INL/EXT-05-00654
Rev. 1

Systems Analysis Programs for Hands-On Integrated Reliability Evaluations (SAPHIRE) Tutorial

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

August 2008

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

INL/EXT-05-00654
Rev. 1

Systems Analysis Programs for Hands-On Integrated Reliability Evaluations (SAPHIRE) Tutorial

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

August 2008

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 N6203

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)

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

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 system. In this document, a series of lessons are provided that guide the user through basic steps common to most analyses preformed 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 hypothetical problem involving a vehicle accident, highlighting the program's most fundamental features. 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 software.

FOREWORD

The U.S. Nuclear Regulatory Commission has developed the Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) software 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.

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 can also 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 it includes special features for transforming models built for internal event analysis to models for external event analysis. It can also be used 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. The versions currently in use (6 and 7) run in the Microsoft Windows® environment. A user-friendly interface, Graphical Evaluation Module (GEM), streamlines and automates selected SAPHIRE inputs and processes for performing event assessments.

SAPHIRE has also evolved with users' needs, and Versions 6 and 7 include new features and capabilities for developing and using larger, more complex models. For example, Version 7 can solve up to 2 million sequences and includes enhancements for cut set slicing, event tree rule linkage, and reporting options.

This NUREG-series report comprises seven volumes, which address SAPHIRE/GEM Versions 6 and 7. Volume 1, "Overview/Summary," gives an overview of the functions available in SAPHIRE and presents general instructions for using the software. Volume 2, "Technical Reference," discusses the theoretical background behind the SAPHIRE functions. Volume 3, "SAPHIRE Users' Manual," provides installation instructions and a step-by-step approach to using the program's features. Volume 4, "SAPHIRE Tutorial Manual," provides an example of the overall process of constructing a PRA database. Volume 5, "GEM/GEMDATA Reference Manual," discusses the use of GEM. Volume 6, "SAPHIRE Quality Assurance (QA) Manual," discusses QA methods and tests. Lastly, Volume 7, "SAPHIRE Data Loading Manual," assists the user in entering PRA data into SAPHIRE using the built-in MAR-D ASCII-text file data transfer process.

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

CONTENTS

PREVIOUS REPORTS ii

ABSTRACT iii

FOREWORD v

LIST OF FIGURES viii

LIST OF TABLES ix

EXECUTIVE SUMMARY xi

ACRONYMS xiii

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 17

2.3.1 Basic Event Symbols 17
2.3.2 Fault Tree Logic Gate Symbols 17
2.3.3 Creating a Fault Tree in SAPHIRE 18

2.4 Lesson 4 – Basic Event Information 27

2.4.1 Entering Basic Event Information as Base Case Data 29
2.4.2 Generating Basic Event Data to the Current Case 31

2.5 Lesson 5 – Graphical Output 33

2.5.1 Printing Event Tree Graphics 33
2.5.2 Printing Fault Tree Graphics 34

2.6 Lesson 6 – Generating Fault Tree Cut sets 36

2.6.1 STEP 1: Generating Fault Tree Cut Sets 36
2.6.2 STEP 2: Displaying Fault Tree Cut Sets 38
2.6.3 STEP 3: Reporting Fault Tree Cut Sets Results 40

2.7 Lesson 7 – Fault Tree Uncertainty Analysis 42

2.8 Lesson 8 – Generating Event Tree Cut Sets 46

2.8.1 STEP 1: Linking Event Tree Sequences 46
2.8.2 STEP 2: Solve For Sequence Cut Sets 48
2.8.3 STEP 3: Performing Sequence Uncertainty Analysis 51

2.8.4 Reporting Sequence Cut Set Results 54
3. Advanced Lessons 57
3.1 Lesson 10 – Creating Change Sets and Flag Sets 58
3.1.1 Creating a Change Set 58
3.1.2 Creating a Flag Set 64
3.2 Lesson 11 – The "Linking" Rule Editor 68
3.2.1 STEP 1: Creating the New Substitution Fault Tree 70
3.2.2 STEP 2: Creating the Linking Rule 72
3.2.3 STEP 3: Re-Linking the Event Tree Sequence 73
3.2.4 STEP 3: Re-solve For Sequence Cut Sets 74
3.3 Lesson 12 – Recovery Rules 75
3.3.1 STEP 1: Creating a Recovery Rule 77
3.3.2 STEP 2: Re-solve For Fault Tree Cut Sets 80
3.4 Completion of the Tutorial 82

LIST OF FIGURES

Figure 1 SAPHIRE Main Menu 4
Figure 2 SAPHIRE Open Project Screen 6
Figure 3 The OBSTRUCT Event Tree 7
Figure 4 Event Tree List 8
Figure 5 Example of the graphical event tree editor 10
Figure 6 Example of branch point editing 11
Figure 7 BRAKE-SYS fault tree 19
Figure 8 OP-MANEUVER fault tree 19
Figure 9 LOAD fault tree 20
Figure 10 Fault Trees List 21
Figure 11 Graphical Fault Tree Editor 22
Figure 12 Addition of Connecting Lines for Fault Tree BRAKE-SYS 24
Figure 13 Shape Definition Entry Box 26

Figure 14 Basic Event Listing 29

Figure 15 Modify Event Window 30

Figure 16 Cut Set Generation Screen 37

Figure 17 Cut Set Generation Results Screen 38

Figure 18 Displaying cut sets screen 39

Figure 19 Selected Cut Set Viewer 40

Figure 20 LOAD1 Substitution Fault Tree 71

LIST OF TABLES

Table 1 Basic Event Data Values 28

Table 2. SAPHIRE Linking Rule Search Criteria Examples 69

Table 3. SAPHIRE Recovery Rule Search Criteria Examples 76

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 package. 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
- Recovery 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.

ACRONYMS

IE	initiating event
INL	Idaho National Laboratory
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)

Vol. 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. 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 Change Sets and Flag Sets
This lesson describes the difference between a flag sets and a change 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. Recovery Rules
Lesson 12 introduces the user to the "Recovery Rule" editor of SAPHIRE. The recovery 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:

1. 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.

2. The user has a basic knowledge of event trees and fault trees.

3. The user has read Section 2.0 of the SAPHIRE Reference Manual 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 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 semi-truck hauling hazardous waste encounters an obstruction in the roadway (downed tree, fallen rock, animal, 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 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 hazardous 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 1, the user will add a new project called PRACTICE. This is the first step in developing the sample example.

SAPHIRE is a menu-driven application. The Main Screen consists of a title bar, menu bar, and a graphical tool bar. The main menu is shown below in **Figure 1**. SAPHIRE allows the user various alternatives to set up the menu structure. To select any of the menu options, maneuver the mouse pointer over the desired option in either the menu bar or the graphical tool bar and click the left mouse button. Both the menu bar and graphical tool bar perform the same function in accessing the menu options. For the remainder of the tutorial, only the graphical tool bar will be referenced.

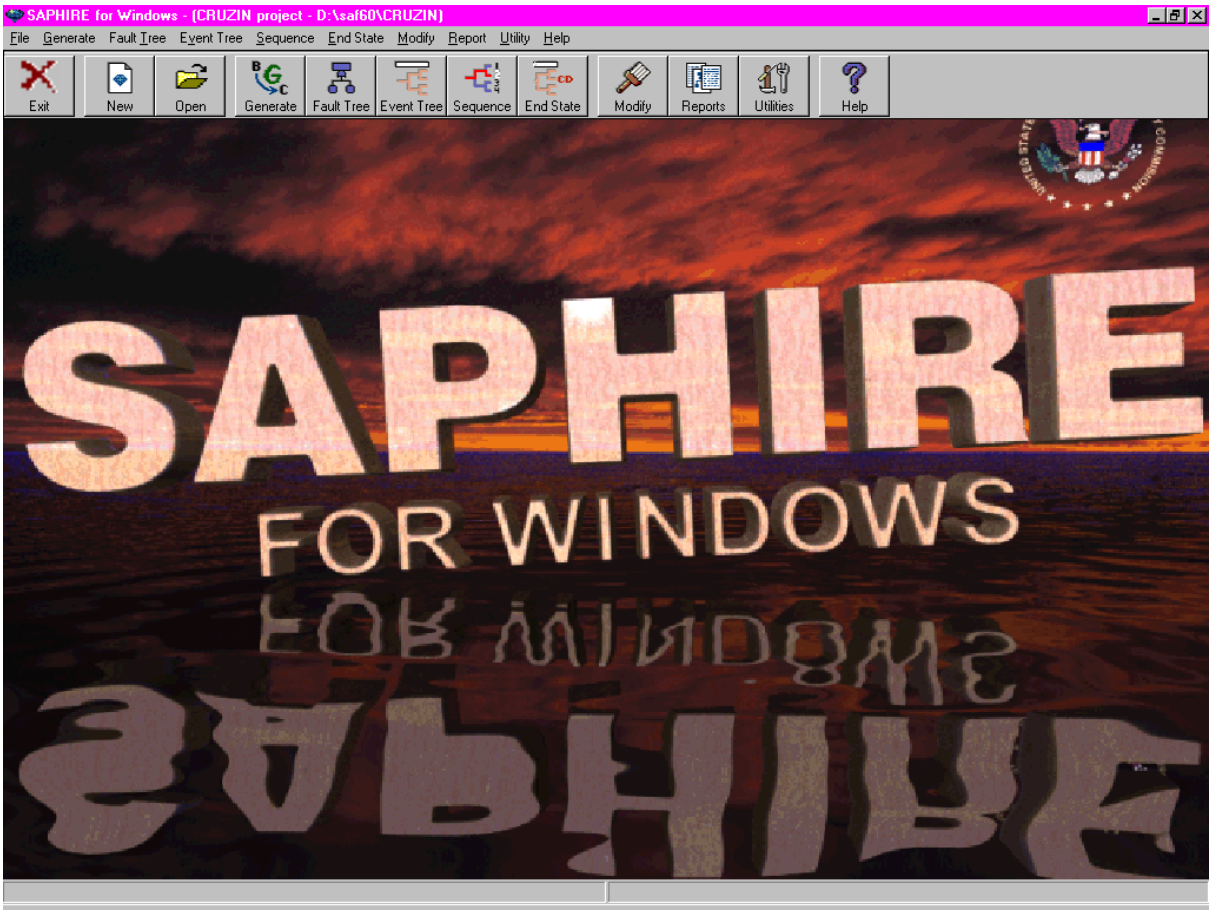


Figure 1 SAPHIRE Main Menu

To change the menu appearance, select **Utilities** in the Toolbar and then **Constants** and select the desired display method, Toolbar buttons, and Toolbar positions by selecting the desired radio buttons.

STEP 1: Add a New Project

The SAPHIRE database structure is divided into "projects". A project is any logical grouping of fault trees, event trees, cut sets and basic events with their respected reliability and uncertainty data. Each project in 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 **New** on the *Graphical Tool Bar* (or **File** on the *Menu Bar*, then **New Project**). SAPHIRE will ask if the user wants to "create a new project". Select **Yes.**

The "New Project-Project Name" box will appear. Type in the new project name, 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:\SAF60). However, the user has the ability to change to the location of the project folder to any location by clicking on the "Browse…" button. Click on **OK** to create the new project folder.

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

STEP 2: Select a Project (Optional)

Since the new project created in SAPHIRE is automatically selected after its 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 (for example, the DEMO project), this step would be necessary.

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

Select **Open** on the *Graphical Tool Bar* (or **File** on the *Menu Bar*, then **Open Project**). In the browse box, select the desired project folder using the drop down menu button or the "Up Folder" button as shown in **Figure 2**.

Select the project name ".SRA" file (i.e. <project name>.SRA) and click on **Open**. The project is now selected and available for analysis.

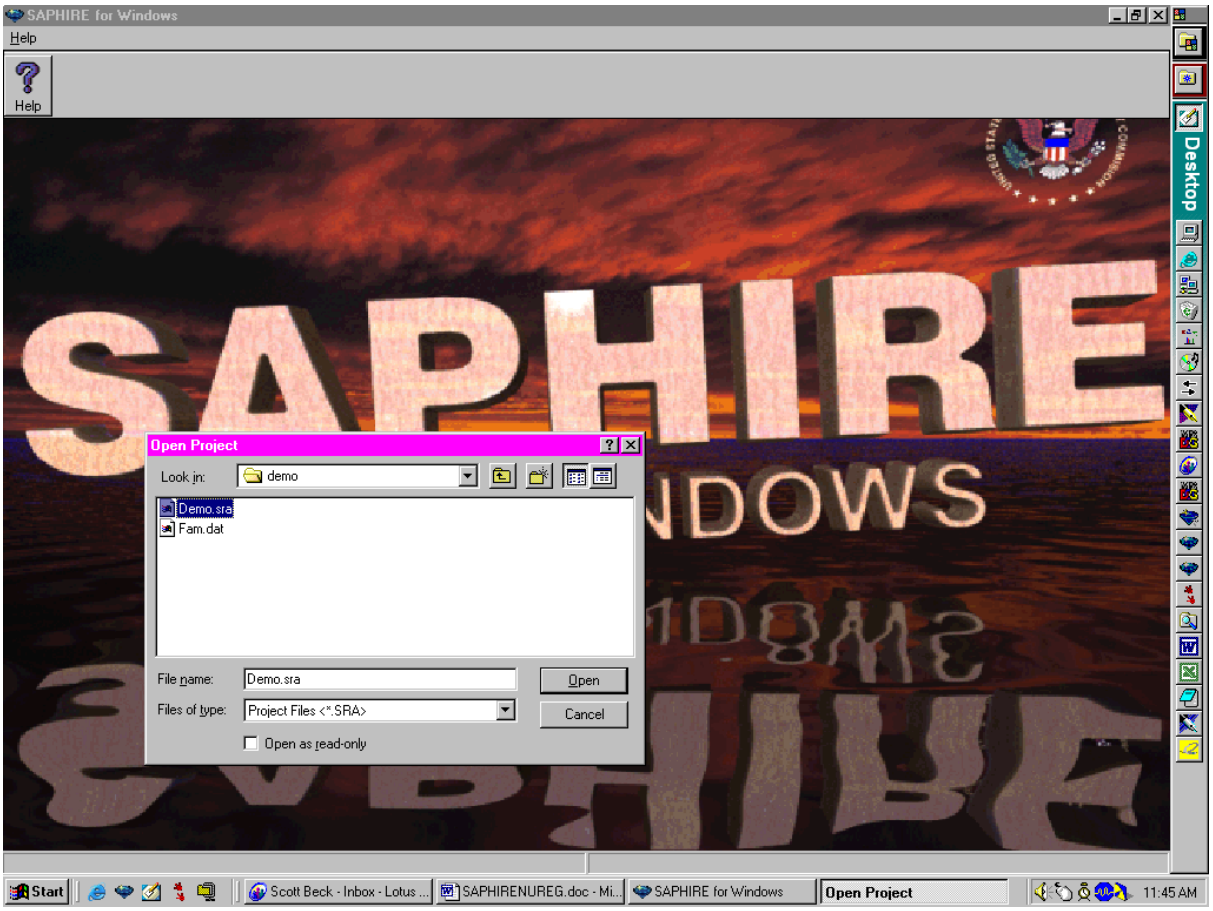


Figure 2 SAPHIRE Open Project Screen

2.2 Lesson 2 – Building Event Trees

In Lesson 2, the user will create the "OBSTRUCT" event tree shown below in Figure 3**.** 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.

