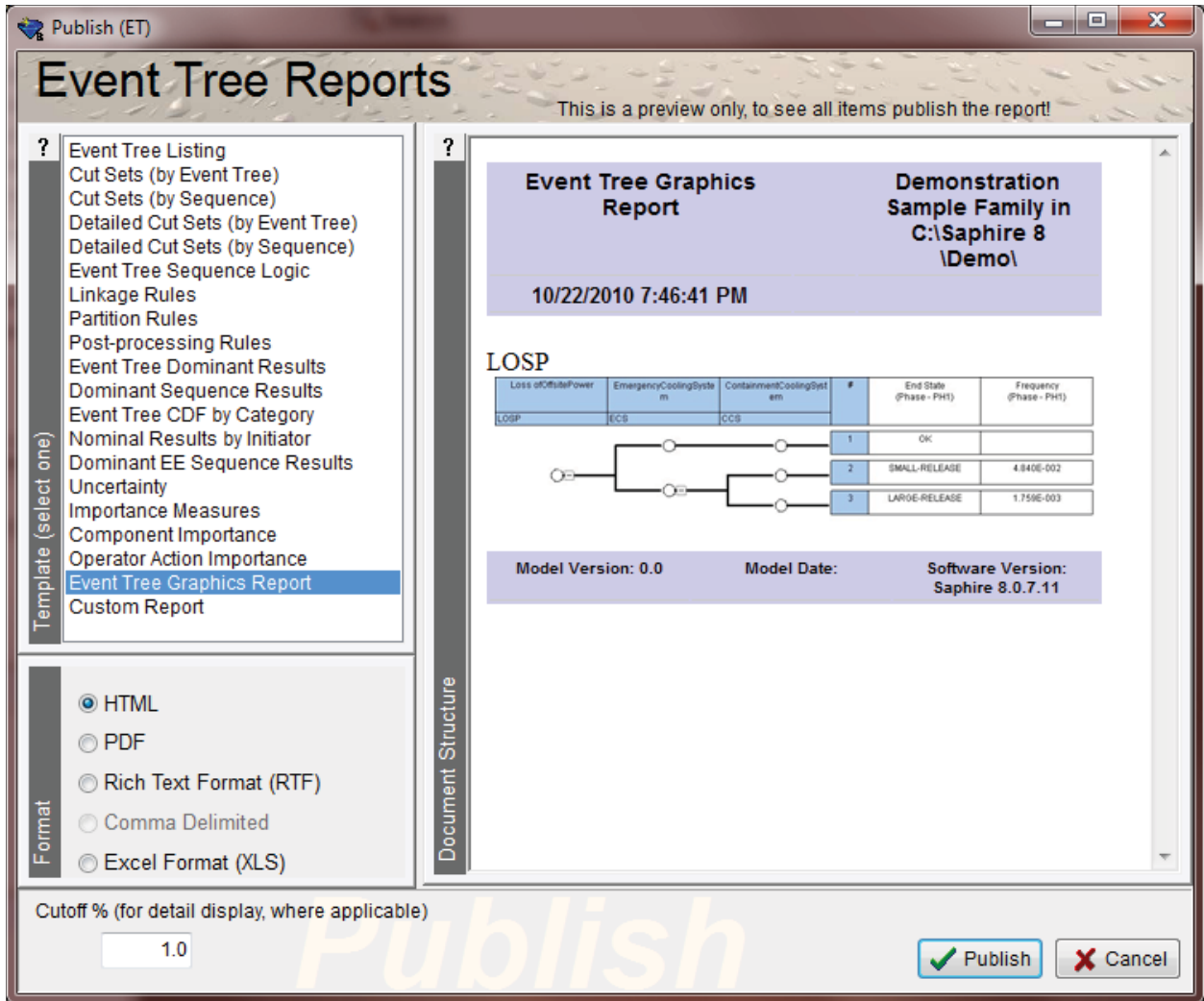


Figure 24. Event tree graphics publish reports screen

## 2.5.2 Printing Fault Tree Graphics

Printing a fault tree can be accomplished in two ways: (1) from the **fault tree graphical editor** or (2) using the Publish feature of SAPHIRE (see Figure 25). To print a fault tree from the **fault tree graphical editor** (access the fault tree editor by double clicking the fault tree). Then select File → Print.

To report the fault tree graphic, select (from the main menu) the Publish → Fault Tree Report → Fault Tree Graphics Report option.

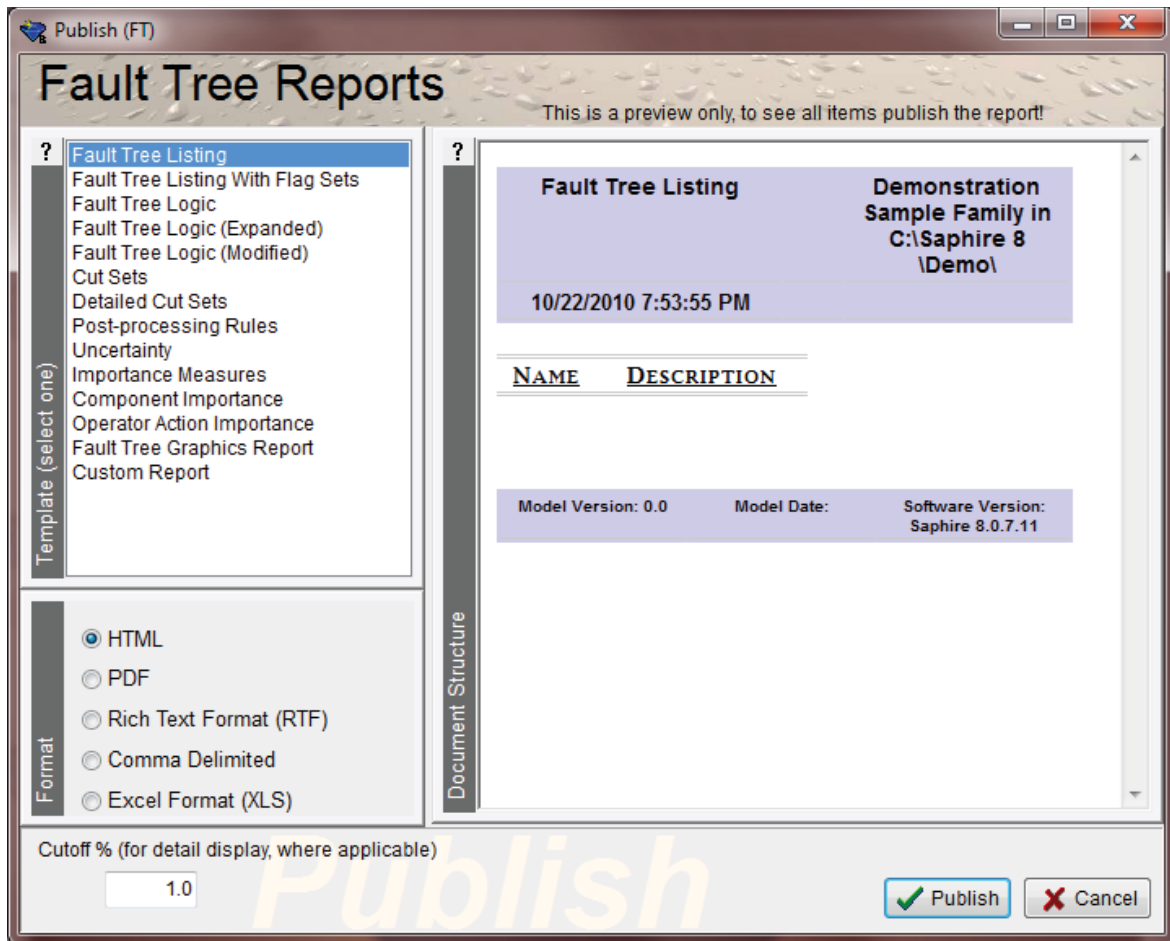


Figure 25. Fault tree publish reports screen

To export the fault tree graphics, either (2) enter the graphical editor (as described earlier) and select the File → Export Image option or (2) select the Fault Tree Graphics Report from the Publish option. Note that from the graphical editor only two format options are available, either metafiles (.wmf) or bitmaps (.bmp). From the publish option, four output options are available, HTML, PDF, RTF, and XLS. Note that the HTML and PDF outputs bring up a “viewer” for the respective format, while RTF files are sent to Microsoft Word and XLS files are sent to Microsoft Excel. Note that basic events and transfer gates will print according to their stacked appearance on the fault tree graphic.

## 2.6 Lesson 6 – Generating Fault Tree Cut sets

In Lesson 6, the user will learn how to generate and quantify fault tree cut sets, view fault tree cut set, and produce a report on fault tree cut set results. For this lesson, cut sets will be generated for the three fault trees (BRAKE-SYS, OP-MANEUVER, and LOAD) created in Lesson 3.

After the fault tree structure, or logic, has been developed and saved, fault tree cut sets can be produced and quantified. A cut set is a set of basic event combinations that, if they occur together, will cause the modeled top gate failure to occur. A minimal cut set has no other cut set as a subset. When generating fault tree cut sets, SAPHIRE reduces the and/or fault tree logic to minimal cut sets and uses the numerical rate values of the basic event data to calculate the probability of failure of the top event for the fault tree being quantified.

### 2.6.1 STEP 1: Generating Fault Tree Cut Sets

To generate new fault tree cut sets for the fault tree BRAKE-SYS, perform the following steps:

Select the BRAKE-SYS fault tree from the main fault tree list. Click the right mouse button to evoke the pop-up menu and move the highlight cursor down to Solve. This action will bring up the solve dialog (Figure 26). If the “Cut Set Truncation” box is used, then those cut sets with quantified value below the specified cutoff value will not be retained. Two fault tree truncation options are available, (1) Global, that uses the same truncation for all fault trees or (2) System-specific, that uses a fault-tree specific truncation level. Click the Solve button to start the solve process.

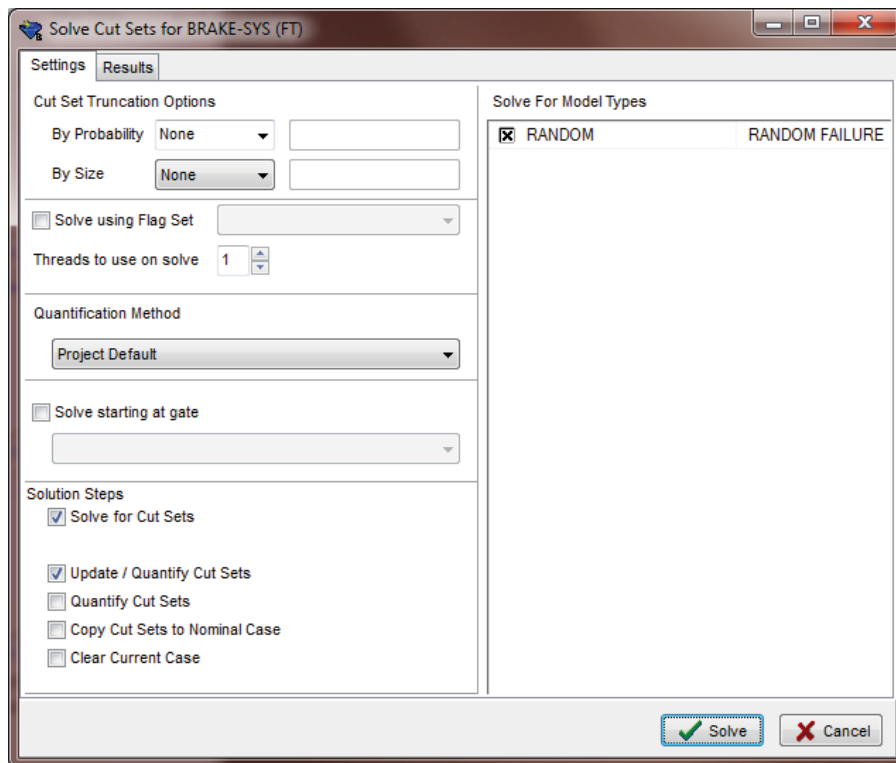


Figure 26. Solve fault tree dialog

### QUICK TIP

To highlight all fault trees in the Fault Tree List (to generate cut sets for all the listed fault trees), select a one of the fault trees and press CTRL-A.

The other options shown on the Fault Tree solve dialog are discussed in Volume 3 of this NUREG.

The Cut Set Generation Results screen will be displayed as shown in Figure 27. This screen shows the number of cut sets generated (for each fault tree), the minimal cut set upper bound approximation (under the Value column) result, and the time it took to solve the fault tree(s). Click on the Close button to close the dialog, or the Cut Sets button to view the minimal cut sets. Note that solving the top gate of a fault tree will automatically expand and use all the transfers (sub-trees) during the analysis.

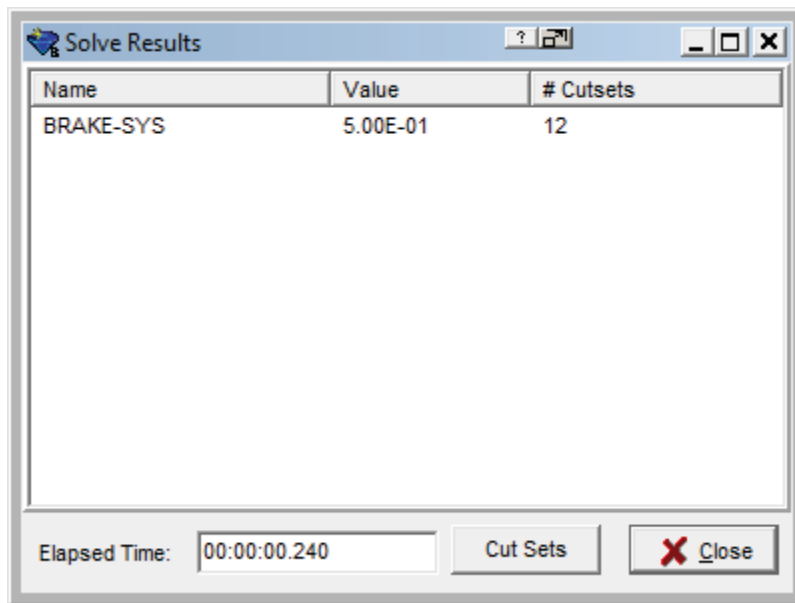


Figure 27. Cut set solve process dialog

## 2.6.2 STEP 2: Displaying Fault Tree Cut Sets

To display the new fault tree cut sets for the fault tree BRAKE-SYS, perform the following steps:

From the Fault Tree List screen, highlight the BRAKE-SYS fault tree. **Click the right mouse button** to evoke the pop-up menu and select View Cut Sets option. The cut sets generated for fault tree, BRAKE-SYS are now displayed as shown below in Figure 28. From this screen,

details of each cut set and the minimal cut set upper bound approximation can be observed by clicking the plus [+] node to the left of each cut set. The details for all cut sets can be seen by clicking the Expand All button.

The cut sets can be saved to an end state by clicking the Save to End State button then specifying a name for the end state. The select box in the upper-right corner of the cut set screen can be used to see the current, nominal, or both sets of cut sets. Select the Close button when complete.

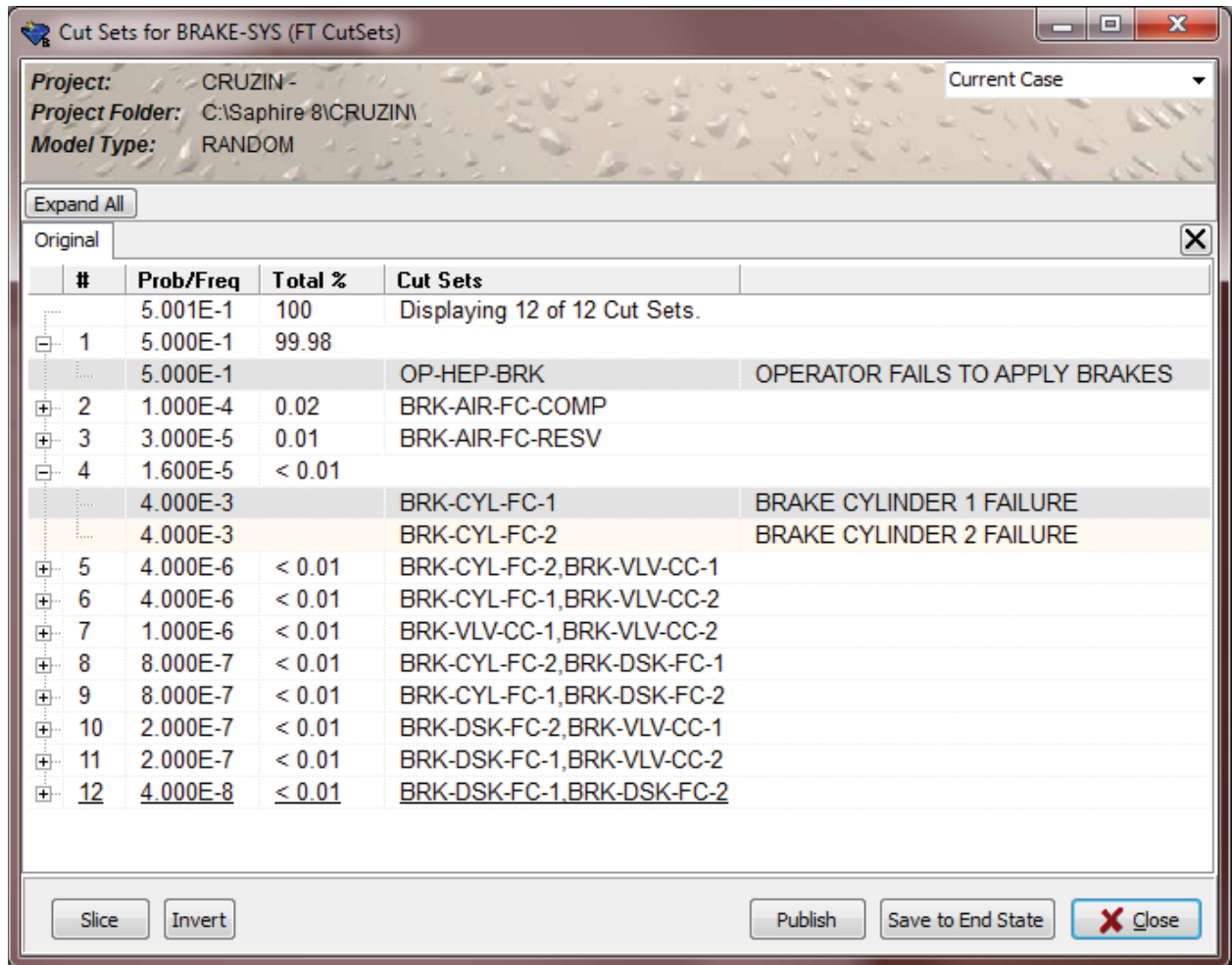
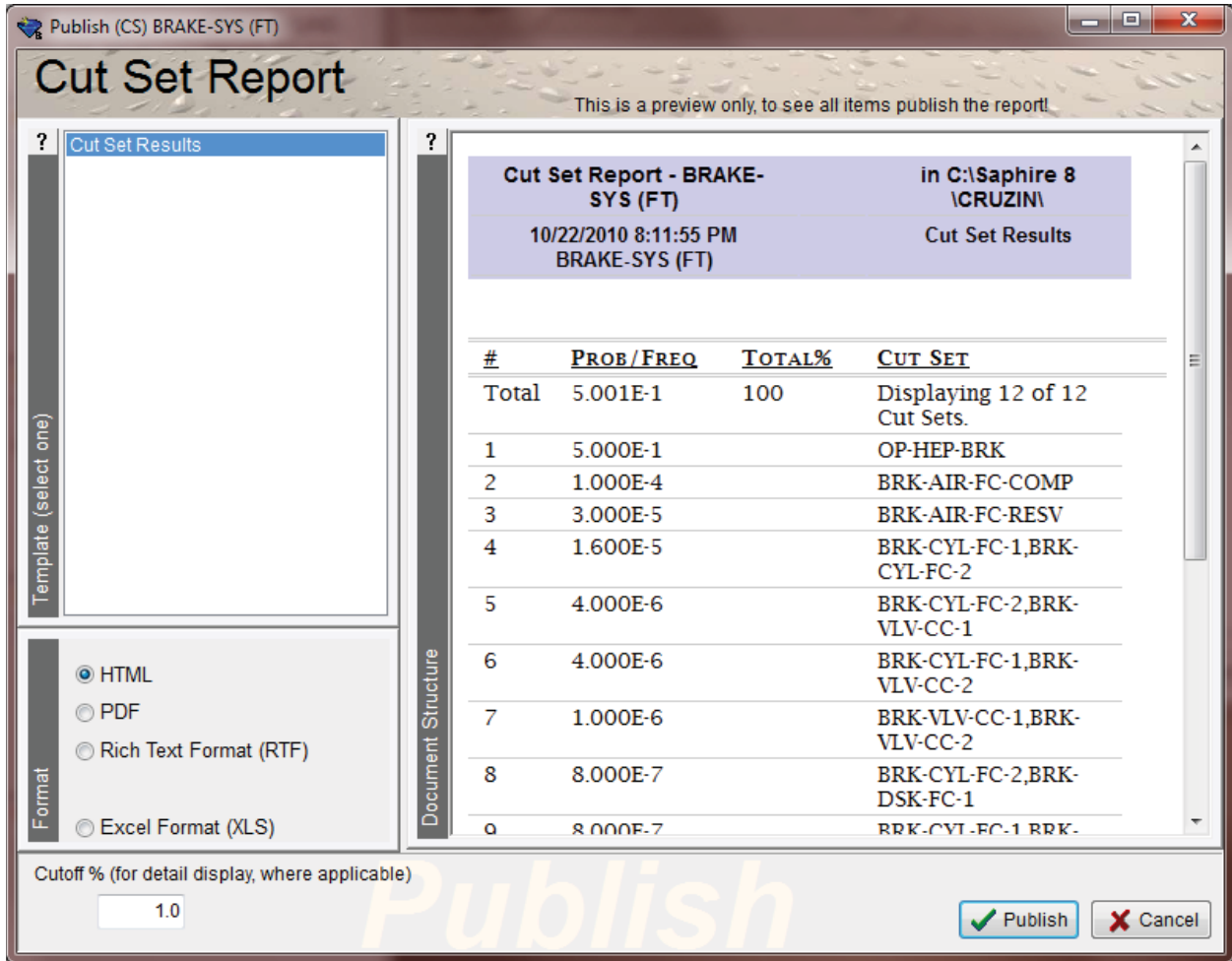


Figure 28. Viewing fault tree cut sets screen

### 2.6.3 STEP 3: Reporting Fault Tree Cut Sets Results

To generate report results for the new fault tree cut sets for the fault tree BRAKE-SYS, from the Cut Set Viewer screen shown in Figure 28, click on the Publish button. The results screen is displayed as shown below.



Click on the Publish button to send the report to a specified output file. Click on the Cancel button to complete the report function.

An additional option to obtain a cut set report is through the Publish feature of SAPHIRE. To obtain a fault tree cut set report through the Publish feature, first highlight the fault tree(s) then select the Publish → Fault Tree Report option from the main menu. The screen shown in Figure 29 will be displayed.

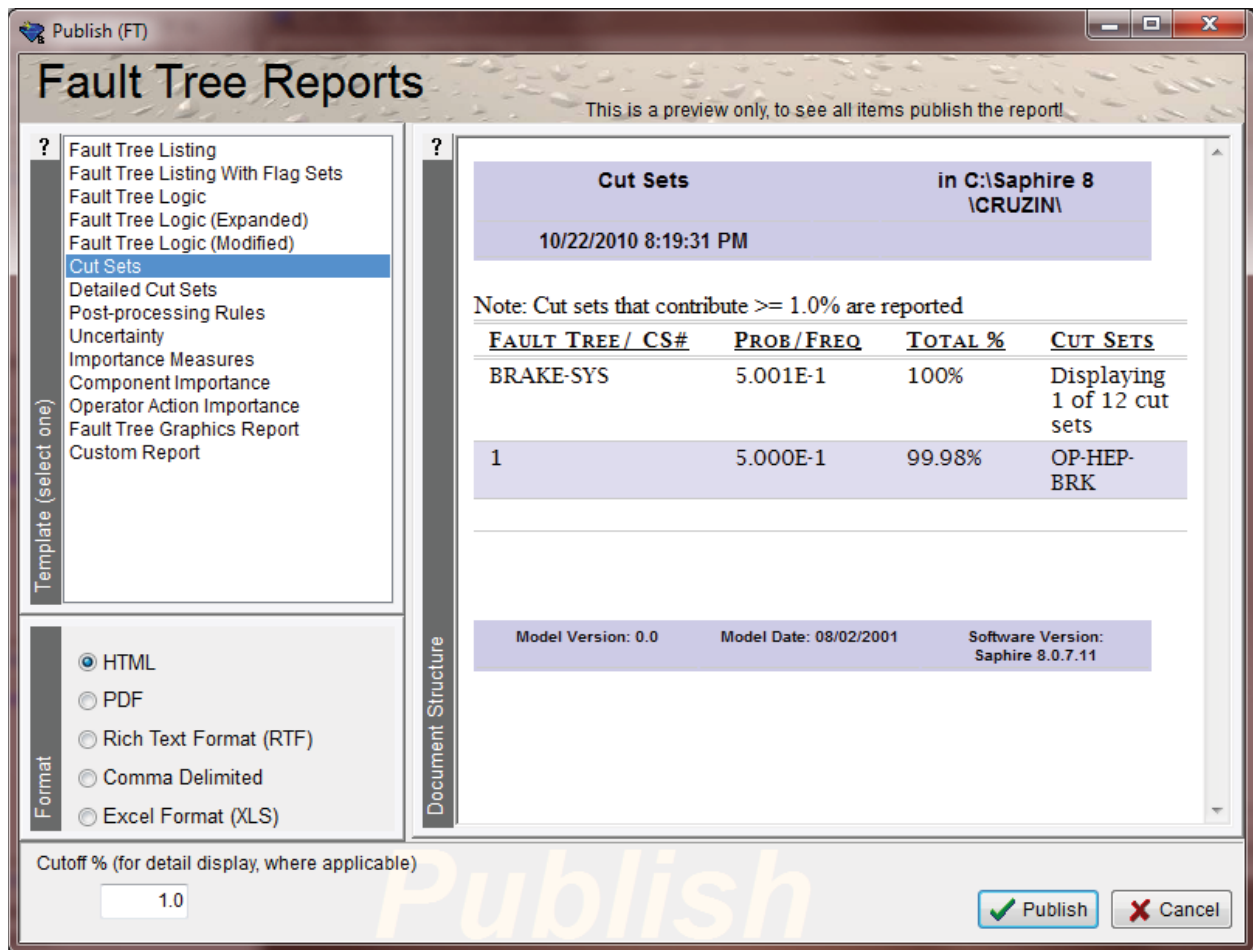


Figure 29. Fault tree cut sets report using the publish option

## 2.7 Lesson 7 – Fault Tree Uncertainty Analysis

In Lesson 7, the user will learn how to perform an uncertainty analysis in SAPHIRE for fault trees. Uncertainty analysis calculates the variability of a fault tree top event probability resulting from uncertainties in the basic event probabilities. SAPHIRE provides two uncertainty analysis techniques:

Simple Monte Carlo Sampling. This technique makes repeated quantifications of the system cut sets using each random variable sampled from the basic event uncertainty distributions.



Latin Hypercube Sampling. This is a stratified sampling technique, with the random variable distributions divided into equal probability intervals.

SAPHIRE supports the following basic event distribution types:

Distribution	Code	Parameter
Blank		(Use point estimate value only)
Lognormal	L	Error factor
Normal	N	Standard deviation
Beta	B	B in Beta(a, b)
Dirichlet	D	Beta value of success branch
Gamma	G	r in $\Gamma(r)$
Chi-squared	C	Degrees of freedom
Exponential	E	(none)
Uniform	U	Upper end point
Histograms	H	Histogram number
Maximum Entropy	M	Lower and upper end point
Seismic	S	Beta r, Beta u
Constrained Noninformative	O	(none)
Triangular	T	Upper and lower endpoints

For the OBSTRUCT project, the basic events only implemented the “Lognormal” distribution. For additional information on the uncertainty analysis capabilities of SAPHIRE, please refer to the SAPHIRE Technical Reference Guide.