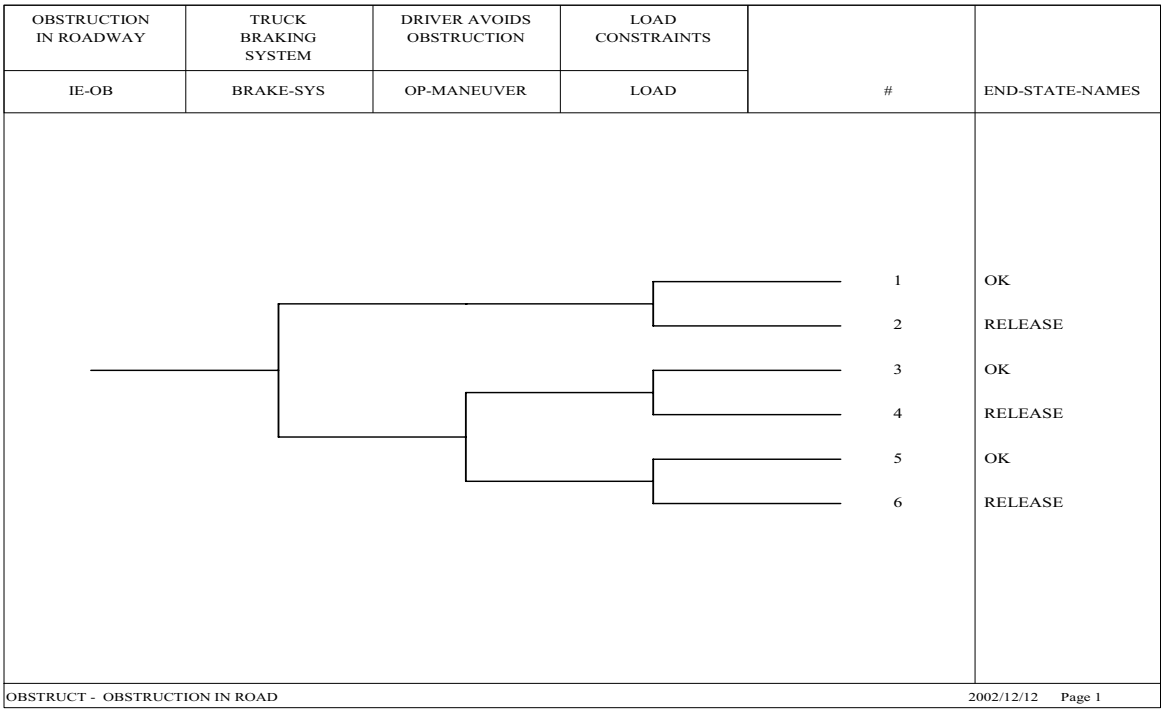


Figure 3 The OBSTRUCT Event Tree

Shown in **Figure 3**, 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".

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 an 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 3**, will be constructed using the detailed steps below.

To develop a new event tree, first access the event tree graphics feature by selecting the **Event Tree** button in the Main Menu (6th tool button from the left). This will bring up a list of event trees in the database as shown in **Figure 4**. The current database for this tutorial has no event trees; therefore, the screen is blank with no event tree names listed.

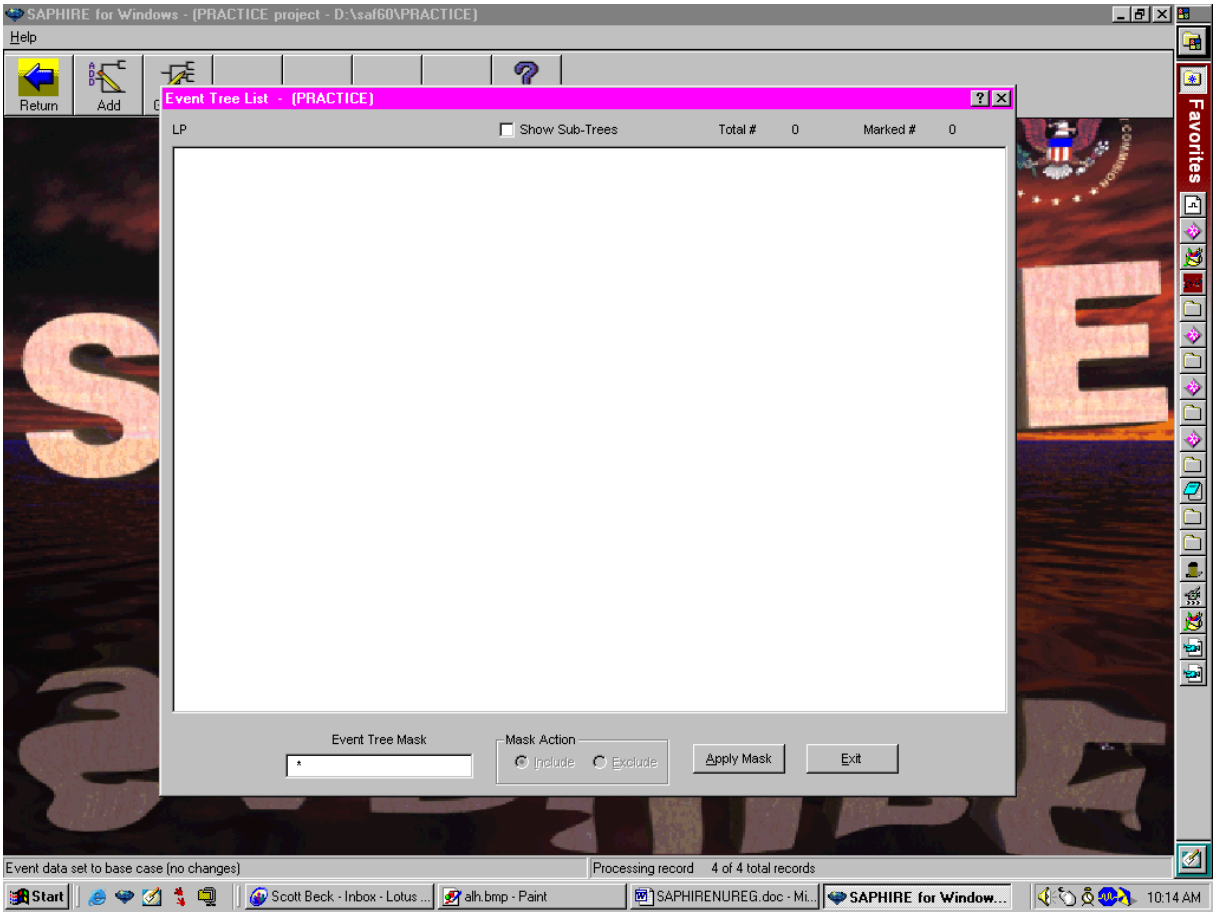
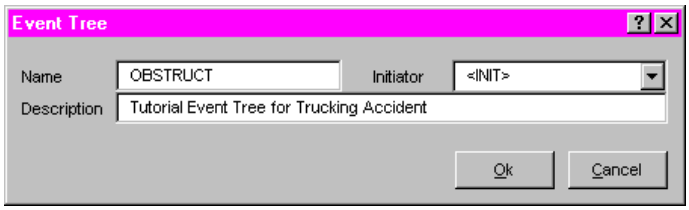


Figure 4 Event Tree List

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

Place the mouse cursor anywhere in the "Event Tree List" box and click the right mouse button to access the pop-up menu. Highlight **Add Event Tree** and then click the left mouse button.

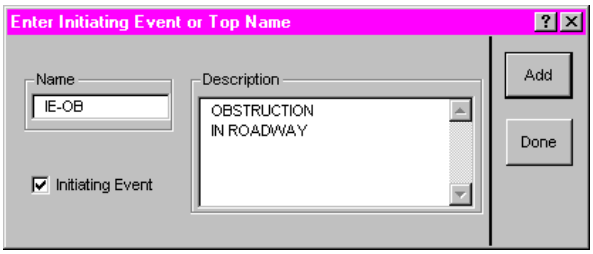
The "Event Tree" box will be displayed. Type in the word "OBSTRUCT" for the Name of the event tree and any description desired. Click on the OK button when completed. There now will be the event tree OBSTRUCT listed in the Event Tree List box.



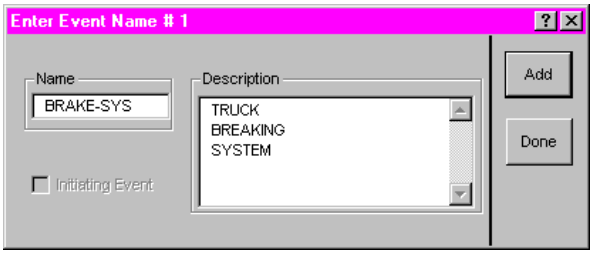
Highlight the new OBSTRUCT event tree name and click on the right mouse button. Highlight the "Edit Graphics" option.

SAPHIRE will inform the user that no graphic exist for the selected event tree and ask if the user would like to create the event tree. Click on **Yes**.

A series of questions regarding the top events will be displayed. The first question asks for the initiating event name and description. Type in IE-OB for the **Name** and any **Description** desired. Make certain that the **Initiating Event** box is checked ([). When finished, click on the **Add** button.



The next box displayed is the "Enter Event Name #1" box. Type in BRAKE-SYS and any description desired and then click on the Add button.



The "Enter Event Name #2" box will appear. Type in OP-MANEUVER and any description desired. Click on the Add button when finished. The "Enter Event Name #3 box will appear. Type in LOAD and any desired description. When all the top events have been entered, click on Done

After all the top events have been added, SAPHIRE will show the user the top event layout for the new fault tree. Figure 5 shows the Event Tree Editor screen that is displayed. From this screen, the user can edit any of the top event and descriptions, add branching, and add end states. Also in Figure 5 is an example of the graphical layout of the Event Tree Editor tool bar (located at the top).

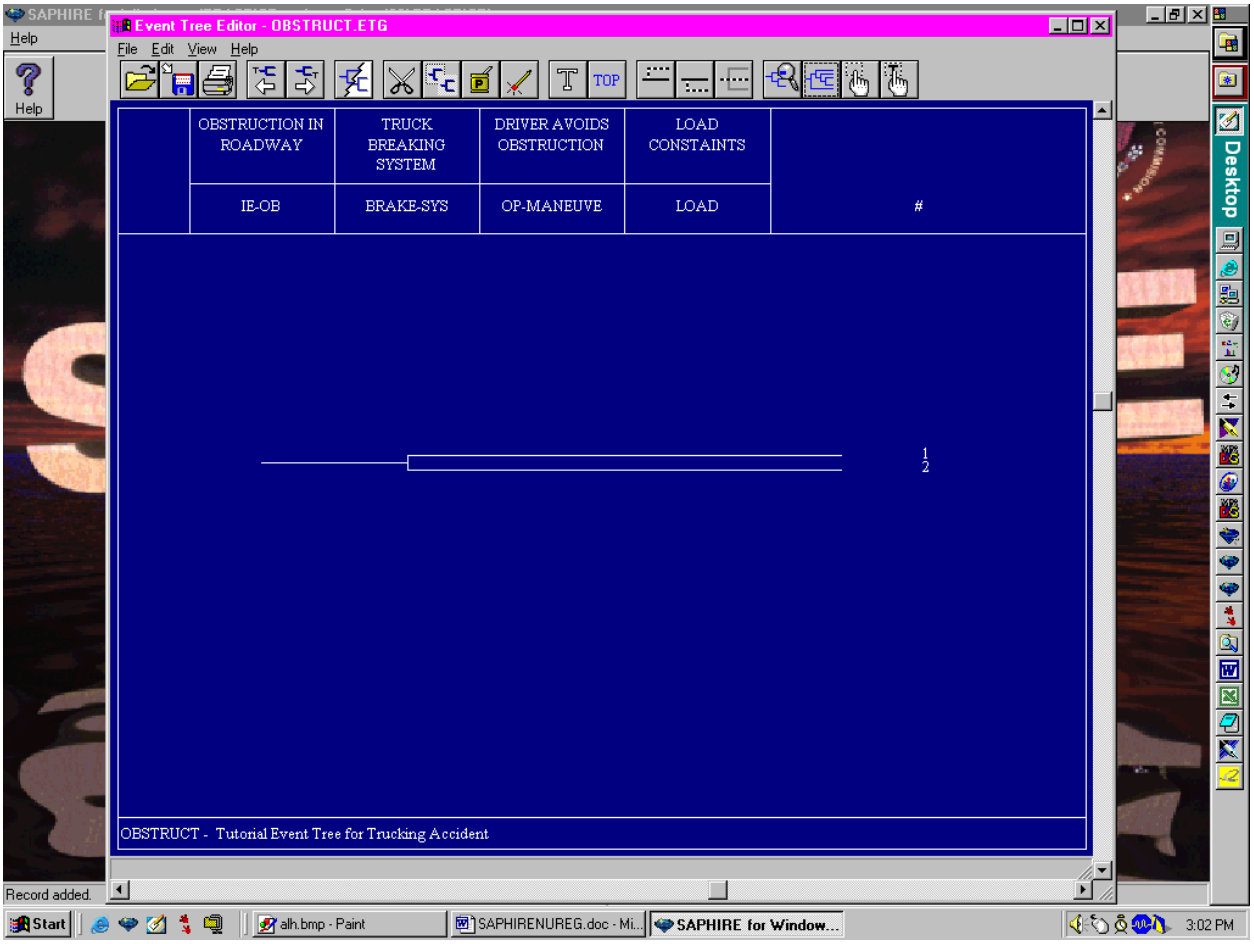


Figure 5 Example of the graphical event tree editor

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, Figure 2-3 show that 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-MANEUER, questions the top event, LOAD. To add these branches to the new event tree, OBSTRUCT, perform the following steps:

To add a branch under the top event LOAD, place the index finger of the pick curser on the branch point of the success branch of BRAKE-SYSTEM directly under the LOAD top event. Click the right mouse button to select this as a branch point. The line at the branch point will turn yellow as shown in Figure 6.

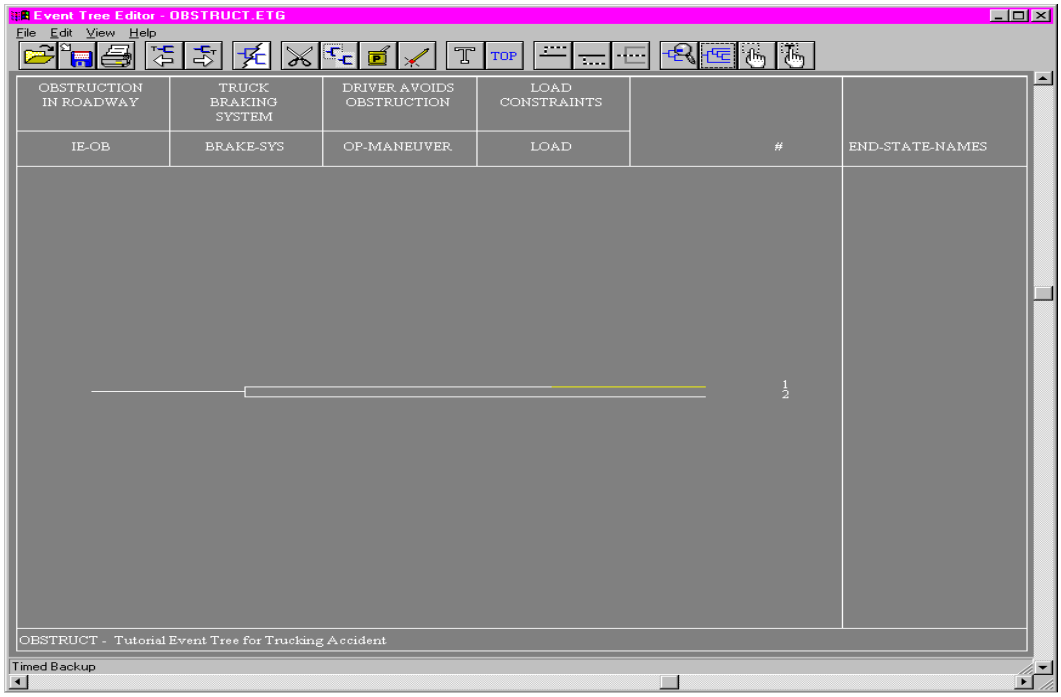


Figure 6 Example of branch point editing

QUICK TIP

Grid lines can be added to the Event Tree Editor to aid the user in determining top event branch points. To add grid lines to the Event Tree Editor Screen: Select from the menu bar **View → Show Grid**.

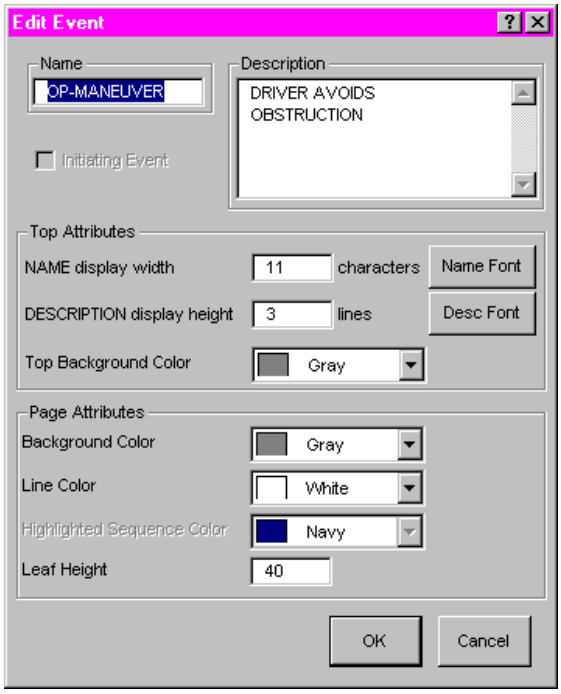
Add a branch by either selecting the "Branch-up" button on the *tool bar* or by evoking the pop-up menu by clicking the right mouse button and selecting **Add Branch Above.** A third option to add a branch is to select the branch point then, in the *menu bar*, click on **Edit → Add Branch Above**.

To delete an unwanted branch, Use the point cursor to select the unwanted branch (turns yellow). Select **Edit → Delete** from the menu bar or use the pop-menu (right click) and select **Delete**.

Continue Steps 1 and 2 for each of the branch points in **Figure 6**.

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, select the top event using the Pick cursor (place the cursor on the top event and click the left mouse button. Click the right mouse button. The **Edit Event** dialog box will be displayed. From this box, the Name and Description of the event can be modified. In addition, font sizes and colors can be changed.



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-

To delete the top event, click on the button in the graphical tool bar with the left mouse button.

To add a top event**,** choose the "Top" object button from the graphical tool bar. 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.

QUICK TIP

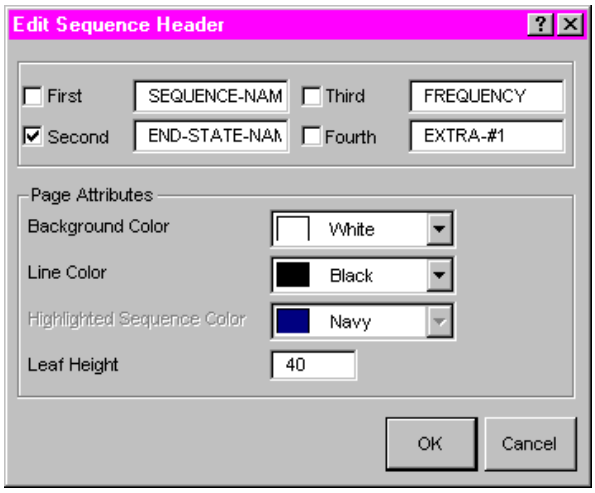
To change the Leaf Height (i.e. spacing between the event tree branches), evoke the pop-up menu by clicking the right mouse button anywhere on the event tree graphics screen. Increase the leaf height by increasing the number located in the **Leaf Height** box. This number is a "pixel" number so, increase the number by increments of 10 or 20 until the desired spacing is reached.

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

Step 3 of this section 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 column reserved for the sequence number. The default for SAPHIRE is to display only one of the header columns. However, the user can adjust the number of sequence header columns displayed, the text font and color, and header column titles by accessing the **Edit Sequence Header** dialog box.

To adjust the Sequence Headers Dialog, perform the following steps:

1. To modify sequence header columns or activate end-state and sequence headers, chose the pick button and position cursor over the any of the sequence header column fields. Click the right mouse button. The Edit sequence header dialog box, as shown below, will be displayed.



The four dialog boxes with a check mark are the Sequence Header columns that are currently displayed. To activate the other Sequence Header columns, place the cursor over the check box and click the left mouse button to add a check mark. To deactivate a Sequence Header column (i.e. not displayed on the event tree), click on the check mark with the left mouse button to remove the check mark.

QUICK TIP

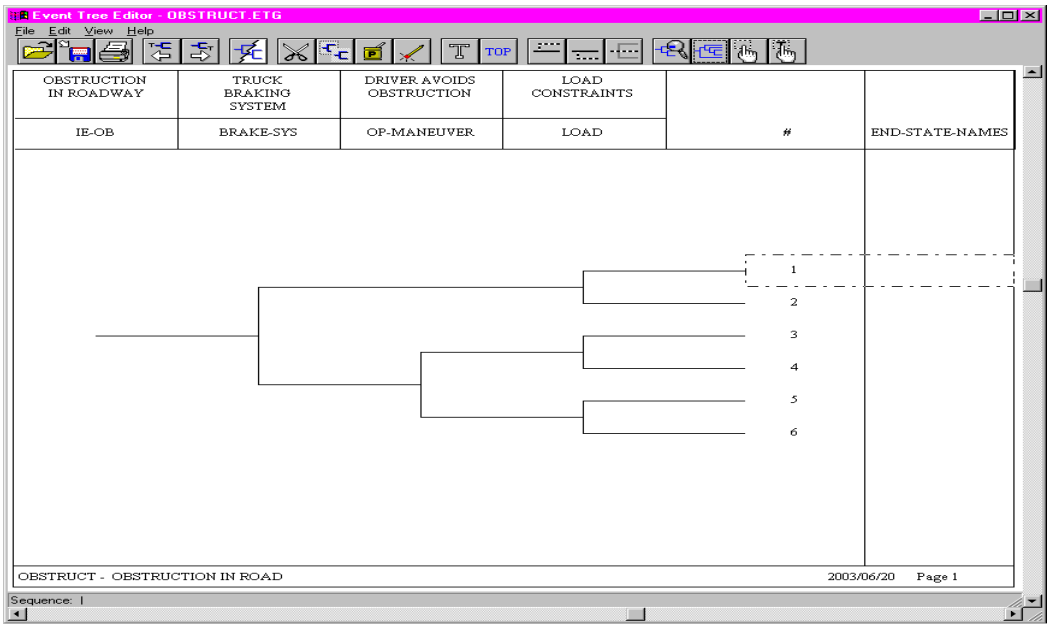
To move or change the width of a sequence header, position the pick cursor over the line of the desired column. The cursor will change to the ↔ cursor. Drag the mouse with the left mouse button pressed to the desired location and then release the mouse button to set the new column location.

To change the Sequence Header description, move the mouse cursor into a header box that is to be change and modify the header name as desired. When finished, select the **OK** button.

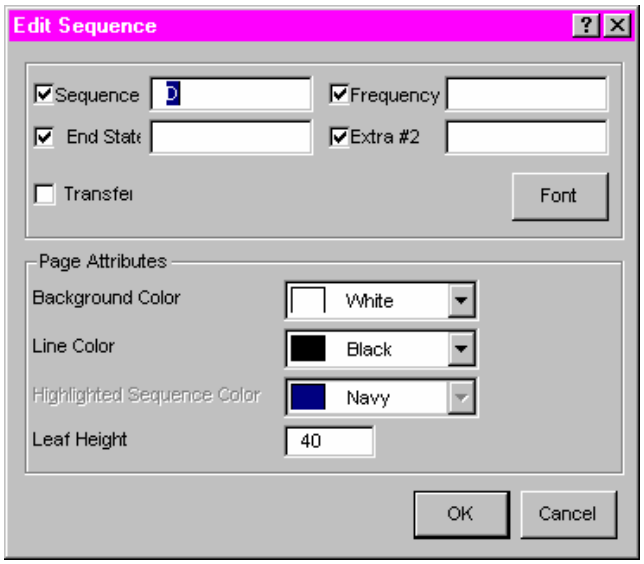
On the far right of the event tree are the end-state headers. There are four header columns.

Each sequence branch in the event tree needs 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, the "@" symbol, or the word "Success"—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 accessing the **Edit Sequence** dialog box.
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 number "1" and clicking the left mouse key. The sequence header row will be highlighted with a dotted line as shown below.



With the sequence highlighted, click the right mouse button to evoke the **Edit Sequence** dialog box as shown below:



At the top of Edit Sequence dialog box there are four available sequence information boxes, each with check box used to evoke them for display in the event tree graphics. The fields “Sequence”, “Frequency” and “Extra 2” are additional information boxes used to display extra text about the sequence. These fields

are not required to be filled. The "End State" field is the name of the sequence end-state. This field needs to be filled in order for SAPHIRE to generate sequences. For success branches, this field can be either OK, @, or Success.

For sequence number 1, move the mouse cursor to the End-state field and click the left mouse button. Since this is a success branch, type in the work "OK" in this field. Make sure this field is checked ([) to ensure that this sequence end-state appears in the event tree graphics.

Highlight sequence number 2. This is a down (or failure) branch and results in some consequence. Click the right mouse button to obtain the **Edit Sequence** dialog box. Enter in to the End State box the end-state name RELEASE and ensure that this sequence box is checked. Click **OK** when completed.

Repeat step 3 for the remaining 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:

When all the sequence end states are labeled, click on the save button in the graphical tool bar or click on **File → Save** in the menu bar. This will save all modifications under the original event tree name.

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.

To exit the event tree editor, click on **File → Exit**. This will return the user to the **Event Tree List** box.

QUICK TIP

For modifying larger event trees, the zoom feature can be accessed. To access zoom, click on . Move the zoom cursor to the area of the event tree that needs to be zoomed. Click the left mouse button to zoom in and the right mouse button to zoom out.

To redraw the diagram to the original full screen view, click on button.

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 Basic Event Symbols

This section reviews the basic event 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.

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

This symbol also represents a **basic event**. The system provides a box to contain description text.

This symbol is used to represent a **table of basic events**. This system is used for space saving layouts.

Represents the **undeveloped event**. The undeveloped event is treated as a basic event but represents an event that requires further development (i.e. a possible system).

Represents a **"house event"**. A house event is used to denote a guaranteed failure (set to "True") or a guaranteed success (set to "False"). House events are generally used to turn specific gates or logic "on" or "off".

Represents an **undeveloped transfer**. This symbol is treated as a basic event. However, it indicates that the event is complex enough that it may require future development into it's own fault tree logic.

2.3.2 Fault Tree Logic Gate Symbols

This section reviews the logic gate system in SAPHIRE used in fault tree construction.

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

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.

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 2/5 gate, any combination of 2 of the 5 input events must occur for a failure to occur.

Represents an INHIBIT gate. The inhibit gate is a special type of AND gate. The logic operation for this gate requires that in the presence of the enabling condition (or conditioning event), and a single input fault occurs then a failure has occurred.

This symbol represents a NAND gate (or NOT AND gate). The logic output of a NAND gate occurs if any one of the inputs does not fail (success). The NAND gate is the same as an OR gate with complimented events (success events) for the inputs.

This symbol represents a NOR gate (or NOT OR gate). The logic output of a NOR gate occurs if none of the inputs fail. The NOR gate is a logically the same as a AND gate with complimented events (success events) for the inputs.

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.

This symbol is used to indicate where a transfer has taken place in the fault tree. It has no logical or functional purpose and is only used for graphical purposes.

This symbol is a descriptive text box. It is used for additional information to be added to the structure of the fault tree.

This symbol allows for the creation of connecting lines to connect the gates and basic event together. The connecting lines can be solid or dashed and can be drawn at various angles.

2.3.3 Creating a Fault Tree in SAPHIRE

Fault trees, or fault tree logic may be created with two methods in SAPHIRE. The first method is with the use of the Fault Tree Logic Editor. The fault tree logic editor is a convenient tool for quickly editing fault tree logic and entering gate and basic event data. Fault tree gate descriptions can be entered from this editor as well as basic event descriptions and data. The Fault Tree Logic Editor will not be discussed in example problem. Details for the fault tree logical editor can be found in the Technical Reference Guide. The second method to create fault tree is with the use of the Fault Tree Graphical Editor. This method will be discussed in detail for the example problem and will be used for the construction of all fault trees.