For this example, the fault tree BRAKE-SYS will be used to perform an uncertainty analysis. To perform a fault tree uncertainty analysis, perform the following steps:

1. Highlight the applicable fault trees from the main object list.
2. Click the right mouse button to evoke the popup menu.

3. Move the mouse highlight cursor to View Uncertainty and click the left mouse button.

If more than one fault tree is selected, the uncertainty will be calculated for each fault tree highlighted. The “Uncertainty for:” dialog will be displayed (Figure 30).

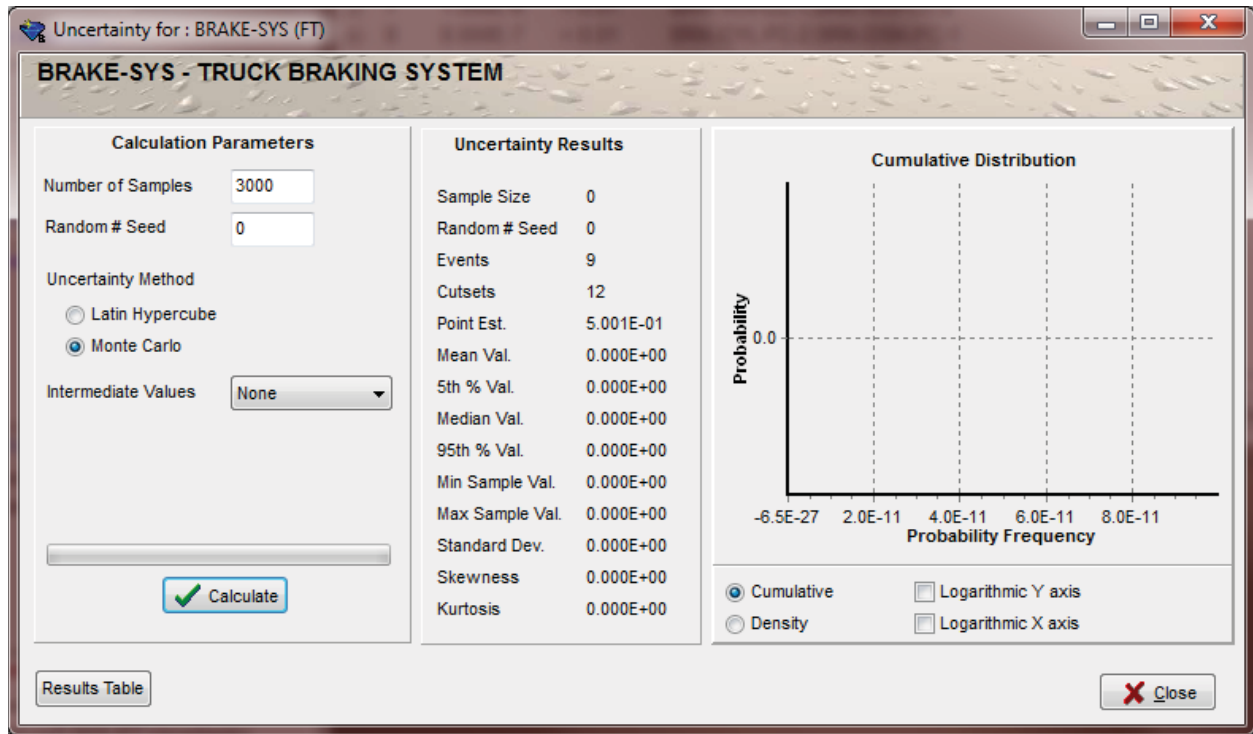


Figure 30. Fault tree uncertainty analysis dialog

Options to modify include:

- Number of samples (the larger the number, the more accurate the results) = 5000
- Seed for random number generator = 54321
- Uncertainty method = Monte Carlo (this is the default method)
- Intermediate Values = None

Click the Calculate button when all the values are entered.

Uncertainty results will be displayed on the dialog screen following the uncertainty calculation on the right part of the dialog (as seen in Figure 31).

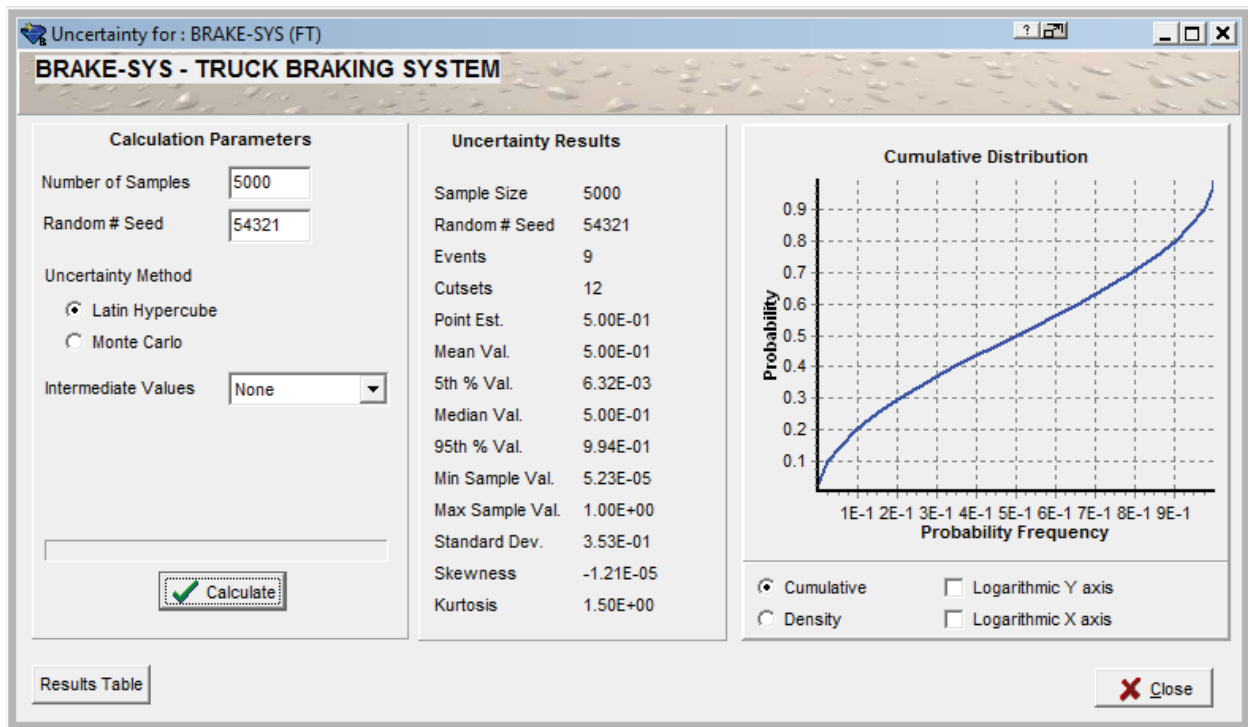


Figure 31. Fault tree uncertainty analysis results

## 2.8 Lesson 8 – Generating Event Tree Cut Sets

In Lesson 8, the user will learn how to generate and quantify event tree (or sequence) cut sets, view sequence cut sets, and produce a report on the sequence cut set results. For this lesson, cut sets will be generated for the event tree, OBSTRUCT created in Lesson 2.

After the event tree structure and end states have been developed and saved, event tree sequences can be generated by “linking” the event tree. Once the event tree(s) are linked and sequences generated, cut sets can then be produced and quantified for each sequence.

This lesson will guide the user over four aspects to developing and analyzing event tree or sequence cut sets:

- Linking the event tree.
- Solve for sequence cut sets.
- Performing sequence uncertainty analysis.
- Reporting sequence cut set results.

### **2.8.1 STEP 1: Linking Event Tree Sequences**

The term “linking” event trees is the process of generating sequences logic using the event tree graphical files and incorporating the top events of the event tree with the respected fault tree logic in the event tree sequences.

SAPHIRE 8 automatically links event trees when the event tree is saved from the graphical editor. To re-link an event tree (for example, after linking rules are modified) and generate sequence logic, perform the following steps:

Select the Event Tree from the main event tree list.

**Right-click** to invoke the pop-up menu. Select the Link option.

### **2.8.2 STEP 2: Solve For Sequence Cut Sets**

Once linking has been completed and sequence logic was created for the OBSTRUCT event tree, sequence cut sets can now be generated. To generate sequence cut sets for the event tree OBSTRUCT, perform the following steps:

Select the Event Tree(s) from the main event tree list.

Right click to invoke the pop-up menu and select the Solve option.

The Cut Set Solve window (Figure 32) will be displayed. If the “Cut Set Truncation” option is used, then those cut sets with quantified value below the specified cutoff value will not be retained. Click the Solve button to accept the default cut set generation and truncation settings and to start the analysis.

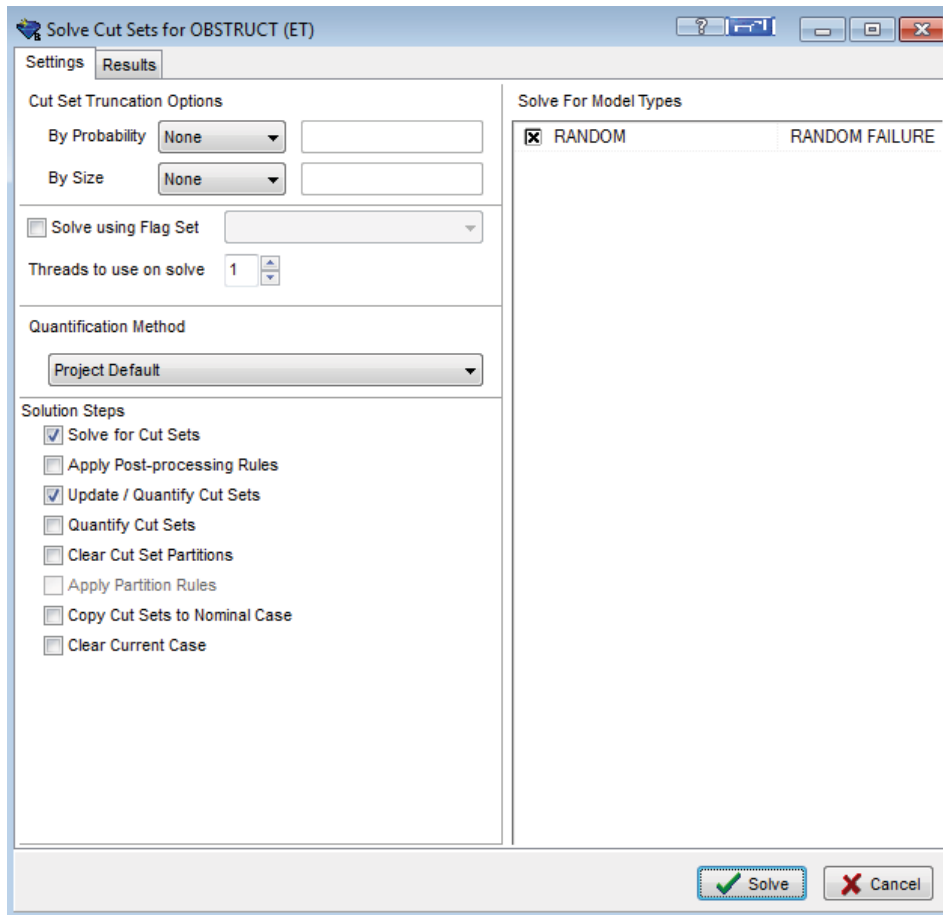


Figure 32. Event tree cut set solve dialog

After the sequence cut sets have been generated, SAPHIRE displays a summary screen (Figure 33). Note that this screen is the “Results” tab from the Solve dialog.

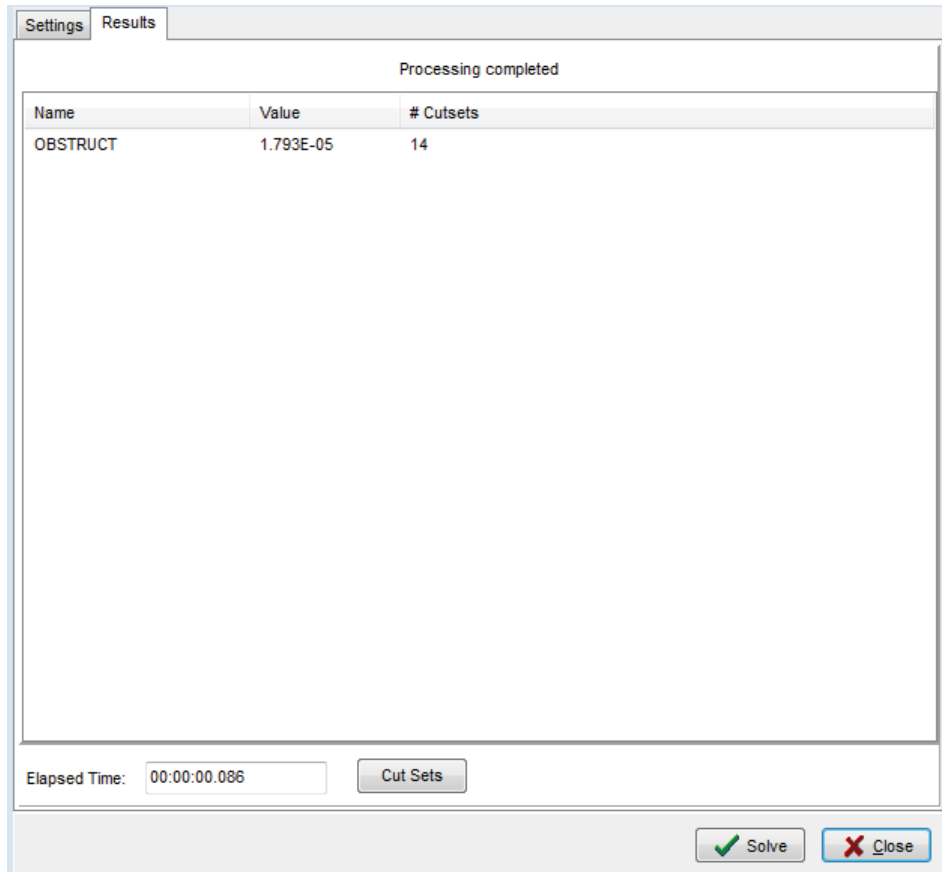


Figure 33. Event tree cut set summary for solving dialog

To display the event tree cut sets, either:

1. Click the Cut Sets button as shown above
2. Select the event tree (on the main window) then right click. Select the View Cut Sets option.