Figure 7, Figure 8, and Figure 9 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:

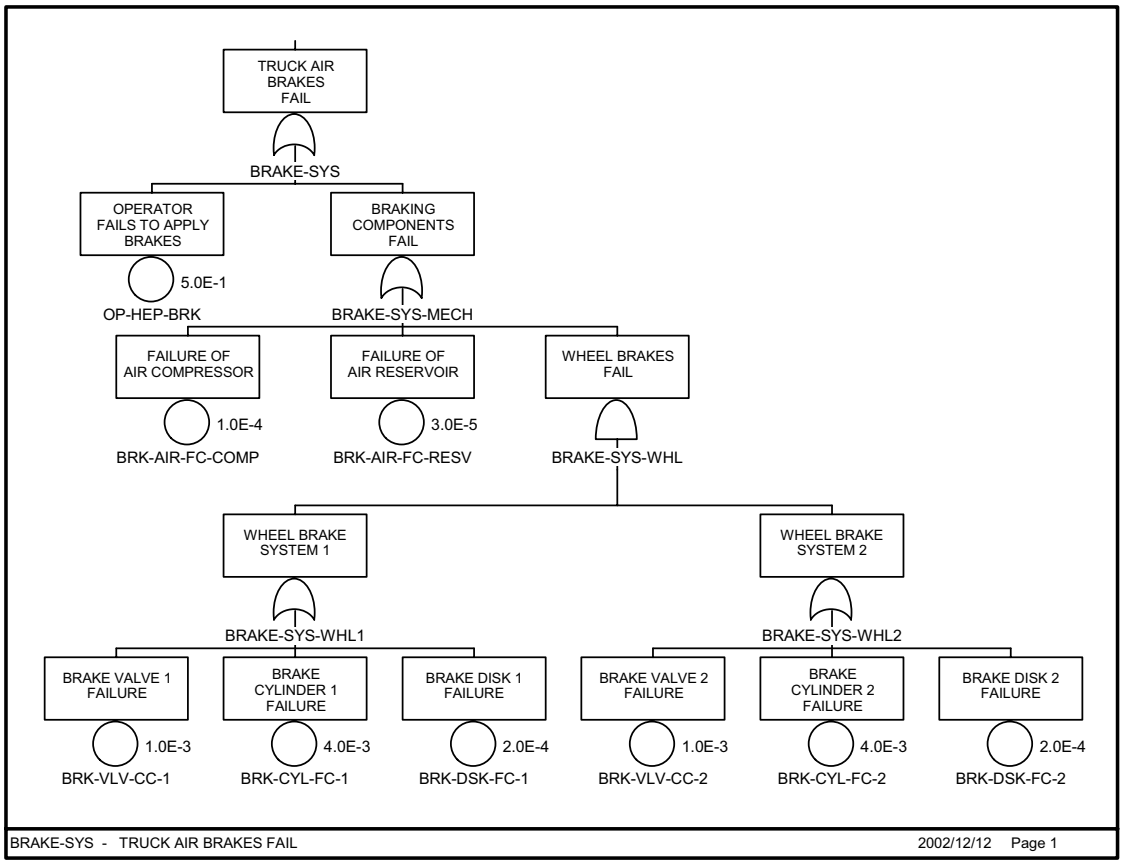


Figure 7 BRAKE-SYS fault tree

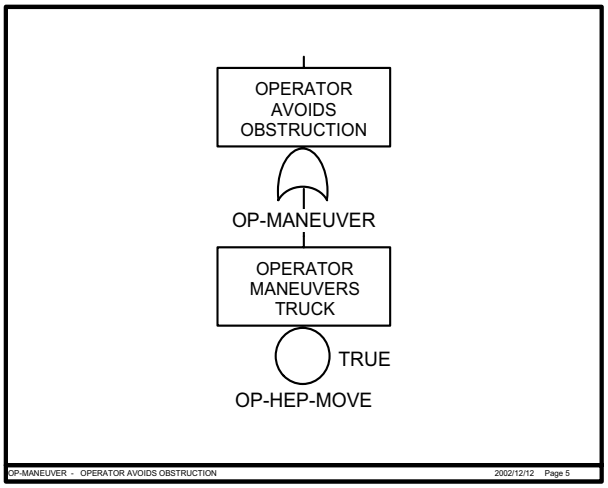


Figure 8 OP-MANEUVER fault tree

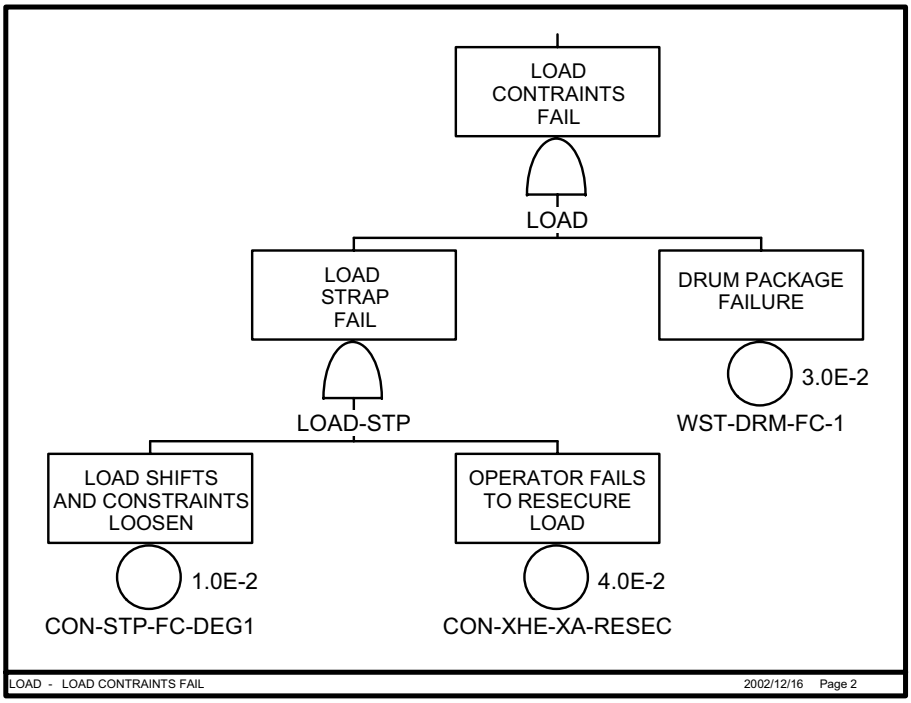


Figure 9 LOAD fault tree

The following steps guide the user on creating a new fault tree in SAPIRE. The three fault trees shown above in Figure 7 through Figure 9 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 selecting the **Fault Tree** button

on the graphical tool bar (5th button from the left) or click on Fault Tree from the menu bar. This will bring up a list of fault trees in the database as shown in Figure 10. The current database for this tutorial has three fault trees. These fault trees have no logic structure incorporated with them. They were automatically created when the event tree OBSTRUCT was created and used these as system top events.

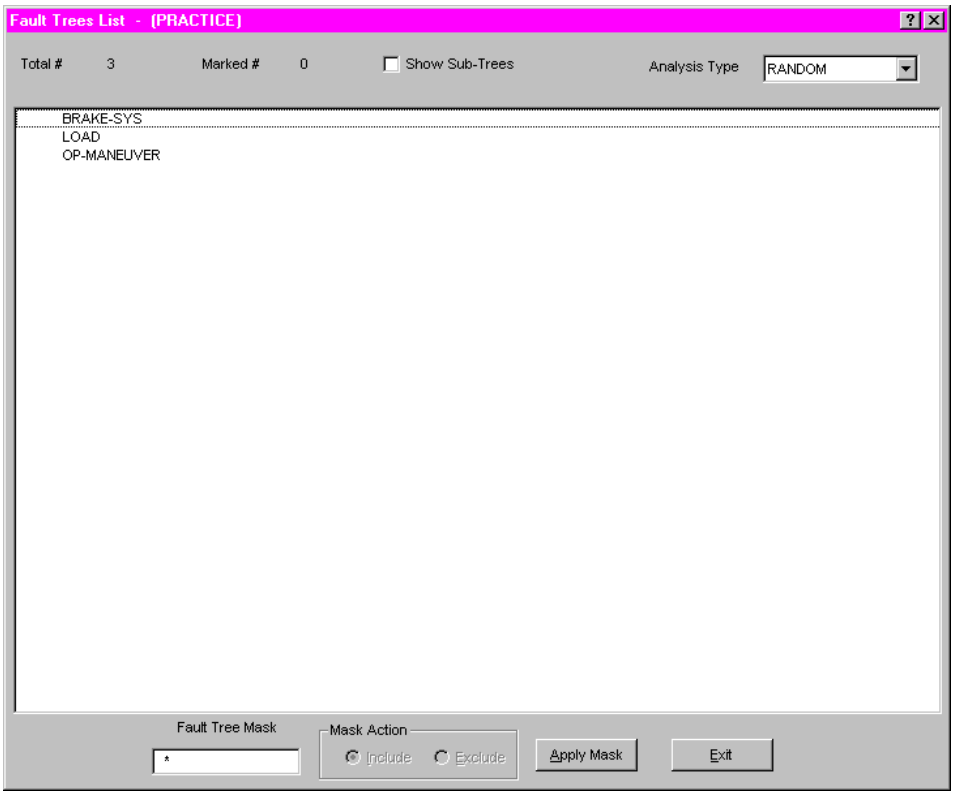


Figure 10 Fault Trees List

Highlight the BRAKE-SYS fault tree name and then left click to evoke the pop-up menu shown below.

Highlight **Edit Graphics** in the pop-up menu. A blank fault tree graphical editor screen will appear as shown in Figure 11. All the available graphical features need to complete any fault tree are displayed at the top of the editor in the tool bar.

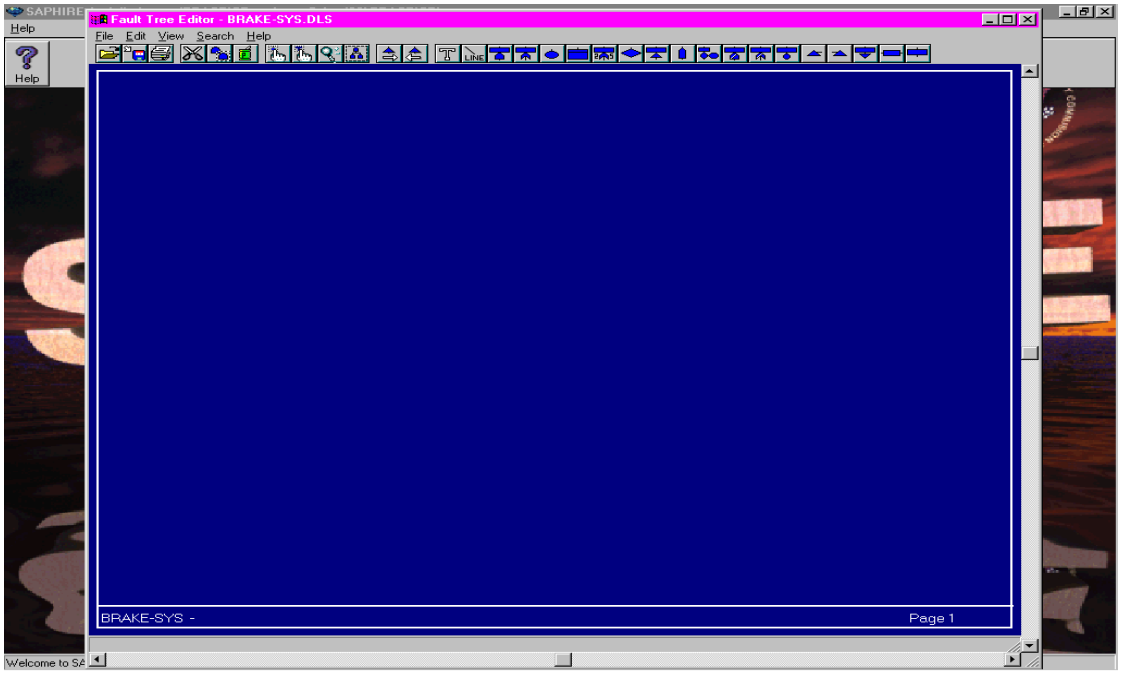


Figure 11 Graphical Fault Tree Editor

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. Move the cursor to tool bar and select the tool bar button (click the left mouse button). The cursor will change to a "OR" gate ghost cursor. Move the cursor to the desired location on the editor page and click the left mouse button to place the object on the page. The right mouse button terminates the selected ghost shape option and returns the cursor to the original "pick cursor". To add additional selected objects, position the ghost cursor at the desired location in the editor and click the left mouse button. The ghost cursor will remain the selected object until another object is selected or until the cursor is terminated with a right click.

NOTICE: *The placement of the fault tree symbols in the graphical editor is NOT a "drag and drop" procedure. Selecting the object and then holding down the left mouse key* **WILL NOT** *drag the object over to editor screen.*

Fault tree BRAKE-SYS shows that underneath the top gate is a single basic event and another OR gate. If the OR gate ghost cursor is still active, simply place the cursor at the desired location in the editor, and click the left mouse button. To add the "BASIC EVENT" symbol, move the cursor back to the tool bar and select the tool bar button. The cursor will change to a "BASIC EVENT" ghost cursor. Move the cursor to the desired location on the editor page and click the left mouse button to place object on the page.

STEP 3: Connecting Fault Tree Symbols with Lines
After the various logic symbols in the fault tree have been added, the logic gates and basic events must be linked together to form logical branches using connecting lines. To add connecting lines to the fault tree BRAKE-SYS perform the following steps:

Select the **LINE** button from the fault tree editor graphical tool bar.

Move the "LINE" ghost cursor to the desired starting location and **press and hold** the left mouse button. Drag the line cursor to the ending point of the connection line and **release** the left mouse button. The endpoint location can be anywhere in editor screen to lengthen, shorten, or change the direction of the line. **Figure 12** illustrates the three of the symbols and their connecting lines for fault tree BRAKE-SYS.

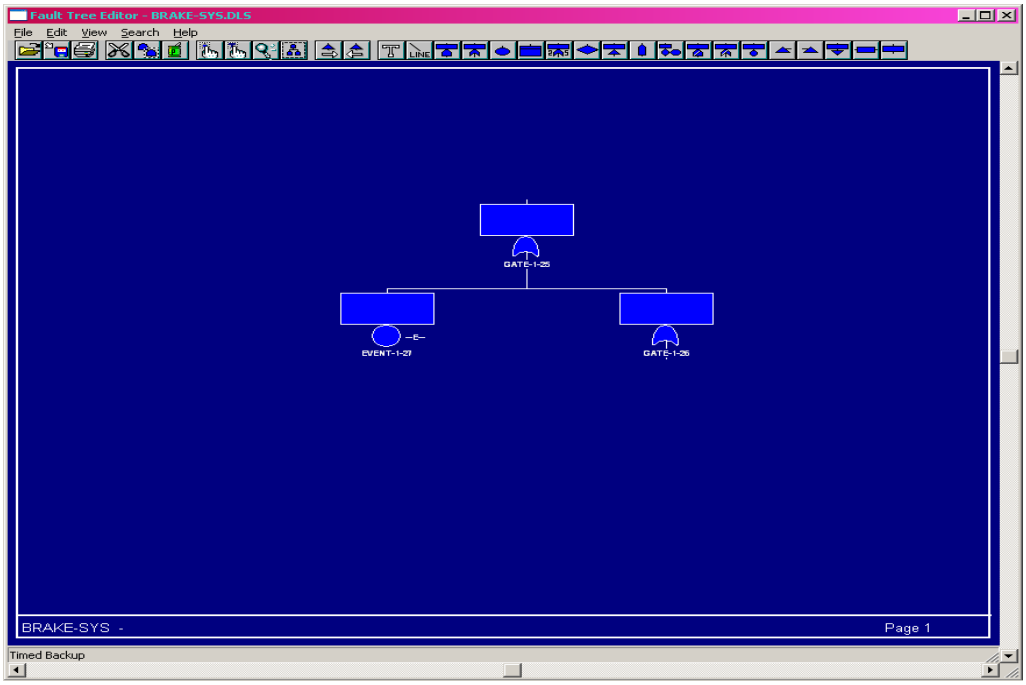


Figure 12 Addition of Connecting Lines for Fault Tree BRAKE-SYS

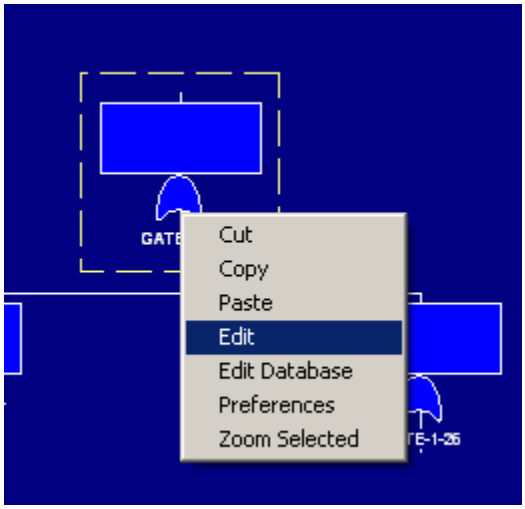
QUICK TIP

To **move or reposition** a fault tree symbol, place the "pick" cursor over the object to be moved. Press and hold on the left mouse button and move the highlighted object to the new, desired location.

STEP 4: Adding Symbol Definition

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 my placing the "pick" cursor over the object and clicking the left mouse button (the object selected will be highlighted with a dotted line.) For this example, click on the top OR gate of the fault tree, BRAKE-SYS. Click the right mouse button to evoke the drop-down menu shown below. Highlight EDIT and click the left mouse button. [An additional approach is from the *Fault Tree Editor* menu bar, **Select Edit → Attributes…**]



The "**Shape Definition**" window appears as shown in **Figure 13**. From this window, the user can change the shape type (example: change an OR gate to an AND gate) and color, add/modify shape names and descriptions, adjust text justifications, and change the connecting line characteristics.

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 7** through **Figure 9**.

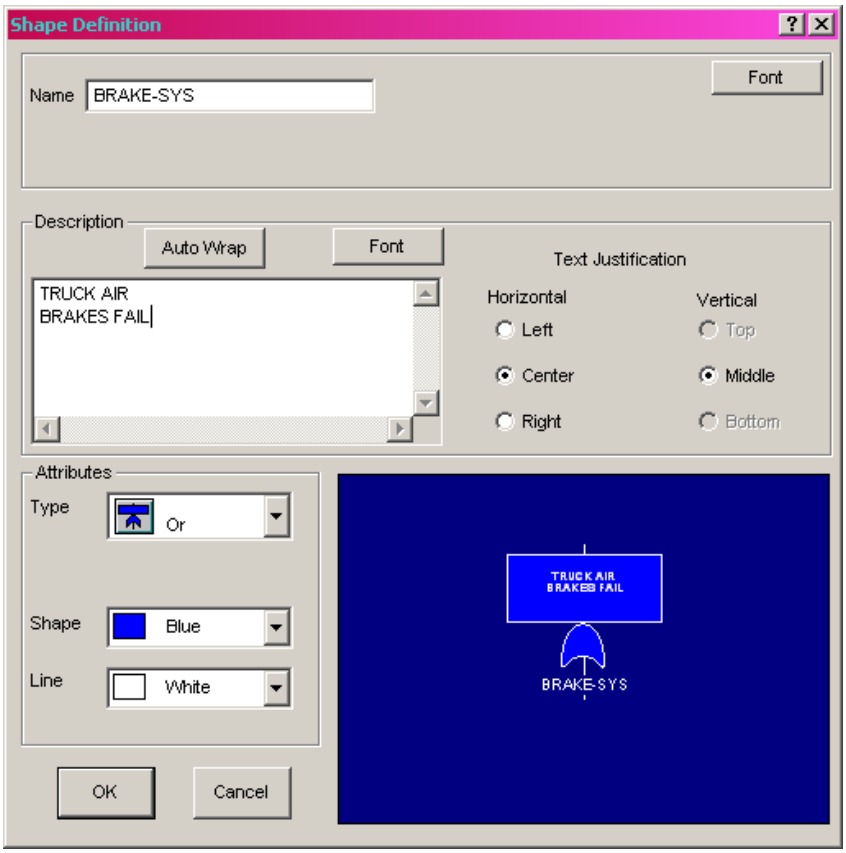


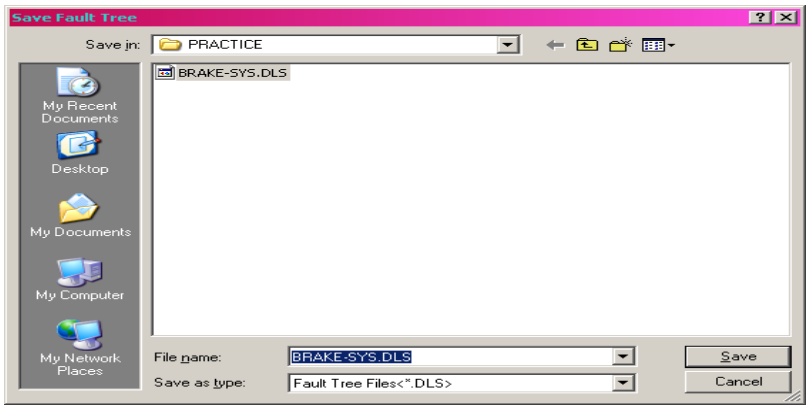
Figure 13 Shape Definition Entry Box

STEP 5: 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:

To save the fault tree graphic, click on the Save button located in the graphical tool bar (second button from the left). **Or** — From the *Fault Tree Editor* menu bar, select **File → Save as…**

For fault tree, BRAKE-SYS, save the file under the name "BRAKE-SYS".



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.

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, SAPHIRE added all the basic event names, gates, event tree top events (i.e. fault trees), and initiating events. 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 "base case" for use by SAPHIRE during an analysis. SAPHIRE separates the data in a project database into two separate parts: **base case** and **current case**. **Base Case** data is stored in the data base files as a "permanent" record. Current Case data is used to perform an analysis (e.g., cut set generation and quantification).

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.

Table 1 Basic Event Data Values*

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

* These values are for demonstration purposes only and may not be applicable for any actual application.

2.4.1 Entering Basic Event Information as Base Case Data

Basic event data loaded or entered into SAPHIRE is placed into the project database as base case data. 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 enter or modify basic event data, select **Modify** from either the graphical tool bar or the menu bar.

Then select **Basic Events**. A listing of all the basic events in the database will be displayed as shown in Figure 14.

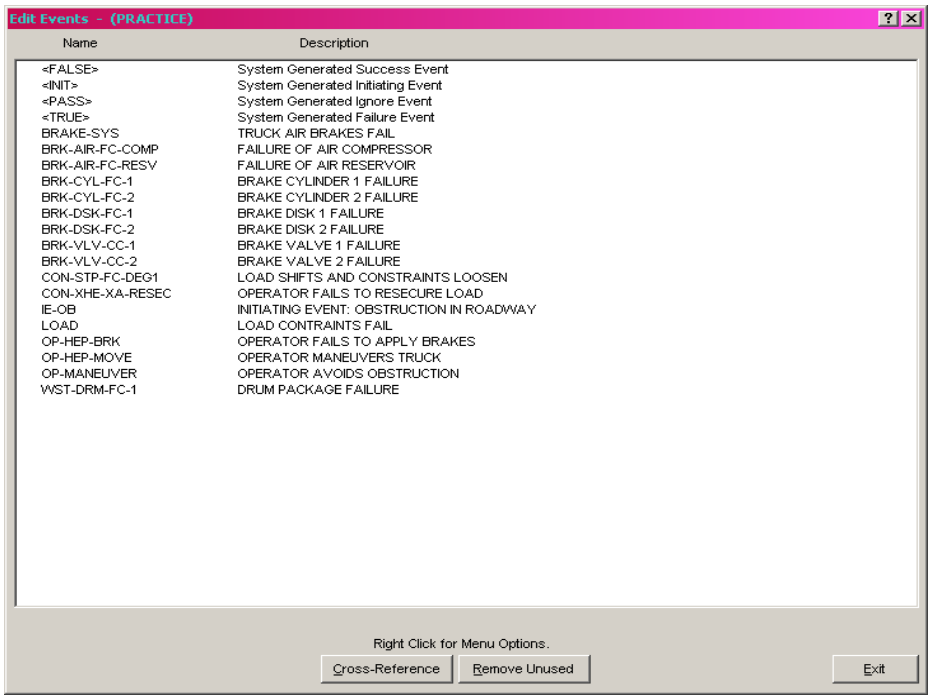


Figure 14 Basic Event Listing

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** from the drop down menu. [Or **Double Click** on the basic event to be modified]

The **Modify Event** screen appears as shown in Figure 15. From this window, all basic event data including uncertainties can be entered. The **Modify Event** screen has "Tab" buttons across the top. The Event tab is accessed automatically when the Modify → Basic Event option is implemented. This tab is used to input all failure and uncertainty data. The other tabs are used for adding other descriptive attributes, process flags and other advanced features. Specific details for these features can be found in

the SAPHIRE Reference Manual.

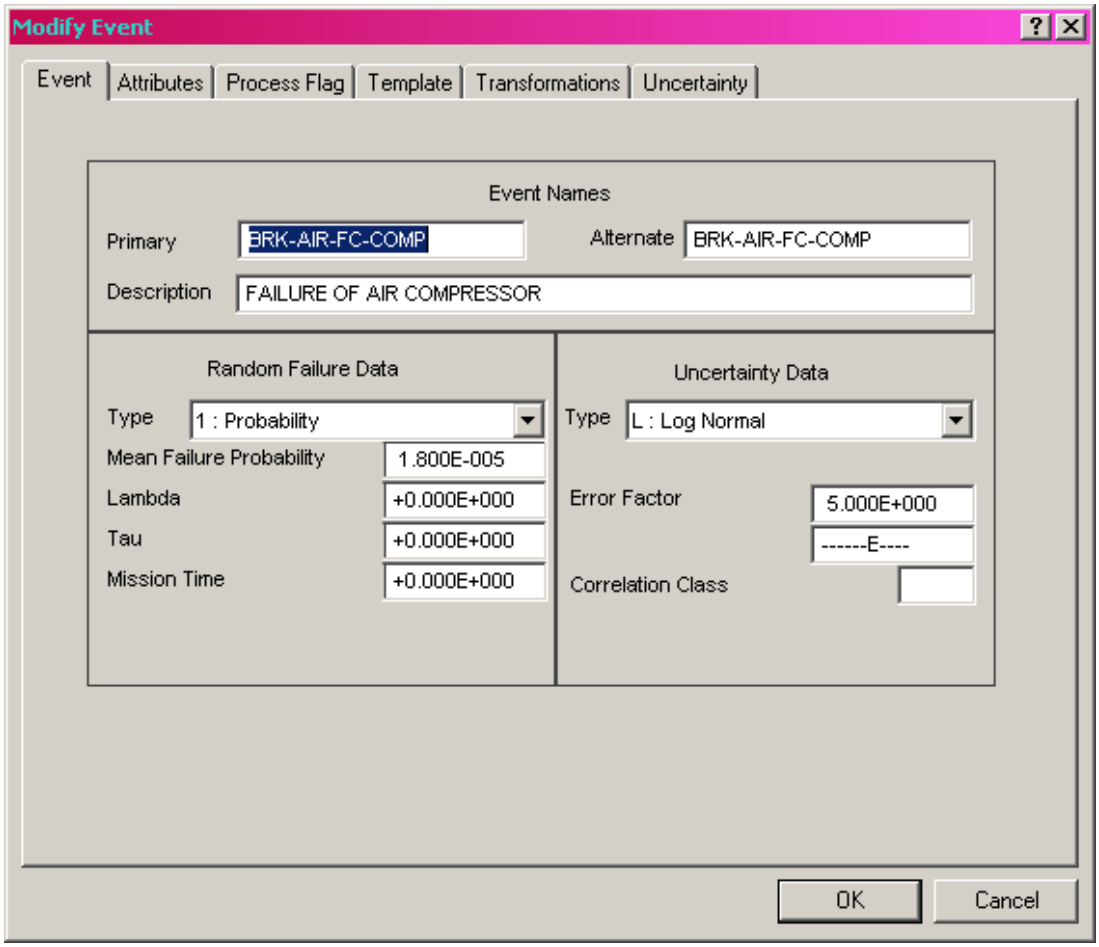


Figure 15 Modify Event Window

The left side of the **Modify Event Window** is where the **Random Failure Data** is entered. This area contains the calculation **Type** and is equipped with a drop down menu system for access to the various calculation types. For the basic event, BRK-AIR-FC-COMP, the calculation Type should be "1" (the default type). The four data entry boxes below the calculation Type are use to enter the various failure data depending on the calculation type used. For calculation Type 1, only the **Mean Failure Probability** value is required. From Table 1, enter the value, 1.8E-5 into the **Mean Failure Probability** box.