The cut sets generated for event tree OBSTRUCT are now displayed as shown in Figure 34. From this screen, details of each cut set and the minimal cut set upper bound approximation can be observed by clicking the plus [+] node to the left of each cut set. The details for all cut sets can be seen by clicking the Expand All button.

Select the Close button when complete.

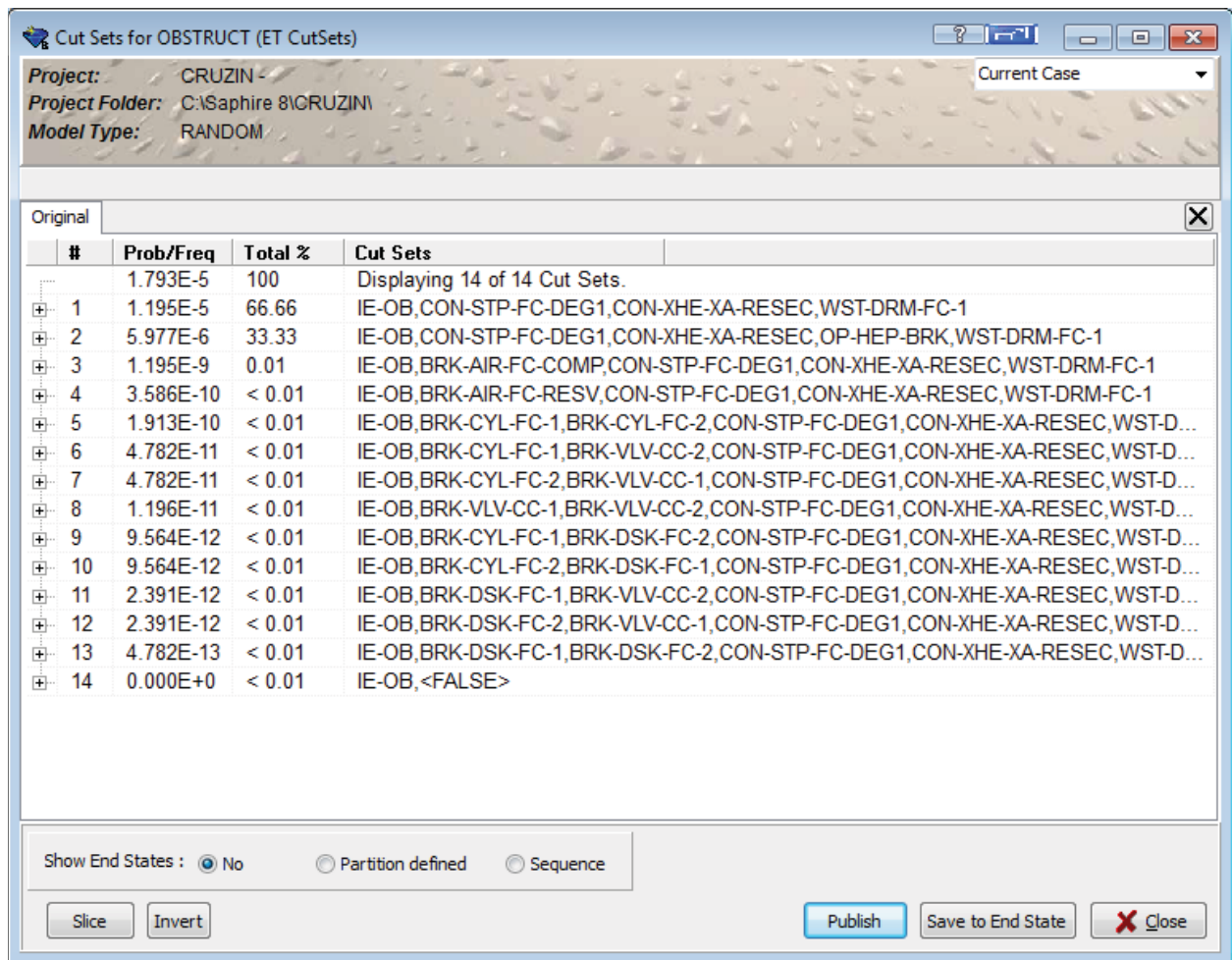


Figure 34. Event tree cut set dialog

### 2.8.3 STEP 3: Performing Sequence Uncertainty Analysis

The uncertainty analysis for sequence cut sets is very similar to that described for fault trees in Lesson 7. See Lesson 7 for a review of SAPHIRE's uncertainty capabilities. For this example, sequence #6 will have an uncertainty analysis performed.

To perform an uncertainty analysis on sequence #6, perform the following steps:

Click the plus [+] node to the left of the event tree OBSTRUCT to expand the list of sequences for that event tree as shown in Figure 35. Then highlight the sequence #6.

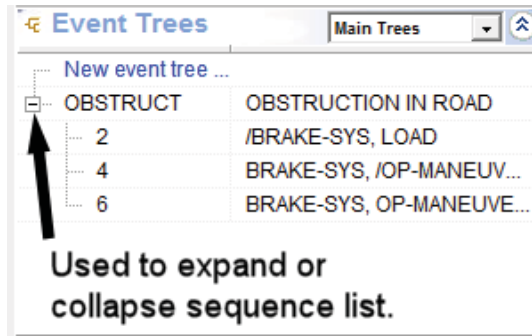


Figure 35. Event tree sequence list

Right click to invoke the pop-up menu. Select View Uncertainty as shown in Figure 36.

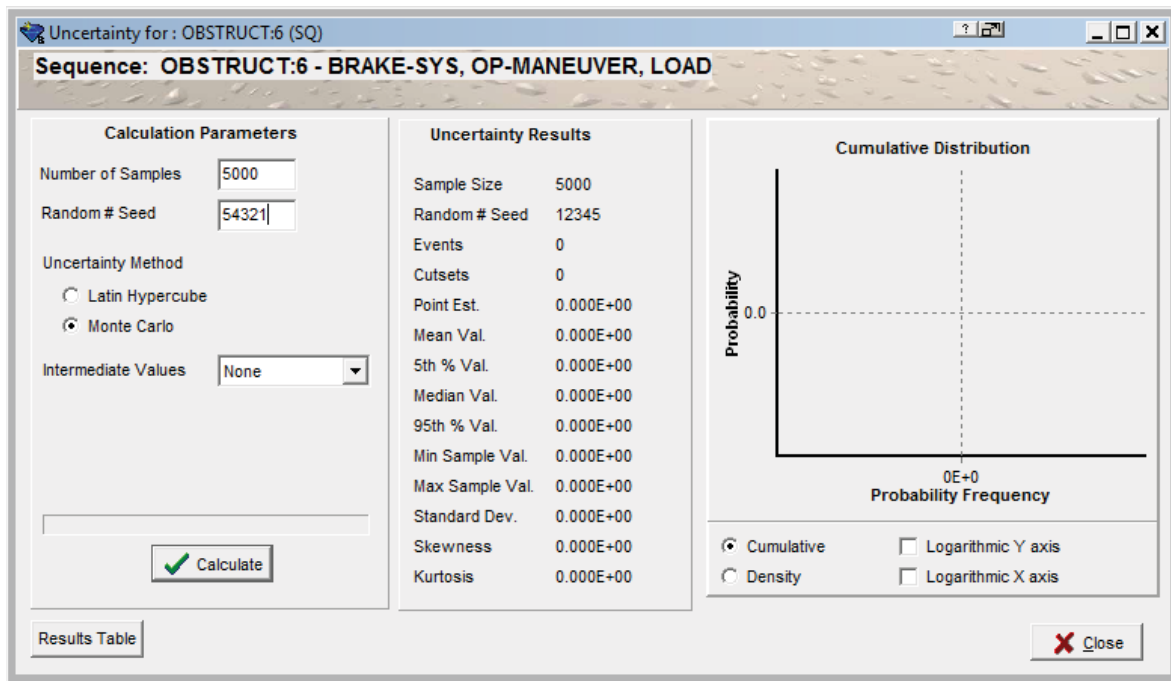


Figure 36. Event tree sequence uncertainty analysis screen

Enter the uncertainty calculation valves given below.

- Number of samples = 5000
- Seed for random number generator = 54321
- Uncertainty method = Monte Carlo



Click the Calculate button to start the uncertainty analysis for sequence #6. Uncertainty results will be displayed on the dialog screen following the uncertainty calculation on the right part of the dialog (as seen in Figure 37).

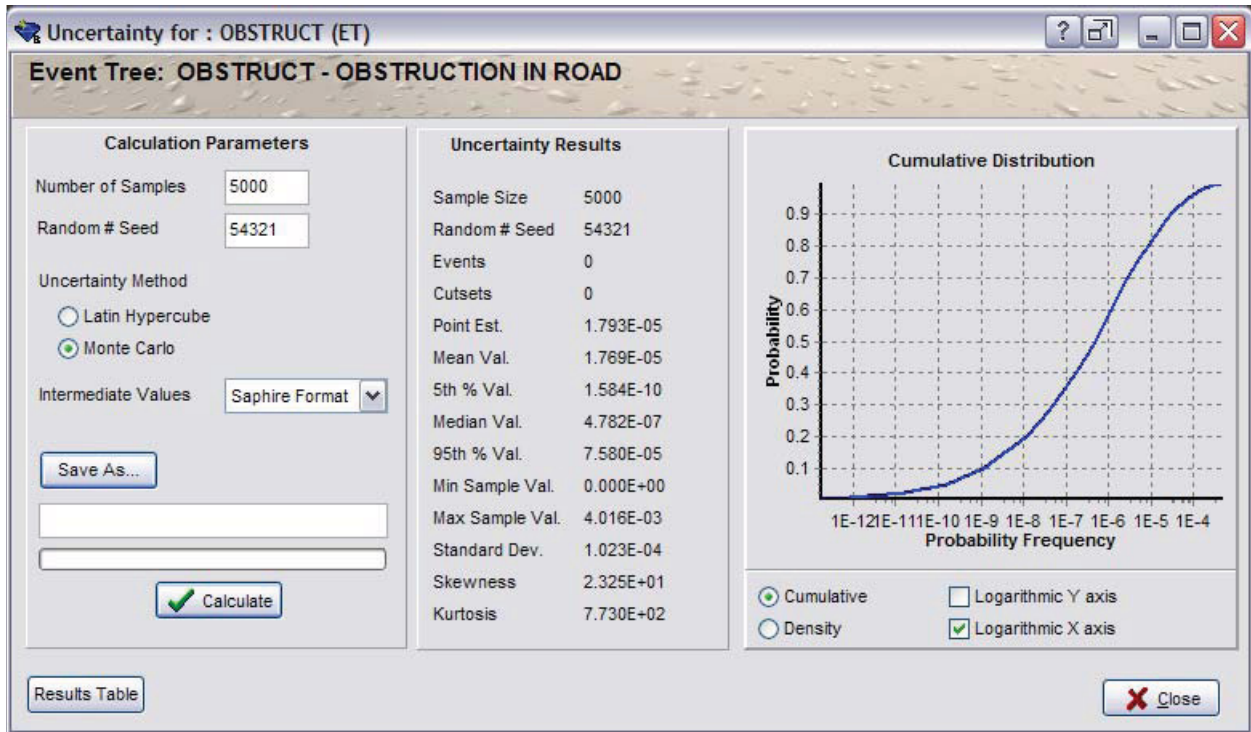


Figure 37. Event tree sequence uncertainty analysis results

Click the Close button to continue.

#### 2.8.4 STEP 4: Reporting Sequence Cut Set Results

To produce report results for the new sequence cut sets for the OBSTRUCT event tree, from the Cut Set Viewer screen shown in Figure 34, click on the Publish button. The results are displayed as shown below.

Cut Set Report -  
OBSTRUCT (ET)

Cut Set Results

#	Prob/ Freq	Total %	Cut Set
Total	1.79E-5	100	Displaying 14 of 14 Cut Sets.
1	1.19E-5	66.66	IE-OB,CON-STP-FC-DEG1,CON-XHE-XA-RESEC,WST-DRM-FC-1
2	5.97E-6	33.33	IE-OB,CON-STP-FC-DEG1,CON-XHE-XA-RESEC,OP-HEP-BRK,WST-DRM-FC-1
3	1.19E-9	0.01	IE-OB,BRK-AIR-FC-COMP,CON-STP-FC-DEG1,CON-XHE-XA-RESEC,WST-DRM-FC-1
4	3.58E-10	< 0.01	IE-OB,BRK-AIR-FC-RESV,CON-STP-FC-DEG1,CON-XHE-XA-RESEC,WST-DRM-FC-1
5	1.91E-10	< 0.01	IE-OB,BRK-CYL-FC-1,BRK-CYL-FC-2,CON-STP-FC-DEG1,CON-XHE-XA-RESEC,WST-DRM-FC-1
6	4.78E-11	< 0.01	IE-OB,BRK-CYL-FC-1,BRK-VLV-CC-2,CON-STP-FC-DEG1,CON-XHE-XA-RESEC,WST-DRM-FC-1
7	4.78E-11	< 0.01	IE-OB,BRK-CYL-FC-2,BRK-VLV-CC-1,CON-STP-FC-DEG1,CON-XHE-XA-RESEC,WST-DRM-FC-1
8	1.19E-11	< 0.01	IE-OB,BRK-VLV-CC-1,BRK-VLV-CC-2,CON-STP-FC-DEG1,CON-XHE-XA-RESEC,WST-DRM-FC-1
9	9.56E-12	< 0.01	IE-OB,BRK-CYL-FC-1,BRK-DSK-FC-2,CON-STP-FC-DEG1,CON-XHE-XA-RESEC,WST-DRM-FC-1
10	9.56E-12	< 0.01	IE-OB,BRK-CYL-FC-2,BRK-DSK-FC-1,CON-STP-FC-DEG1,CON-XHE-XA-RESEC,WST-DRM-FC-1
11	2.39E-12	< 0.01	IE-OB,BRK-DSK-FC-1,BRK-VLV-CC-2,CON-STP-FC-DEG1,CON-XHE-XA-RESEC,WST-DRM-FC-1
12	2.39E-12	< 0.01	IE-OB,BRK-DSK-FC-2,BRK-VLV-CC-1,CON-STP-FC-DEG1,CON-XHE-XA-RESEC,WST-DRM-FC-1
13	4.78E-13	< 0.01	IE-OB,BRK-DSK-FC-1,BRK-DSK-FC-2,CON-STP-FC-DEG1,CON-XHE-XA-RESEC,WST-DRM-FC-1

Click on the Print button to send the report to a printer or click on the Save as... button and type in a desired file name to save the report to a file. Click on the Close button to complete the report function.