The right side of the **Modify Event Window** is where the **Uncertainty Data** is entered. This area contains uncertainty distribution **Type** and is equipped with a drop down menu for access to the various distribution types. For the basic event, BKR-AIR-FC-COMP, the uncertainty **Type** is "L: Log Normal." The entry boxes below the uncertainty data type are used to enter the various uncertainty data. From

Table 1, enter the value "5" into the first **Error Factor** box. A statistical **Correlation Class** number is not required for this example.

Click on the OK button when all the data is entered.

Repeat all the steps outlined above (Steps 1 through 5) 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.

2.4.2 Generating Basic Event Data to the Current Case

Once the base event rate information data is entered into SAPHIRE as base case data, it must be copied over (or transferred) to current case data. SAPHIRE only uses current case data for analysis. The **Generate** option is used to transfer base case data to current case data.

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 or make any event a template event, select the **Attributes** tab, located at the top of the **Modify Event** box. Move the cursor to the "Special Use Flags" area and select Template Event. This flag tells SAPHIRE that this basic event is a template event and rate information can be transferred to other basic events.

Move the cursor up to Template tab and click the left mouse key. The template characteristics that are to be applied to other basic events are checked. By default, all of the template event characteristics are check. To check or uncheck a already check box, click on the selection box. Once all desired characteristics are check, click on OK.

To apply the rate information of the newly created template event to other basic event, access **Modify → Basic Event** and highlight the basic event that is to have the data applied. Click the right mouse button and select Modify. Then click on the **Template** tab. Access the Template selection box with the drop down menu button. Select the desired template event name and click OK.

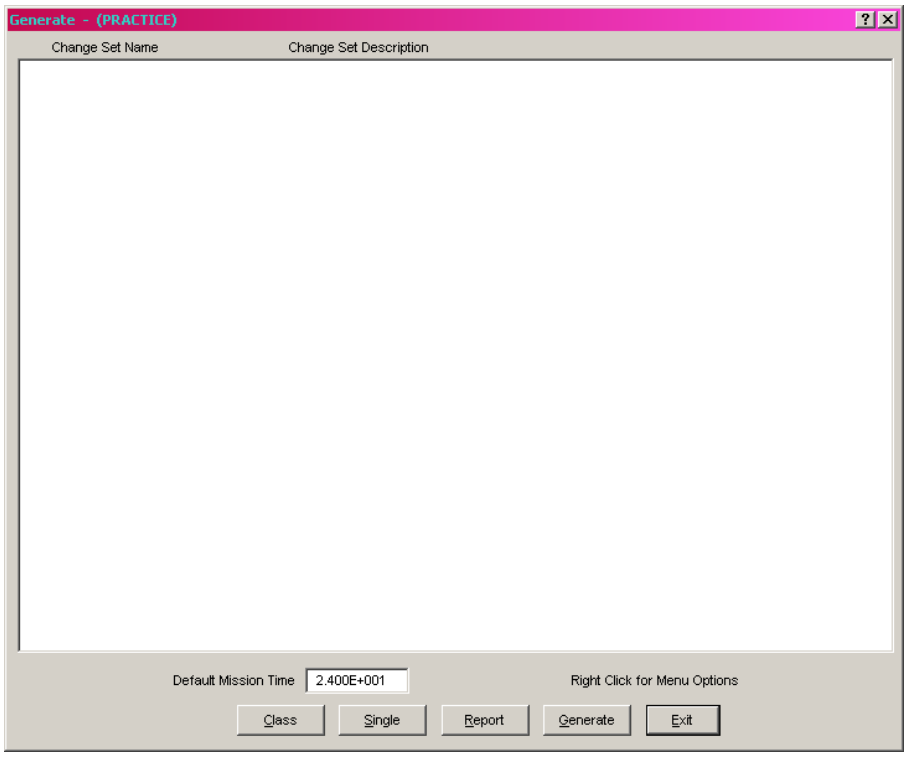
The rate information and attribute data shown, will then be applied (or copied over) to the basic event.

To generate event data, perform the following steps:

Select the Generate Button

from the graphical tool bar or select "Generate" from the menu bar.

The Generate screen, as shown below will be displayed. This screen is used to display and activate charge-sets. Currently, there are no change sets in the OBSTRUCT database.



Click on the Generate button. SAPHIRE will exit this screen automatically when completed.

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 in the form of metafiles 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 printed or exported to a graphical output file capable of being loaded into other software packages.

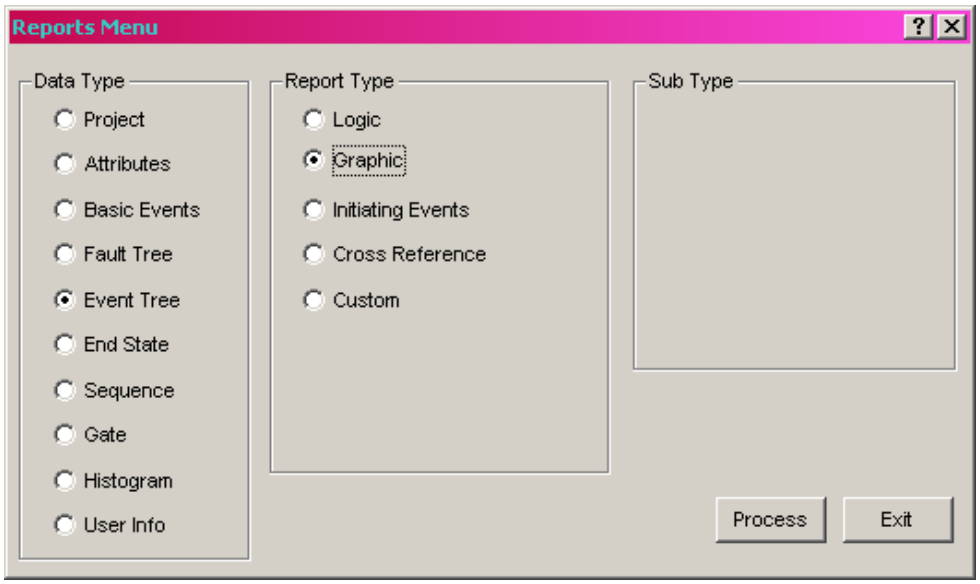
2.5.1 Printing Event Tree Graphics

Printing an event tree can be accomplished in two ways: 1) from the **event tree graphical editor** or 2) using the **Report** feature of SAPHIRE. To print an event tree from the **event tree graphical editor** (access the event tree editor by selecting **Event Tree** (highlighting the OBSTRUCT event tree and right click) → **Edit Graphics.** From the graphical tool bar, select the **Print** button . Select the desired printer from the print screen and click the **Print** button.

To print an event tree using the **Report** feature, perform the following steps:

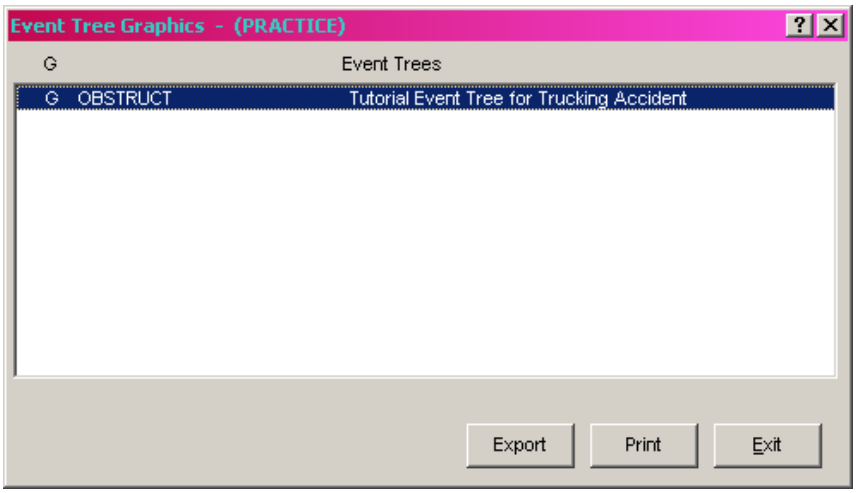
Click on the **Reports** button

located on the right side of the graphical tool bar (or click on Reports from the menu bar).



From the Reports Menu, click on **Event Tree** in the **Data Type** field and click on **Graphic** in the **Report Type** field. When complete, click on the **Process** button or the **Exit** button to exit without printing.

Select from the Event Tree Graphics window, the event tree OBSTRUCT and click on the **Print** button. To export as a Windows metafile, click on the **Export** button.



From the print window, select the desired printer and click on the **Print** button.

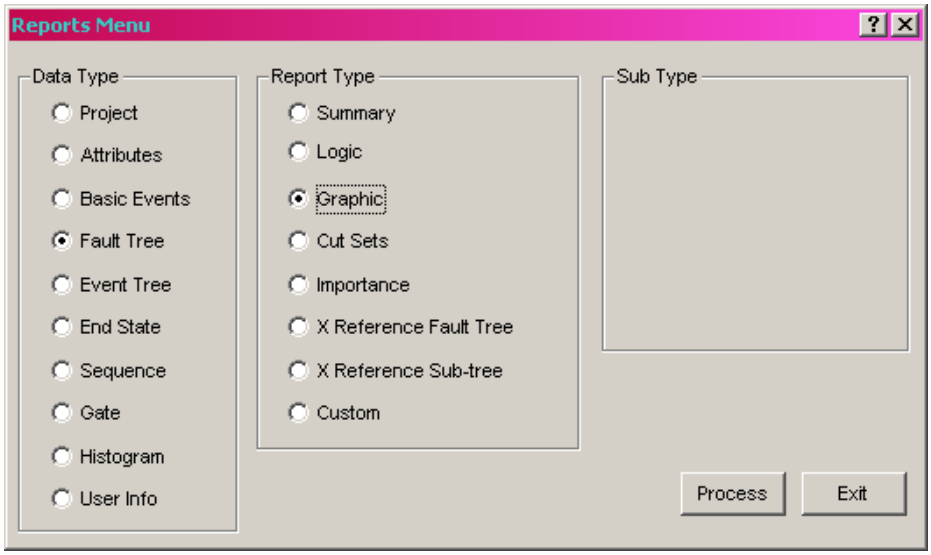
2.5.2 Printing Fault Tree Graphics

Printing a fault tree can be accomplished in two ways: 1) from the **fault tree graphical editor** or 2) using the **Report** feature of SAPHIRE. To print an fault tree from the **fault tree graphical editor** (access the event fault editor by selecting **Fault Tree** (highlighting the desired fault tree to print and right click) → **Edit Graphics.** From the graphical tool bar, select the **Print** button . Select the desired printer from the print screen and click the **Print** button.
To print a fault tree using the **Report** feature, perform the following steps:

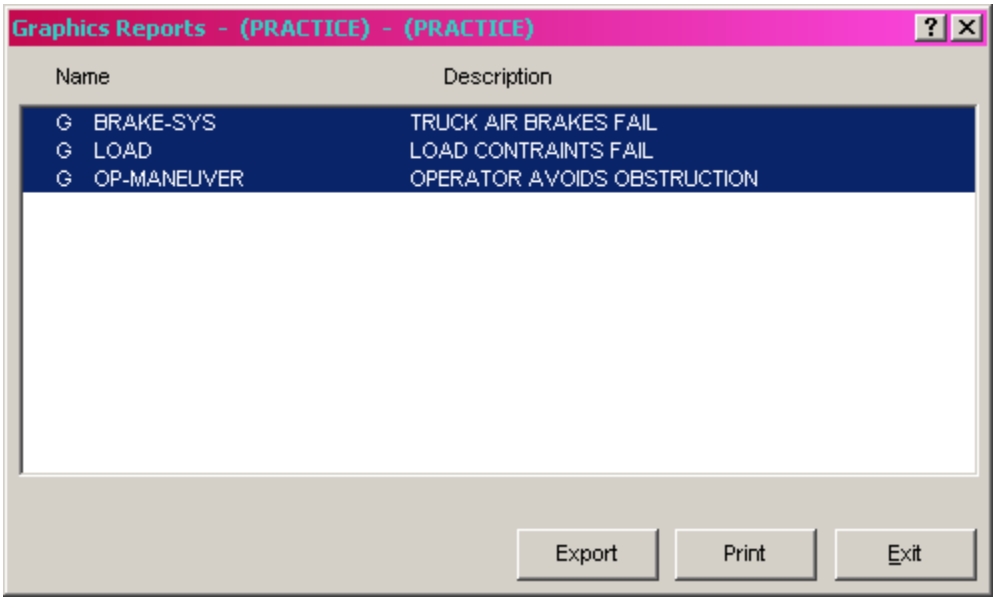
Click on the **Reports** button

located on the right side of the graphical tool bar (or click on **Reports** from the menu bar).

From the Reports Menu, click on **Fault Tree** in the **Data Type** field and click on **Graphic** in the **Report Type** field. When complete, click on the **Process** button or the **Exit** button to exit without printing.



Select from the Fault Tree Graphics window, the desired fault trees to be printed. To select all the fault trees, click on the BRAKE-SYS fault tree and hold down the left mouse button while dragging the mouse to the end of the list. Release the mouse button and click on the Print button located at the bottom the Graphics Reports Window. To export as a Windows metafile, click on the Export button.



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.

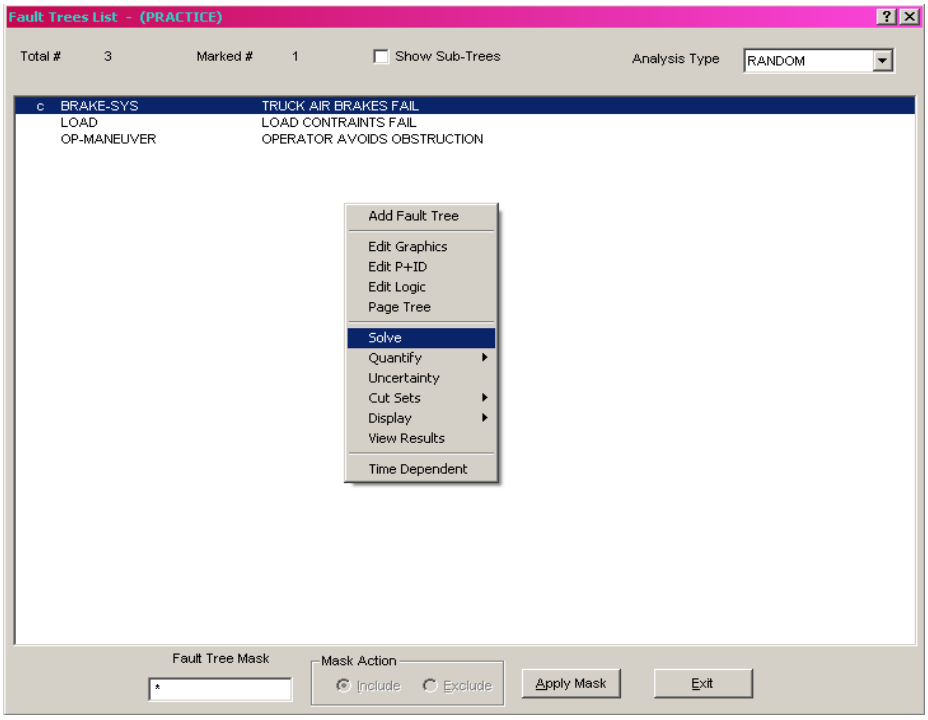
After the fault tree structure, or logic, has been developed and saved, fault tree cut sets can be produced and quantified. A cut set is a set of basic event combinations that, if they occur together, will cause a failure. A minimal cut set has no other cut set a subset. When generating fault tree cut 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 calculated 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 Fault Tree button from the Graphical Tool Bar.

Highlight the fault tree, BRAKE-SYS. Click the right mouse button to evoke the pop-up menu and move the highlight cursor down to **Solve**.



QUICK TIP

To highlight all fault trees in the **Fault Tree List** (to generate cut sets for all the listed fault trees), select the Apply Mask button. This button will highlight all the fault tree names in the list.

The **Cut Set Generation** window as shown in **Figure 16** will be displayed. If the "Cutoff by Cut Set Probability" box is check (default), then those cut sets with quantified value below the specified cutoff value (located in "Global Cutoff Value" entry box) will not be retained. It the box is not checked, then all cut sets will be retained. Click the OK button to accept the default cut set generation and truncation settings.

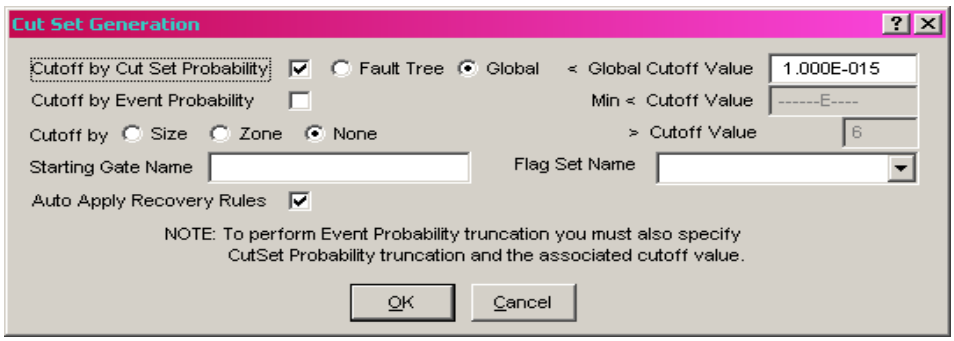


Figure 16 Cut Set Generation Screen

The Cut Set Generation Results screen will be displayed as shown below in Figure 17. This screen shows the number of cut sets generated and minimal cut set upper bound approximation result. Click on the OK button to continue.

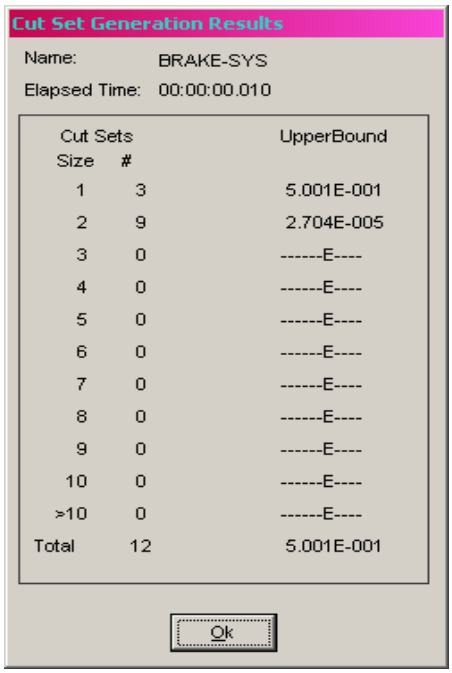


Figure 17 Cut Set Generation Results Screen

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.

*****NOTE***** The "c" located in from the fault tree name indicates that there are now "current case" cut sets for fault tree, BRAKE-SYS.

Click the right mouse button to evoke the pop-up menu and select **Display → Cut Sets** as shown in Figure **18**.

The cut sets generated for fault tree, BRAKE-SYS are now displayed as shown below in Figure 19. From this screen, details of each cut set and the minimal cut set upper bound approximation can be observed.

Select the **Exit** button when complete.

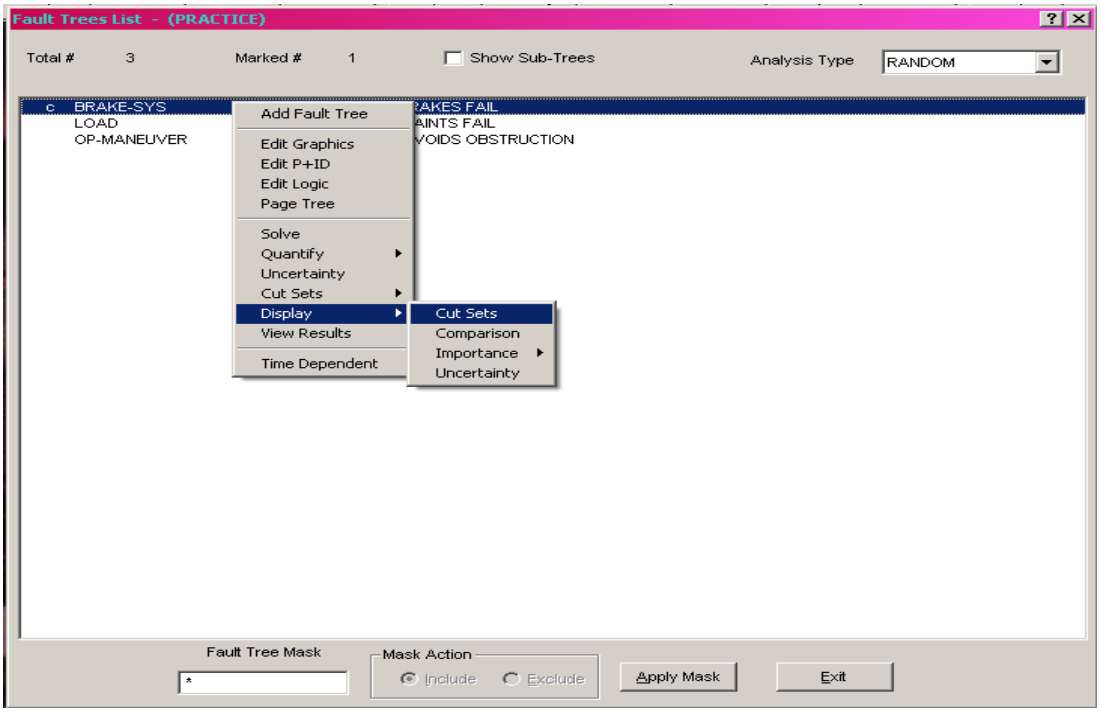


Figure 18 Displaying cut sets screen

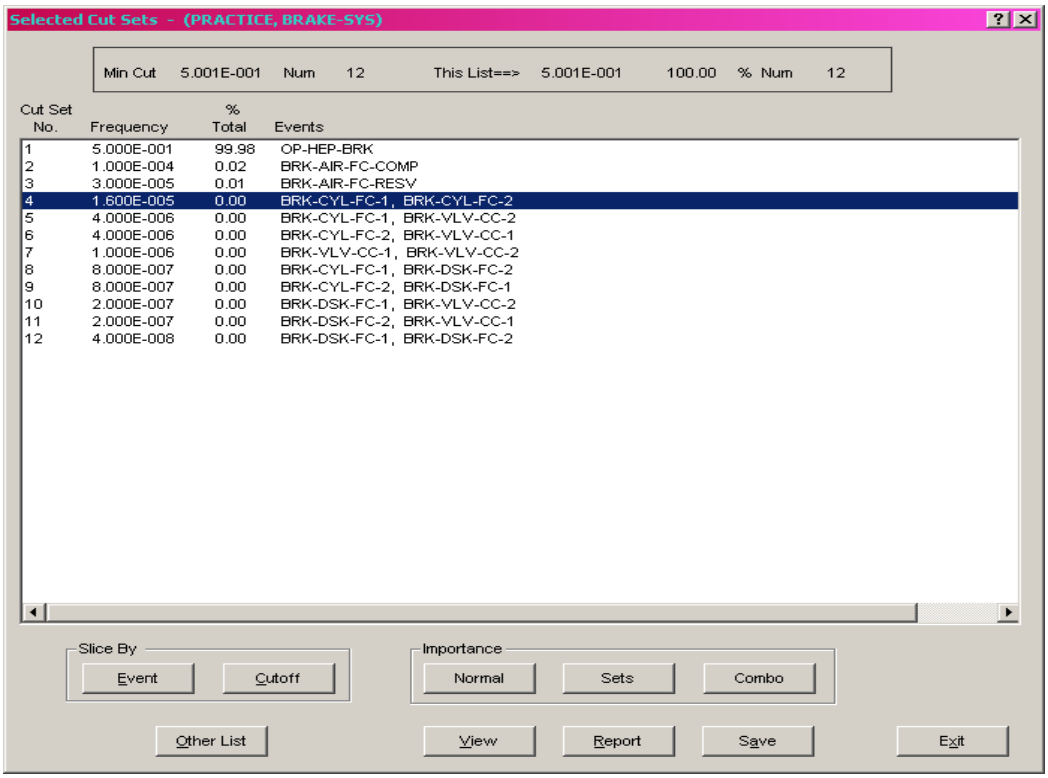


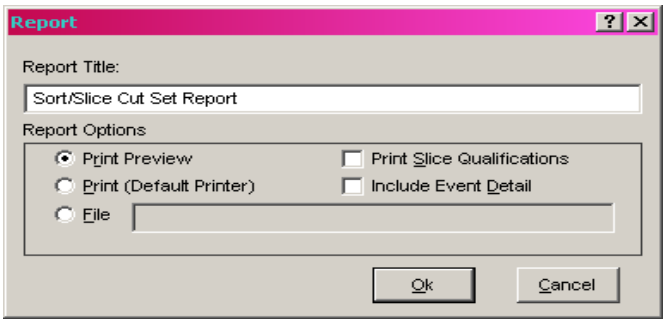
Figure 19 Selected Cut Set Viewer

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, perform the following steps:

From the **Cut Set Viewer** screen shown in Figure 2-16, click on the **Report** button.

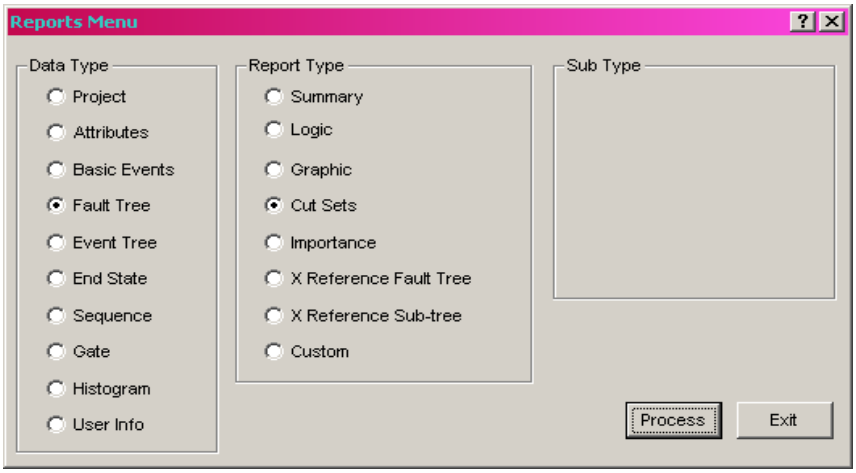
The **Sort/Slice Cut Set Report** screen is displayed as shown below.



Click on the **Print** radio button to send the report to a printer or click on the **File** radio button and type in a desired file name to print the report to file (in text form). Click on the Ok button to complete the report function.

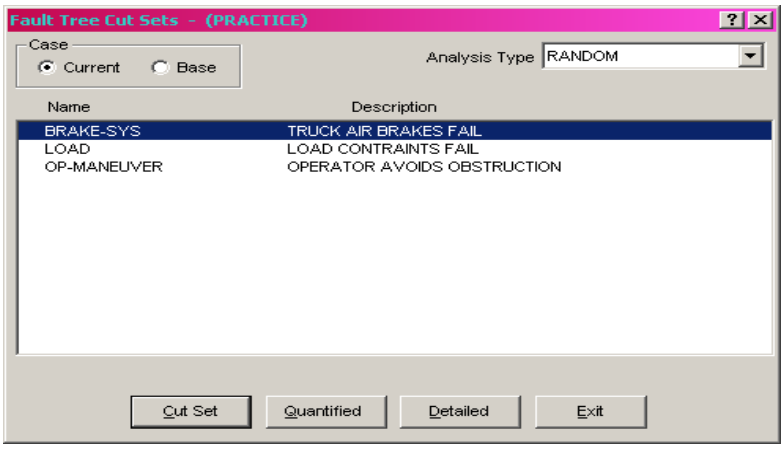
An additional option to obtain a cut set report is through the **Reports** feature of SAPHIRE. To obtain a fault tree cut set report through the Reports feature, click the **Reports** button

in the Graphical Tool Bar. Select from the radio buttons, **Fault Tree**, for Data Type and **Cut Sets** for Report Type as shown below.



Click the **Process** Button.

The list of available fault trees is displayed. Select the BRAKE-SYS fault tree and click the **Cut Set** Button.



The **Cut Set Report Option** screen is displayed. With this screen, the number of cut sets can be specified (i.e. Top 10 cut sets) and the ability to include basic event probabilities is available. If the default values are sufficient, click the **OK** button. Select **Print** to send the report to a printer.