An additional option to obtain a cut set report is through the Publish feature of SAPHIRE. To obtain an event tree cut set report through the Publish feature, first highlight the event tree(s) or sequence(s) then select the Publish → Event Tree Report option from the main menu. The screen shown in Figure 38 will be displayed.

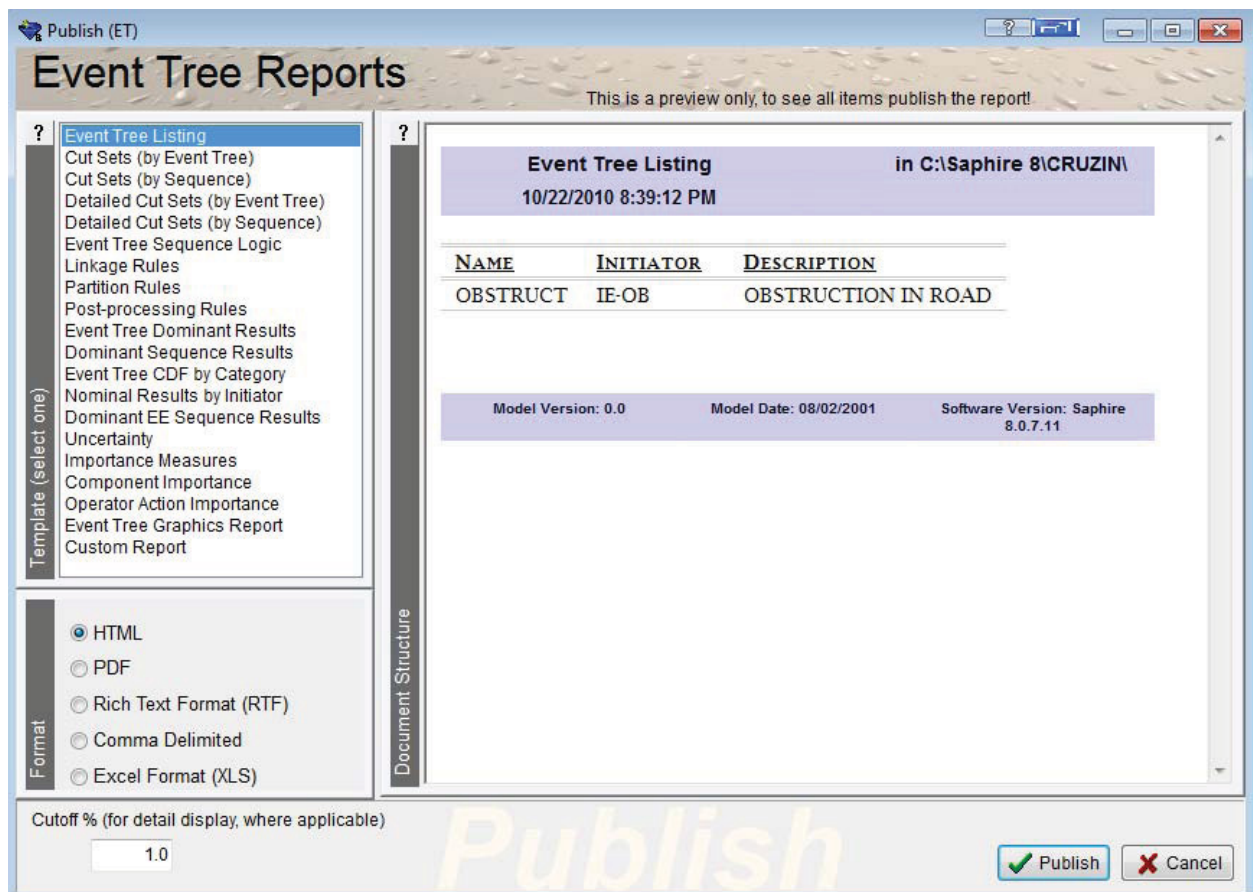


Figure 38. Event tree cut sets report using the publish option



## 3. ADVANCED LESSONS

The purpose of the Advanced Lessons section is to familiarize the user with some of advanced features and operation of SAPHIRE 8. The advanced features are not essential to perform a simple analysis in SAPHIRE. However, for more complicated studies or PRA projects, the lessons covered in this section will prove to be extremely powerful.

The lessons covered in the Advanced Section of this tutorial will expand on the techniques demonstrated in Section 2, Basic Lessons. The lessons will use the Project, fault trees, and the event tree created in Basic Lessons section.

The Advanced Lessons in this section includes the following:

### **Lesson 10. Creating Flag Sets**

This lesson describes flag sets and guides the user on how to implement them into SAPHIRE.

### **Lesson 11. The “Linking” Rule Editor**

Lesson 11 introduces the user to SAPHIRE “Link Event Tree” rule editor. The rule editor allows you create rules that affect the event tree linkage process and sequence generation.

### **Lesson 12. Post-processing Rules**

Lesson 12 introduces the user the “Post-processing Rule” editor of SAPHIRE. The post-processing rule editor allows you to create rules that affect existing cut sets in a “post-processing” manner.

## 3.1 Lesson 10 – Creating Flag Sets

In lesson 10, the user will learn about flag sets and why they are used in SAPHIRE. An example flag set will be created in the project PRACTICE and applied to the OBSTRUCT event tree.

Flag sets are a special type of setting that can only contain either house flag changes to the calculation type or process type changes. Flag sets are primarily used in SAPHIRE to manipulate logic gates by turning house events embedded in the logic on or off, depending on the logic requirements. Flag sets can only change the Calculation type to T (TRUE – i.e. guaranteed failure), F (FALSE – i.e. guaranteed success) or I (IGNORE). In addition, specific types of process flags can be change with a flag set.

### 3.1.1 Creating a Flag Set

Flag sets are created in the Project → Flag Set menu option and are applied to the database by directly “attaching” them to either fault trees or event tree sequences.

The following example will create a simple flag set called “FLAG1” that changes the basic event calculation type of the basic event OP-HEP-BRK to FALSE. The flag set will then be assigned to fault tree BRAKE-SYS and then re-quantified with the flag set.

### STEP 1: Creating and Naming a Flag Set

To create a flag set in SAPHIRE, perform the following steps:

Select the Project → Flag Set menu option. This action will bring up the screen shown in Figure 39



Figure 39. Flag set option

In the **Flag Set** screen, select **Add Flag Set** (from the drop-down list) and click Go. This action will bring up the flag set editor as shown in Figure 40.

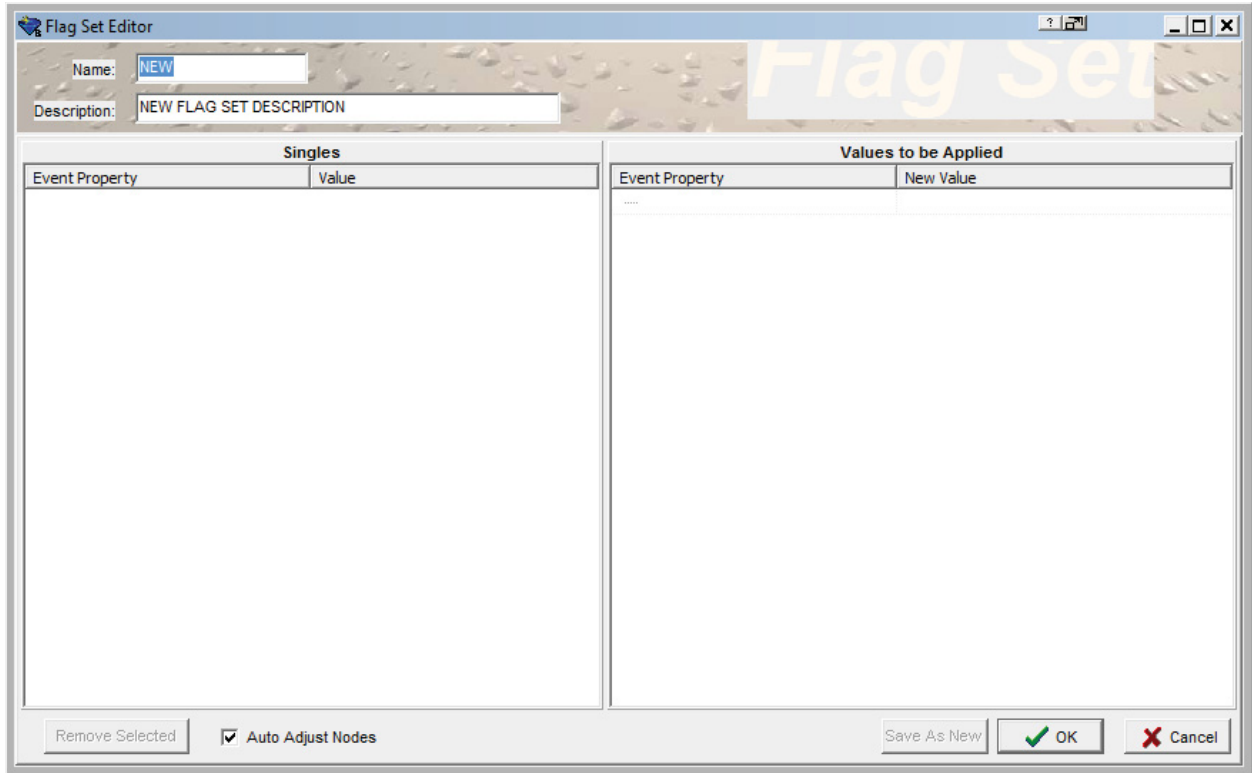


Figure 40. Flag set editor

In the Flag Set Editor screen, enter in the Name box FLAG1. A description is not required, but may be entered (up to 120 characters).

Now, drag the basic event OP-HEP-BRK from the basic event list to the “Singles” box on the left part of the Flag Set Editor, as illustrated in Figure 41.

## STEP 2: Modifying the Calculation Type in the Flag Set

To modify the basic event Calculation Type in the flag set, perform the following steps:

In the **Flag Set Editing** screen (after dragging and dropping the OP-HEP-BRK basic event), the new settings that will be applicable are shown on the right-hand side of the screen (see Figure 42). In this example, the Failure Model needs to be changed from “Failure Probability (1)” to “House event false with probability = 0 (F).” Click the OK button once this change is modified.

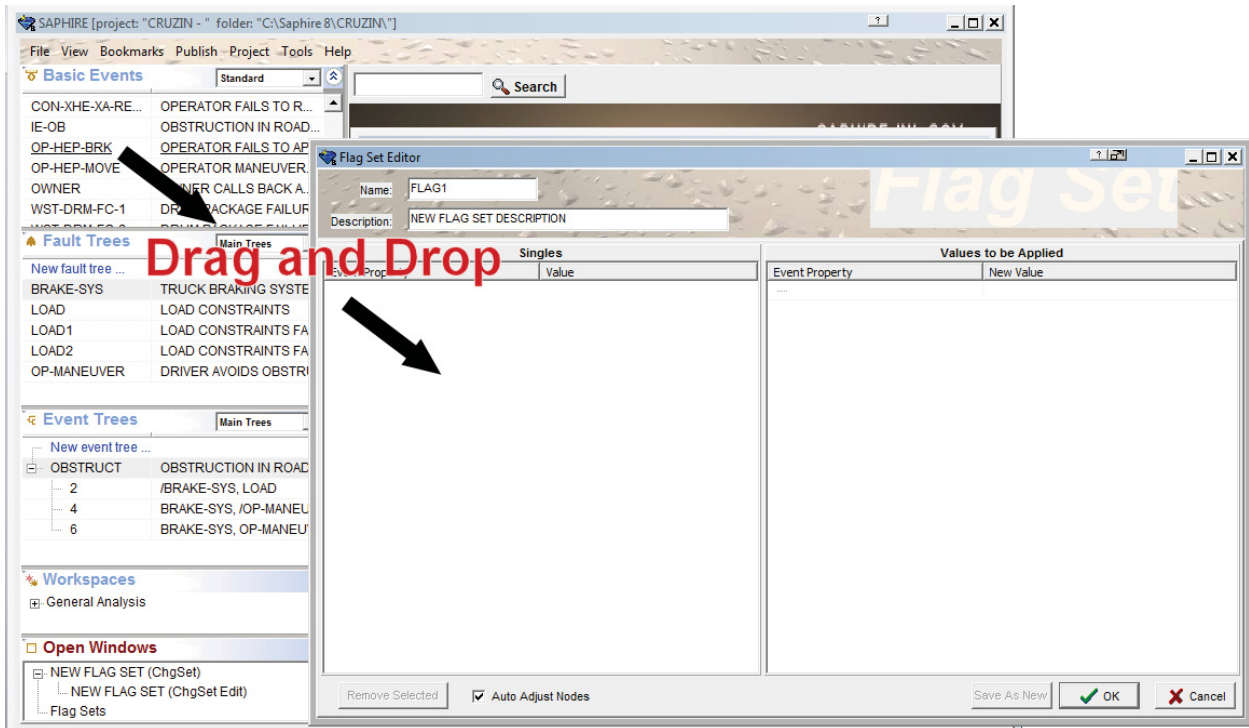


Figure 41. Adding a new event to the FLAG1 flag set

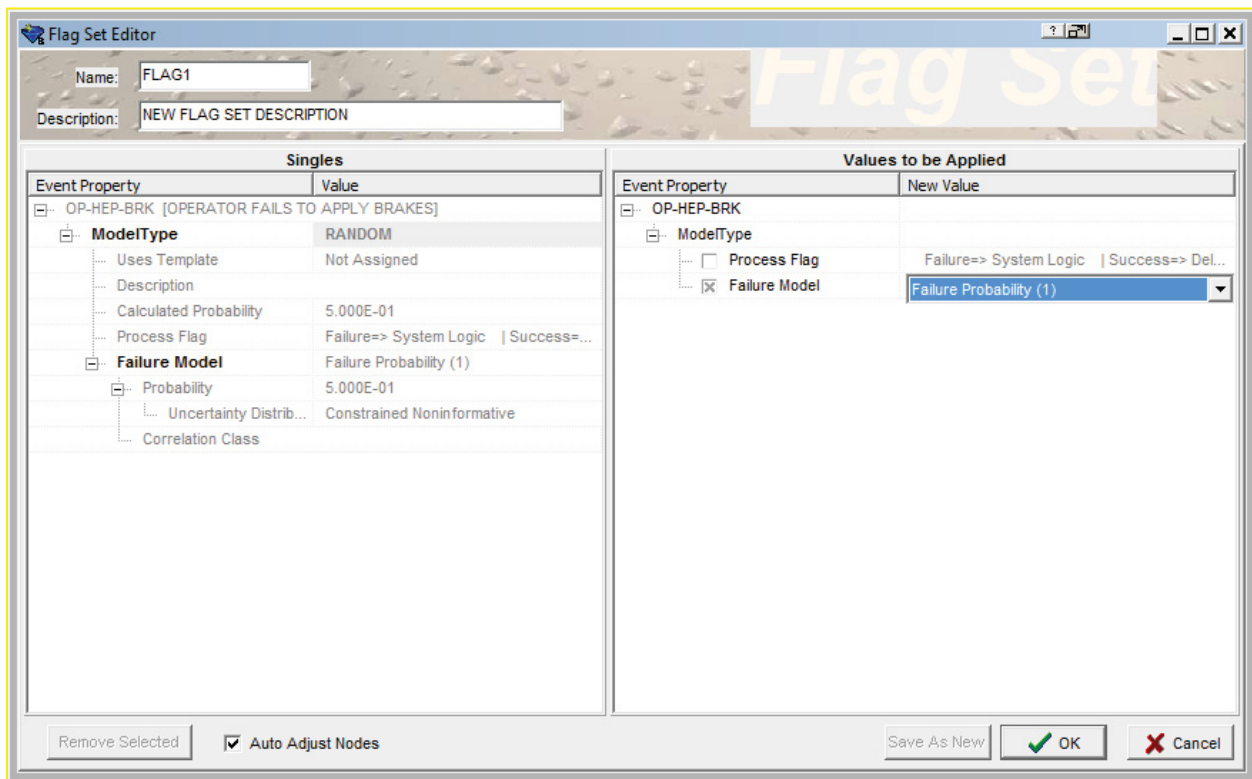


Figure 42. Modifying an event in the FLAG1 flag set



The Flag Set, FLAG1 is now ready to use in the database. Click on the Close button to return to main SAPHIRE menu screen.

### STEP 3: Using the Flag Set

To use a flag set after it has been created, you must assign it to a sequence or a fault tree. The following example assigns the Flag Set, FLAG1 to the fault tree BRAKE-SYS. To assign a flag set to a fault tree, perform the following steps:

1. Select the BRAKE-SYS fault tree from the main fault tree list.
2. Click the right mouse button to evoke the pop-up menu and move the highlight cursor down to Edit Properties. This action will bring up the fault tree properties dialog (Figure 43). To apply the Flag Set, select the Default Flag Set ("FLAG1").

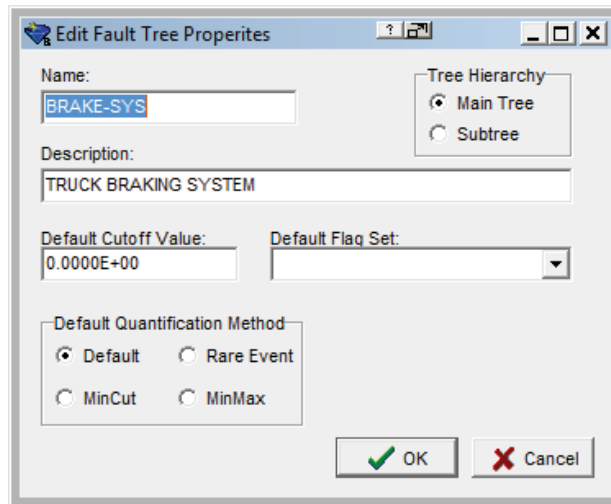


Figure 43. Modifying the flag set assigned to a fault tree

Click the Ok button when complete. The Flag Set is now assigned to the fault tree BRAKE-SYS and will be applied on the fault tree permanently until it is removed from the "Default Flag Set" box. To solve the fault tree, BRAKE-SYS with the flag set applied, select the BRAKE-SYS fault tree and right click, select Solve. The new results with the operator action set to FALSE is shown in Figure 44.

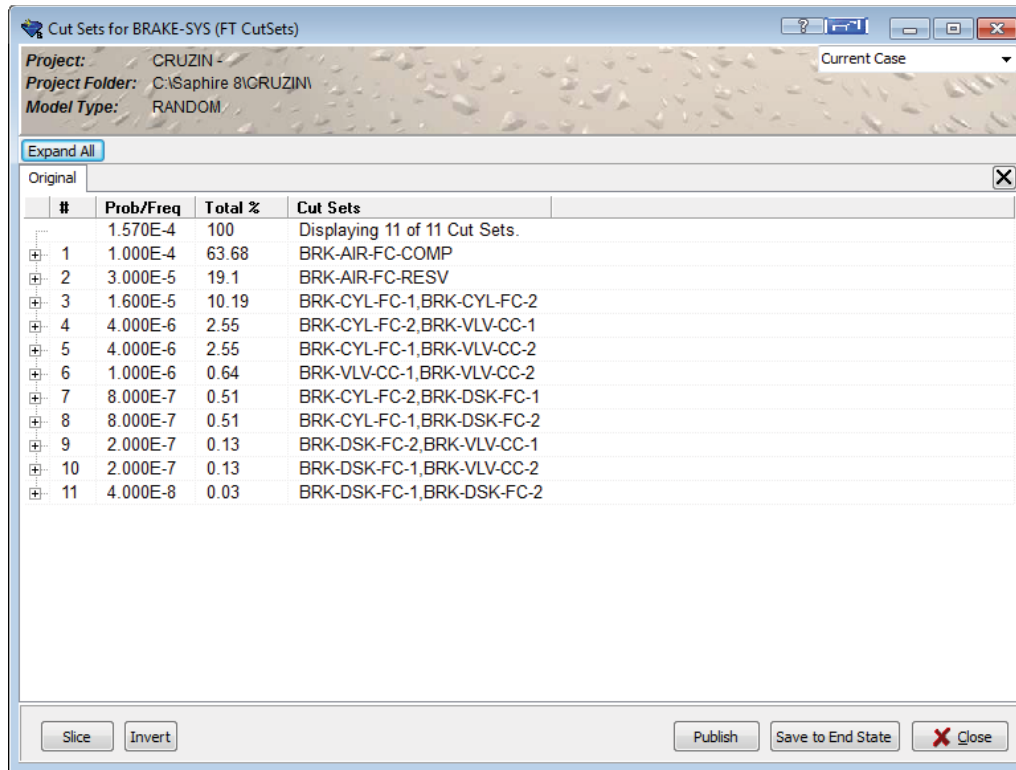


Figure 44. Fault tree cut sets with the FLAG1 flag set assigned to it

#### STEP 4: Removing the Flag Set

To remove the flag set from the fault tree, BRAKE-SYS:

1. Select the BRAKE-SYS fault tree from the main fault tree list.
2. Click the right mouse button to evoke the pop-up menu and move the highlight cursor down to Edit Properties. This action will bring up the fault tree properties dialog (as shown in Figure 16).

To remove the Flag Set, clear the Default Flag Set field so that it is blank. Click Ok.

### 3.2 Lesson 11 – The “Linking” Rule Editor

In lesson 11, the user will be introduced to the linking rule editor. This editor allows rules to be written that are used when sequence logic is generated. Without using linking rules, the sequence logic consists of the logical combinations of failed and successful top events. With these linking rules, the user is allowed to replace the top event with a substituted top event based on the logical conditions dictated by the rule. These rules also allow the user to assign

flag sets (see the previous section) to sequences also based on the logical conditions dictated by the rule.

The linking rule editor uses standard programming (i.e. FORTRAN, BASIC) line format and structure. Standard common symbols include:

```
|      Denotes a comment line
~      Logical operator for "not present"
*      Logical AND operator (i.e. A AND B occur)
+      Logical OR operator (i.e. A OR B occurs)
/      Complement
( )    Parentheses for grouping terms
```

Once linking rules have been written, the rule searches the event tree logic for the specific search criteria specified by the user in the rule. Once the search criteria are met, the rule replaces the default top event (i.e. original top event) by a new top event that is previously specified. Typical search criteria examples used by the linking rule editor are shown in Table 2.

The linking rule editor is implemented when the "Link Event Trees" operation is being performed and the sequence logic is being created. If the linking rules have been written, the rules searches the event tree logic for the search criteria specified in the rule and replaces the original top event in the event tree with a new (substituted) top event. For example, the following linking rule would replace top event C with a new top event C-SYS if the search criteria A \* B is met (i.e., if A and B are both failed):

```
if A * B then
/C = C-SYS;
C = C-SYS;
endif
```

**Table 2. SAPHIRE linking rule search criteria examples.**

<b>Search Criteria</b>	<b>Example</b>	<b>Meaning of Search Criteria</b>
A	<b>If A then</b>	If “A” occurs then do this...
/A	<b>If /A then</b>	If “A” does not occur then do this...
~A	<b>If ~A then</b>	If “A” is not present then do this...
A * B	<b>If A * B then</b>	If “A” and “B” occurs then do this...
A + B	<b>If A + B then</b>	If “A” or “B” occurs then do this...
(A + B)* C	<b>If (A + B) * C then</b>	If “A” or “B” occurs and C occurs then do this...
always	<b>If always then</b>	The search criteria are always met, so do this...

The /C = C-SYS line replaces the success branch of the event tree while the C = C-SYS line replaces the failure branch. In the rule editor, for /C = C-SYS, the complement forward slash for /C-SYS is implied and not required (a complemented top event is automatically replaced by a new complemented top event). In addition, **it is required by the rule editor that any replacement line must end with a semicolon (;).**

There is no limit to the number of replacement lines that can be used in a rule.

More detailed information on rule structure, keywords, and nomenclature can be found in the Technical Reference Manual (Volume 2 of this NUREG).

This lesson will guide the user on adding a substitution fault tree (i.e. top event) to the event tree OBSTRUCT created in Lesson 2. The substitution will require the user to enter a linking rule into SAPHIRE and then re-link the event tree with the compiled linking rule.

### **3.2.1 STEP 1: Creating the New Substitution Fault Tree**

For a new fault tree to substitute for the existing fault tree, the logic for new fault tree must first be created. Refer to Lesson 2 for instructions on creating a fault tree in SAPHIRE. Figure 45 illustrates the new fault tree (LOAD1) that will be substituted in the event tree OBSTRUCT. Create the LOAD1 fault tree in SAPHIRE as indicated in the earlier fault tree Section 2.3.2.

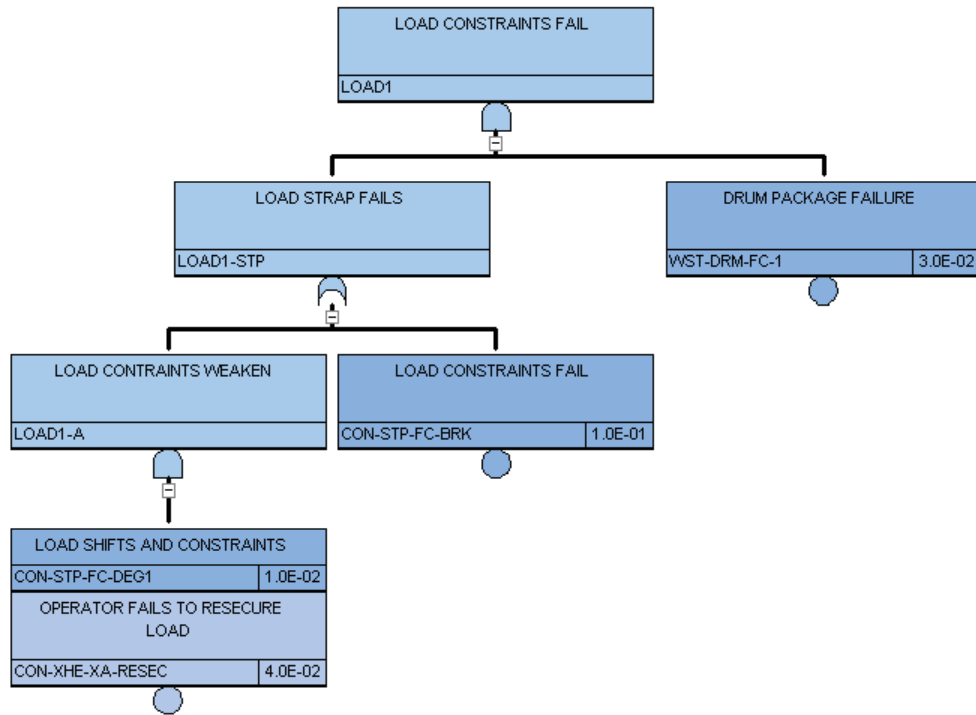


Figure 45. LOAD1 substitution fault tree

With the creation of the new fault tree, LOAD1, there was one basic event added to the database (CON-STP-FC-BRK). The failure probability for this basic event is 1.0E-1 with an error factor of 10. See Lesson 4 for detailed information on modifying basic event data.

### 3.2.2 STEP 2: Creating the Linking Rule

The new fault tree, LOAD1 has logic that is designed to give a slightly higher probability, due to a specific event tree scenario. It was designed for Sequence 4 of the event tree, OBSTRUCT where the truck brake system has failed but the driver has succeeded to avoid the obstruction in the road. Therefore, to be applied to sequence 4, a linking rule needs to be written that substitutes the LOAD top event with LOAD1 when BRAKE-SYS has failed but OP-MANEUVER is successful.

Linking rules are entered into SAPHIRE through the Linkage Rule editor. To access this editor, highlight the applicable event tree (OBSTRUCT) and then click the right mouse button to evoke the pop-up menu. Move the highlight cursor down to Edit Linkage Rules. This action will bring up the event tree linkage rules editor (Figure 46).

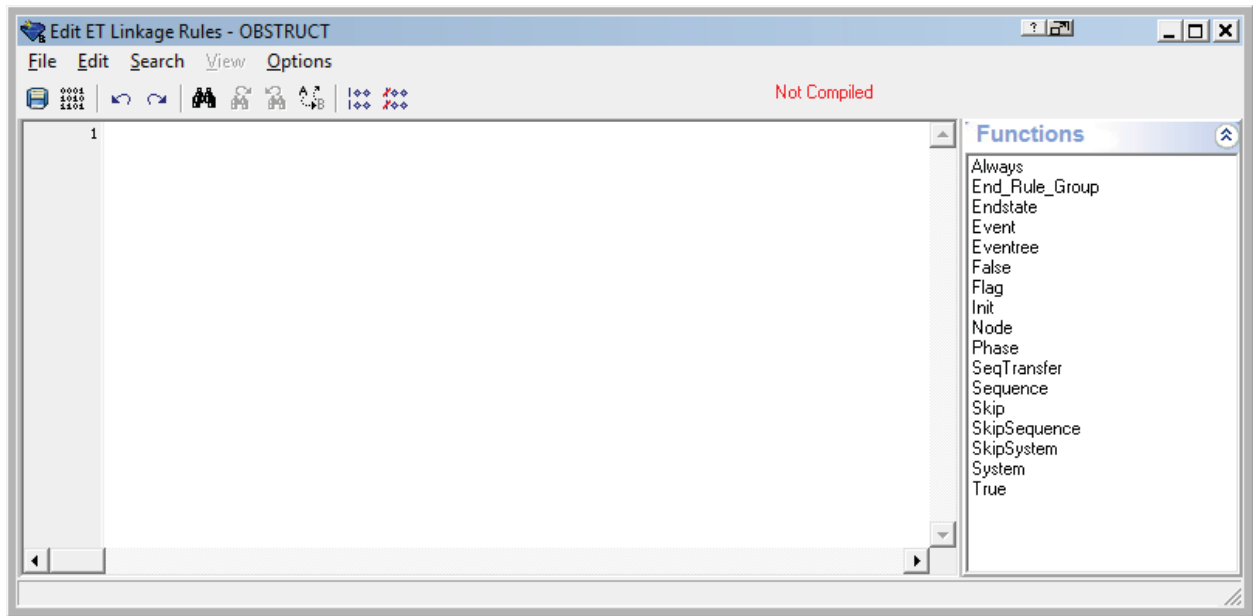



Figure 46. Event tree linkage rule editor (without a rule)

To write a rule that accomplishes the desired substitution, enter into the linkage rule editor the following rule:

```
| Rule substitutes top event LOAD1 for LOAD
| if BRAKE-SYS fails and OP-MANEUVER succeed.
|
if BRAKE-SYS * /OP-MANEUVER then
  /LOAD = LOAD1;
  LOAD = LOAD1;
endif
```

Note: The **Linkage Rule Editor** is drag-and-drop aware, which means that both basic events and fault tree objects may be dragged to the editing window.

To exit the linkage rule editor, click on the Exit button  and click on Yes to save the linkage rule (or click File → Save).

### 3.2.3 STEP 3: Re-Linking the Event Tree Sequence

In order for the substitution rule to take place, the event tree must be re-linked and new event tree sequences generated. To link the event and generate new sequence logic, perform the following steps (for more detail on linking, see Lesson 8):

1. Select the Event Tree from the main event tree list.
2. Click the right mouse button to evoke the pop-up menu. Move the highlight cursor down to Link and select with the left mouse button.

The Event Tree Link screen is displayed (Figure 47). To view the new logic and confirm that the substitution took place, click the check box labeled “Create Report?” to create an on-screen report. Click OK (the other options should be left at their default values).

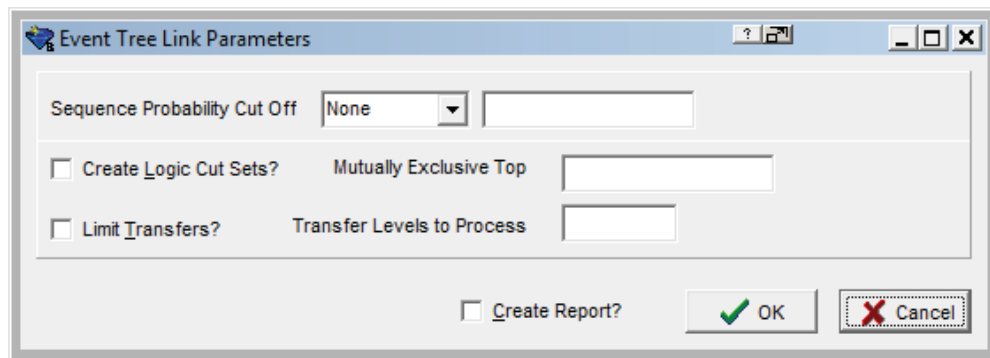


Figure 47. Event tree link creator

### 3.2.4 STEP 4: Re-solve For Sequence Cut Sets

The new sequence can now be solved with the fault tree top event logic being used. To solve for cut sets, perform the following steps:

1. Select the Event Tree(s) from the main event tree list.
2. Right click to invoke the pop-up menu and select the Solve option.

The Cut Set Solve window will be displayed. Adjust the cut set truncation as indicated in Section 2.8.2. Click the Solve button to accept the default cut set generation and truncation settings and to start the analysis. After the sequence cut sets have been generated, close the summary screen.

To display the event tree cut sets, select the event tree then right click to invoke the pop-up menu. Select the View Cut Sets option. The cut sets generated for event tree OBSTRUCT are now displayed as shown in Figure 48. From this screen, details of each cut set and the minimal cut set upper bound approximation can be observed by clicking the plus [+] node to the left of each cut set. The details for all cut sets can be seen by clicking the Expand All button.

Select the Close button when complete.

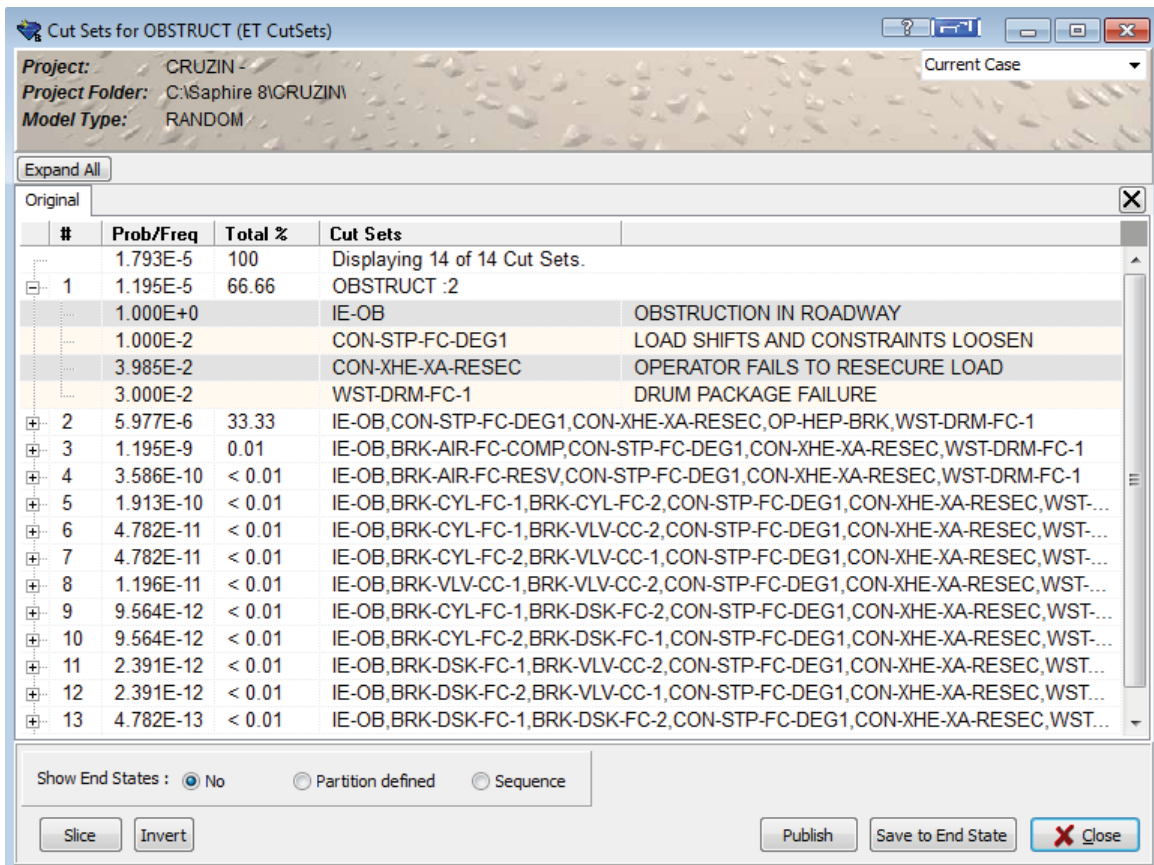


Figure 48. Event tree cut sets with a sequence flag set applied

### 3.3 Lesson 12 – Post-processing Rules

In Lesson 12, the user will be introduced to the “Post-processing Rule” editor. This editor allows rules to be written that affect existing cut sets in a “post-processing” fashion. Note that in previous versions of SAPHIRE (Version 7 and prior) that the post-processing rules were called “recovery rules.” The SAPHIRE Post-Processing Rules are logic rules that allow for the alteration or deletion of fault tree or sequence cut sets. The post-processing rules can be used to add recovery events to both sequence or fault tree cut sets, add common-cause failure



events to cut sets, or eliminate events or combination of events that cannot occur (i.e. mutually-exclusive events).

The Post-processing Rule editor uses standard programming (i.e. FORTRAN, BASIC) line format and structure. Symbols, nomenclature and structure for the post-processing Rule editor are very similar to the Linking Rule editor. The main difference between the two is that Linking Rules search the top events as search criteria while post-processing rules search existing cut sets as a search criteria. Standard common symbols in the SAPHIRE rules include:

- | Denotes a comment line
- ~ Logical operator for "not present"
- \* Logical AND operator (i.e. A AND B occur)
- + Logical OR operator (i.e. A OR B occurs)
- / Complement
- ( ) Parentheses for grouping terms

Once the post-processing rules have been written, the rule searches the either the sequence or fault tree cut sets for the specific search criteria specified by the user in the rule. Once the search criteria is met, the rule manipulates the cut set(s) as directed by the rule (i.e. add basic events to the cut set, delete basic events from the cut set, or delete the cut set altogether). Typical search criteria examples used by the linking rule editor are listed in Table 3.

**Table 3. SAPHIRE post-processing rule search criteria examples.**

Search Criteria	Example	Meaning of Search Criteria
A	<b>If A then</b>	If "A" occurs in the cut set then do this...
/A	<b>If /A then</b>	If "A" does not occur in the cut set then do this...
~A	<b>If ~A then</b>	If "A" is not present in the cut set then do this...
A * B	<b>If A * B then</b>	If "A" and "B" occurs in the cut set then do this...
A + B	<b>If A + B then</b>	If "A" or "B" occurs in the cut set then do this...
(A + B)* C	<b>If (A + B)*C then</b>	If "A" or "B" occurs and C occurs in the cut set then do this...
always	<b>If always then</b>	The search criteria are always met in every cut set so do this...

The Post-processing rules are implemented after the cut sets are generated using the Solve command (i.e., they search on existing cut sets). If the post-processing rules have been written, the rule searches all the cut sets for the search criteria specified in the rule and manipulates the cut set as directed. For example, the following post-processing rule would add a recovery event, B-REC, to any cut set that it found that contained both basic events A and B:

```
if A * B then
recovery = B-REC;
endif
```

**It is required by the rule editor that any action line (after a “then” statement) must end with a semicolon (;).**

There is no limit to the number of search criteria lines that can be used in a rule. More detailed information on rule structure, keywords, and nomenclature can be found in Technical Reference Guide.

This lesson will guide the user on adding a recovery event to fault tree LOAD1. The post-processing rule allows credit in the cut set results for the driver to isolate the spilled contents of drum if the Load constraints fail.

### **3.3.1 STEP 1: Creating a Post-processing Rule**

The post-processing rule example will be applied to fault tree LOAD1. Therefore, this specific post-processing rule is “fault tree post-processing rule” and will only be applied to the fault tree cut sets. If the user wanted it to be applied to the sequence cut sets, it would be written as a “sequence post-processing rule.” Fault tree post-processing rules are entered into SAPPHIRE through the Fault Tree Edit Post-processing Rules option.

The post-processing rule purpose is to add a recovery basic event, OEP-SPILL-ISO, to any cut set in fault tree LOAD1 that contains the basic event that represents the drum rupture (i.e., WST-DRM-FC-1) and the load constraints (or straps) failed (i.e. CON-STP-FC-BRK). To write a rule that accomplishes this function in the LOAD1 fault tree, perform the following steps:

1. Select the LOAD1 Fault Tree from the main screen.
2. Right click to invoke the pop-up menu. Select Edit Post-processing Rules. The rules editor will be displayed as shown in Figure 49.

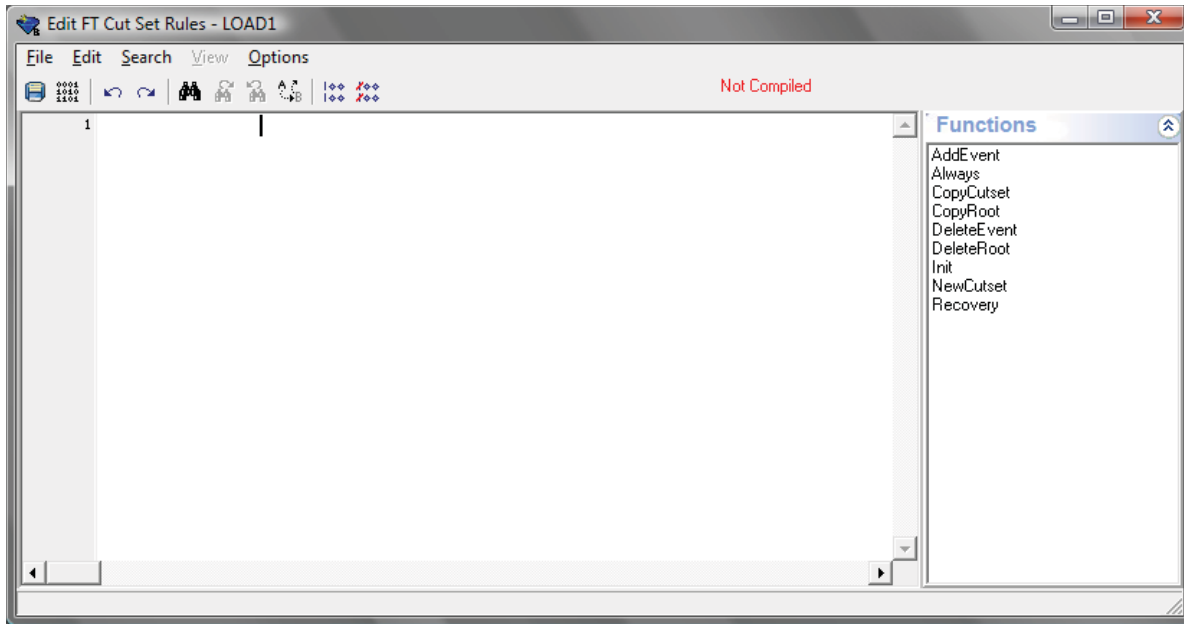


Figure 49. Fault tree post-processing rules editor

The post-processing rule editor requires that the basic event OEP-SPILL-ISO be added to the database prior to compiling the rule. Note that new basic events may be added to the project by using the “Add basic event...” option from the main window or via the File → New → Basic Event option.


Add a new basic event using the “Add basic event...” option found in the main list of basic event. Enter the Name as OEP-SPILL-ISO and a Mean Failure Probability of 1.0E-1 with a Lognormal distribution and an Error Factor of 3. Click on the Ok button when completed.

Now, back in the post-processing editor, type in the rule shown below.

```
| This post-processing rule appends the event OEP-SPILL-ISO to
| any cut set that has the basic events WST-DRM-FC-1 and
| CON-STP-FC-BRK in the cut set
if WST-DRM-FC-1 * CON-STP-FC-BRK then
    recovery2 = OEP-SPILL-ISO;
endif
```

---

<sup>2</sup> The “recovery” keyword is used to apply the basic event to the right of the “=” for cut sets meeting the search criteria specified in the rule.

To exit the post-processing rule editor, click on the Exit button  and click on Yes to save the post-processing rule (alternatively, select File → Save from the editor then Exit).

### 3.3.2 STEP 2: Re-solve For Fault Tree Cut Sets

The fault tree, LOAD1 can now be solved with the fault tree post-processing rule being implemented. To solve for cut sets, highlight the LOAD1 fault tree in the main fault tree list. Right-click to invoke the pop-up menu. Select the Solve option.

The Solve Cut Set dialog will be shown (Figure 50). Make certain that the “Apply Post-processing Rules” check box is checked. This automatically applies the post-processing rules during the Solve routine. Click Solve to continue.

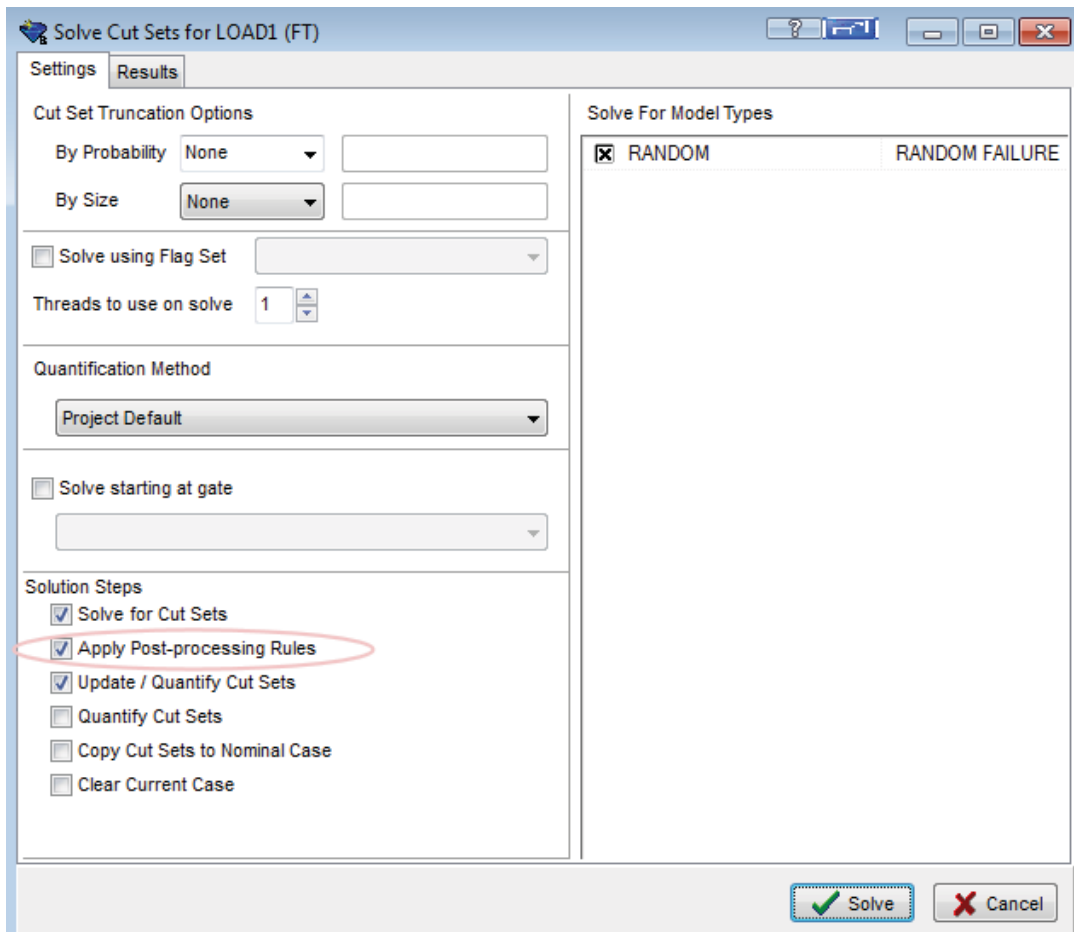


Figure 50. Fault tree cut set solve dialog

To display the cut set with the recovery event applied via the post-processing rules, select the LOAD1 fault tree and right-click to invoke the pop-up menu. Select View Cut Sets, which will bring up the cut set viewer (Figure 51). Notice that the recovery event OEP-SPILL-ISO is present in the cut set with the basic event, WST-DRM-FC-1 and CON-STP-FC-BRK.

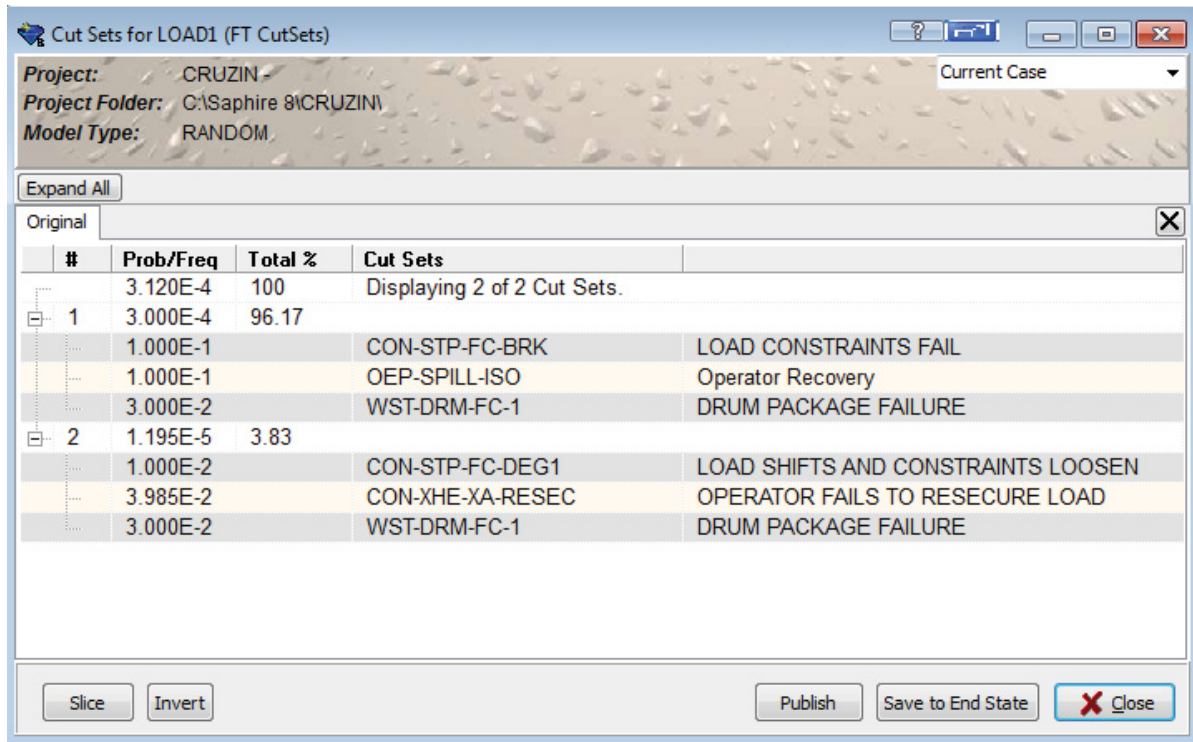


Figure 51. Fault tree cut set viewer showing the post-processed cut sets

### 3.4 Completion of the Tutorial

At this point in the process, both the basic functionality and some advanced features have been addressed. Stepping through the 12 different lessons demonstrated the event and fault tree development process and explained how to generate minimal cut sets using these models. For additional information on SAPHIRE 8, the reader is guided to the other volumes that accompany this report.



NRC FORM 335 (2-89) NRCM 1102, 3201.3202	U.S. NUCLEAR REGULATORY COMMISSION  <b>BIBLIOGRAPHIC DATA SHEET</b> (See Instructions on the reverse)	1. REPORT NUMBER (Assigned by NRC, Add Vol., Supp., Rev., and Addendum Numbers, if any.) <b>NUREG/CR-7039, Vol. 4          INL/EXT-09-17012</b>			
2. TITLE AND SUBTITLE  <b>Systems Analysis Programs for Hands-on Integrated Reliability Evaluations          (SAPHIRE) Version 8</b>  <b>Volume 4: Tutorial</b>	3. DATE REPORT PUBLISHED <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">MONTH</td> <td style="text-align: center;">YEAR</td> </tr> <tr> <td style="text-align: center;">MARCH</td> <td style="text-align: center;">2011</td> </tr> </table>	MONTH	YEAR	MARCH	2011
	MONTH	YEAR			
	MARCH	2011			
4. FIN OR GRANT NUMBER  <b>N6423</b>					
5. AUTHOR(S)  <b>C. L. Smith, S. T. Beck, S. T. Wood</b>	6. TYPE OF REPORT <b>Technical</b>				
	7. PERIOD COVERED (Inclusive Dates)				
8. PERFORMING ORGANIZATION - NAME AND ADDRESS (If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address; if contractor, provide name and mailing address.)  <b>Idaho National Laboratory          Battelle Energy Alliance          P.O. Box 1625          Idaho Falls, ID 83415-3850</b>					
9. SPONSORING ORGANIZATION - NAME AND ADDRESS (If NRC, type "Same as above"; if contractor, provide NRC Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address.)  <b>Division of Risk Analysis          Office of Nuclear Regulatory Research          U.S. Nuclear Regulatory Commission          Washington, DC 20555-0001</b>					
10. SUPPLEMENTARY NOTES  <b>D. O'Neal, NRC Project Manager</b>					
11. ABSTRACT (200 words or less) <b>The Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Version 8 refers to a set of computer programs that were developed to create and analyze probabilistic risk assessment (PRAs). This volume is the tutorial manual for the SAPHIRE system. In this document, a series of lessons are provided that guide the user through basic steps common to most analyses performed with SAPHIRE. The tutorial is divided into two major sections covering both basic and advanced features. The section covering basic topics contains lessons that lead the reader through development of a probabilistic problem involving a vehicle accident. The advanced features section contains additional lessons that expand on fundamental analysis features of SAPHIRE and provide insights into more complex analysis techniques. Together, these two elements provide an overview into the operation and capabilities of the SAPHIRE 8 software.</b>					
12. KEY WORDS/DESCRIPTORS (List words or phrases that will assist researchers in locating the report.)  <b>SAPHIRE 8, software, reliability, risk, safety, PRA, tutorial, fault tree event tree</b>	13. AVAILABILITY STATEMENT <b>Unlimited</b>				
	14. SECURITY CLASSIFICATION (This page) <b>Unclassified</b> (This report) <b>Unclassified</b>				
	15. NUMBER OF PAGES				
	16. PRICE				