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 (version 7 only)	T	Upper and lower endpoints

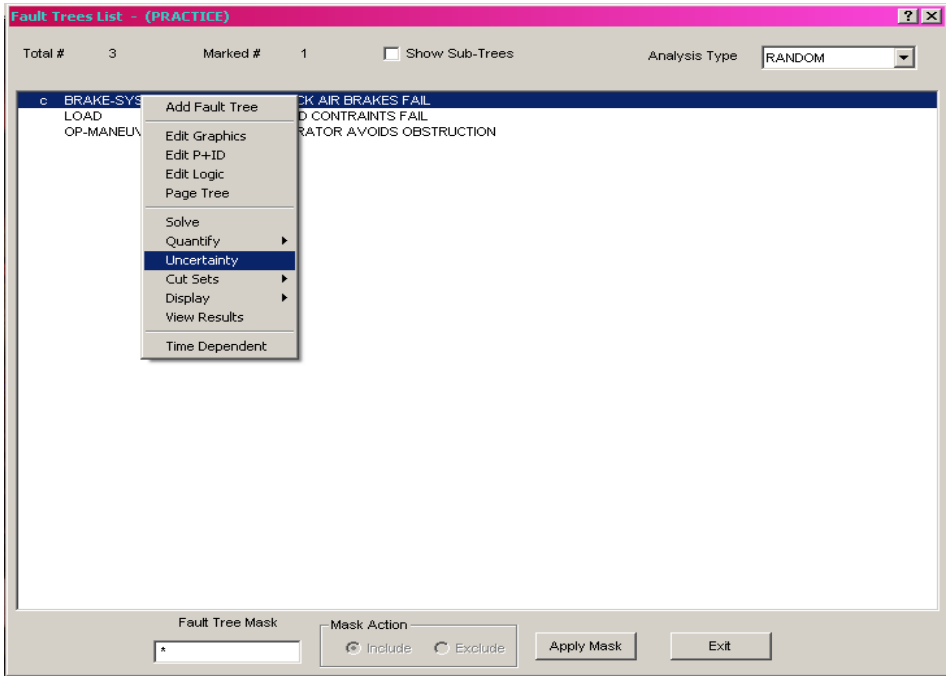
For the OBSTRUCT project, the basic events only implemented the "Lognormal" distribution. For additional information on the uncertainty analysis capabilities of SAPHIRE, please refer to the SAPHIRE Technical Reference Guide.

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:

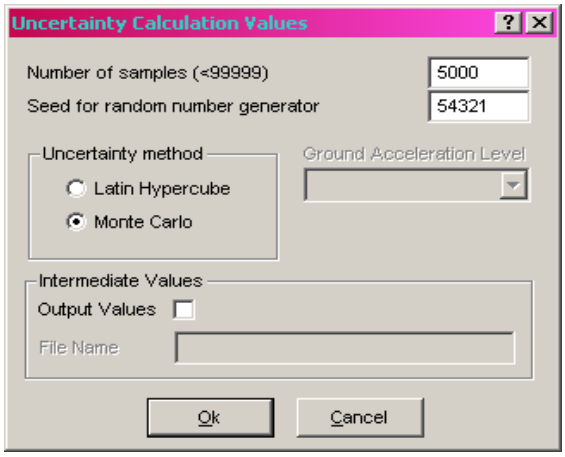
Select from the Graphical Tool Bar, the Fault Tree Button

.

From the Fault Tree List shown below, highlight the BRAKE-SYS fault tree and click the right mouse button to evoke the pop up menu. Move the mouse highlight cursor to 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.



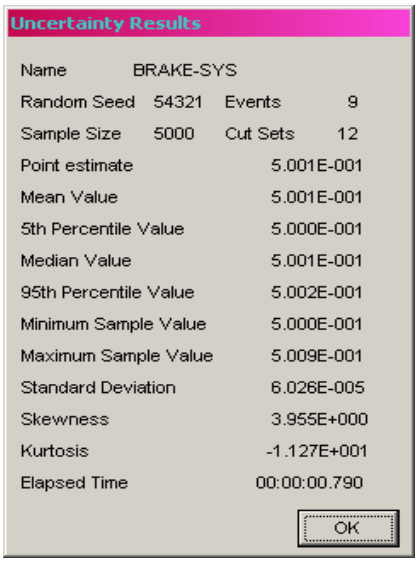
The **Uncertainty Calculation Values** screen will be displayed. Enter the uncertainty calculation values given below.



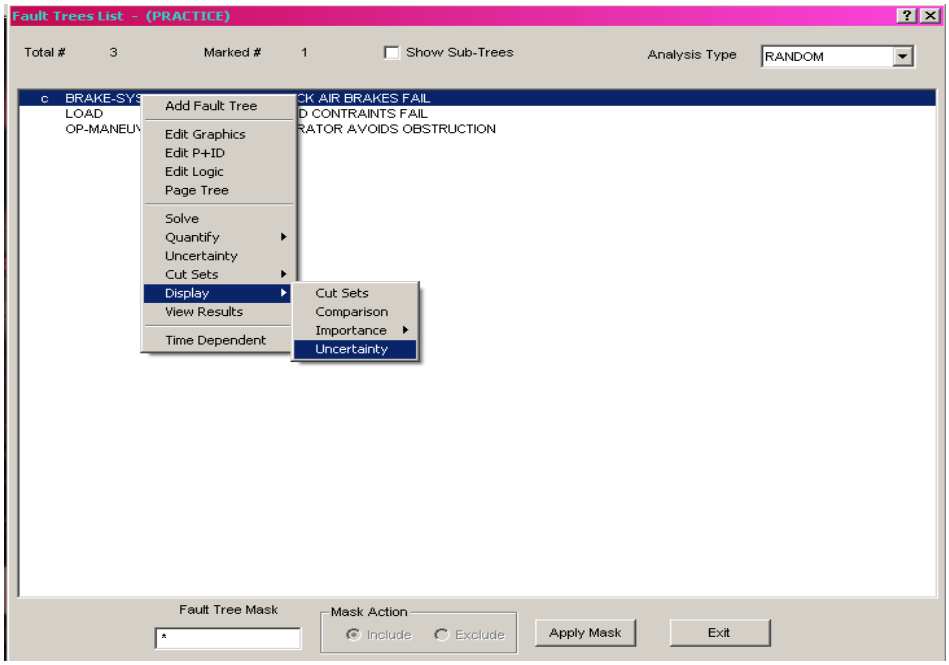
Number of samples (the larger the number, the more accurate the results) = 5000
Seed for random number generator = 54321
Uncertainty method = Monte Carlo (this is the default method)

Click the OK button when all the values are entered.

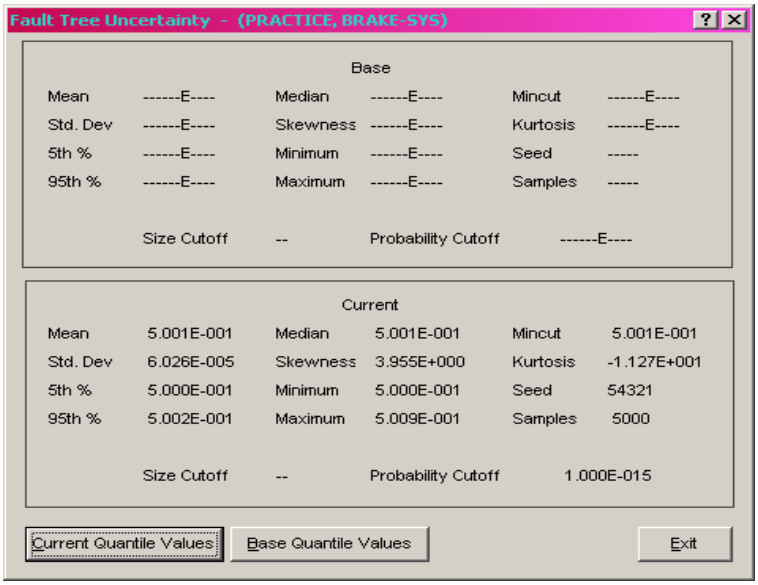
Uncertainty results will be displayed, as shown below, briefly on the computer screen following the uncertainty calculation.



Uncertainty results for each fault tree can also be displayed from selecting the **Fault Tree** button on the Graphical Tool Bar and then selecting **Display → Uncertainty**.



The System Uncertainty Screen appears as follows:



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

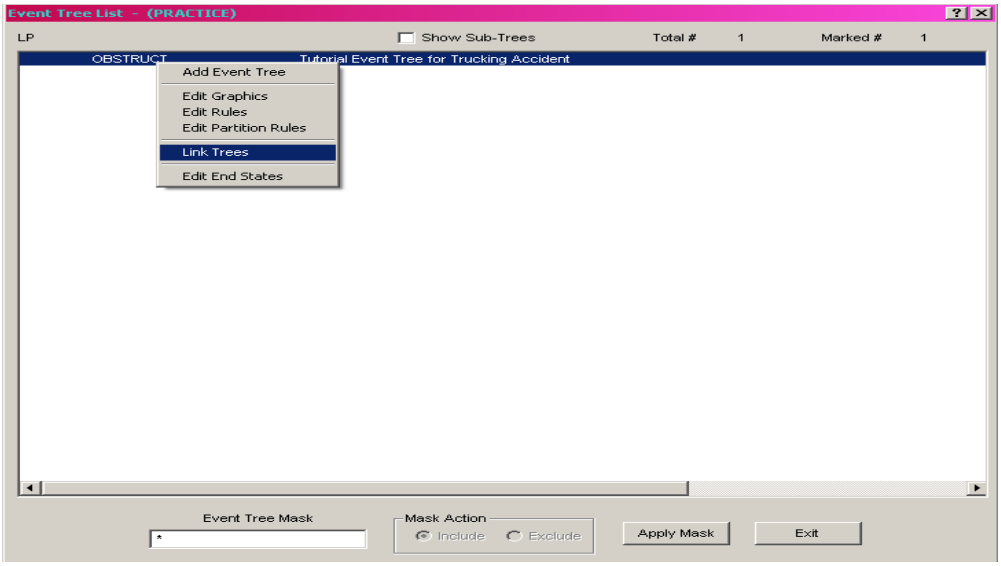
The term "linking" event trees is the process of generating sequences logic using the event tree graphical files and incorporating the top events of the event tree with the respected fault tree logic in the event tree sequences.

To link an event tree and generate sequence logic, perform the following steps:

Select the Event Tree Button from the Graphical Tool Bar

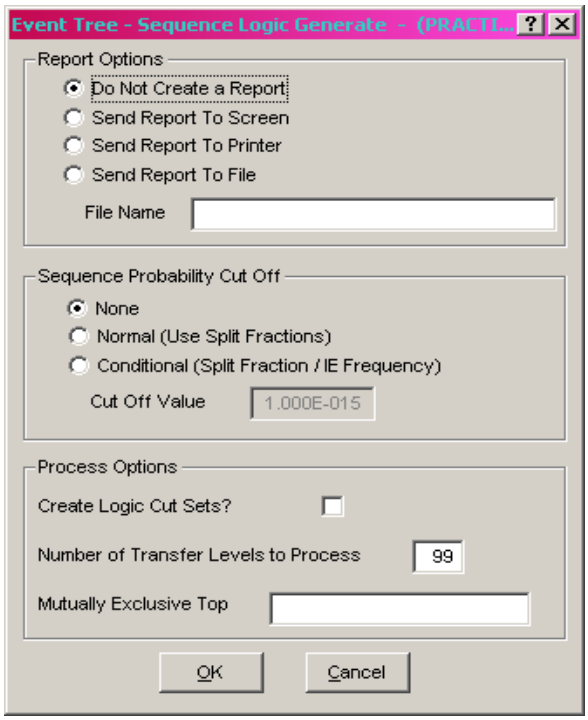
.

From the **Event Tree List Window**, highlight the OBSTRUCT event and **right-click** to invoke the pop-up menu. Select the **Link Trees** option.



The Event Tree – Sequence Logic Generate screen is displayed. This screen allows the user to view or send a sequence logic report to the screen, printer, or file. The default setting is set at "Do Not Create a Report". If the user wishes to see the generated sequence logic, select the "Send Report to Screen" radio button.

Click the OK button to link the OBSTRUCT event tree and generate sequence logic.



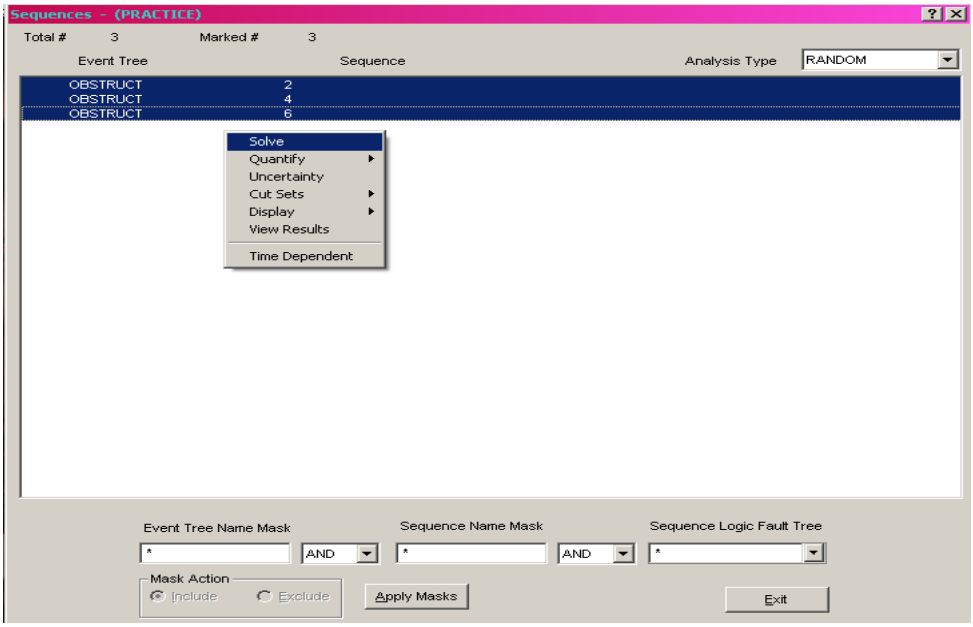
2.8.2 STEP 2: Solve For Sequence Cut Sets

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 **Sequence** button

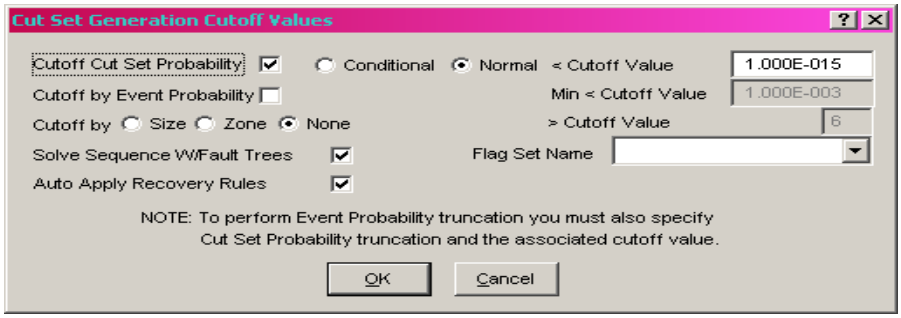
from the Graphical Tool Bar. A listing of all the sequences generated during the linking process will be displayed in **Sequences** screen.

Highlight all the sequences by either selecting the Apply Masks button at the bottom of the Sequences screen or selecting the first sequence and dragging the mouse highlight cursor down the sequence list until all three sequences are marked (or highlighted).

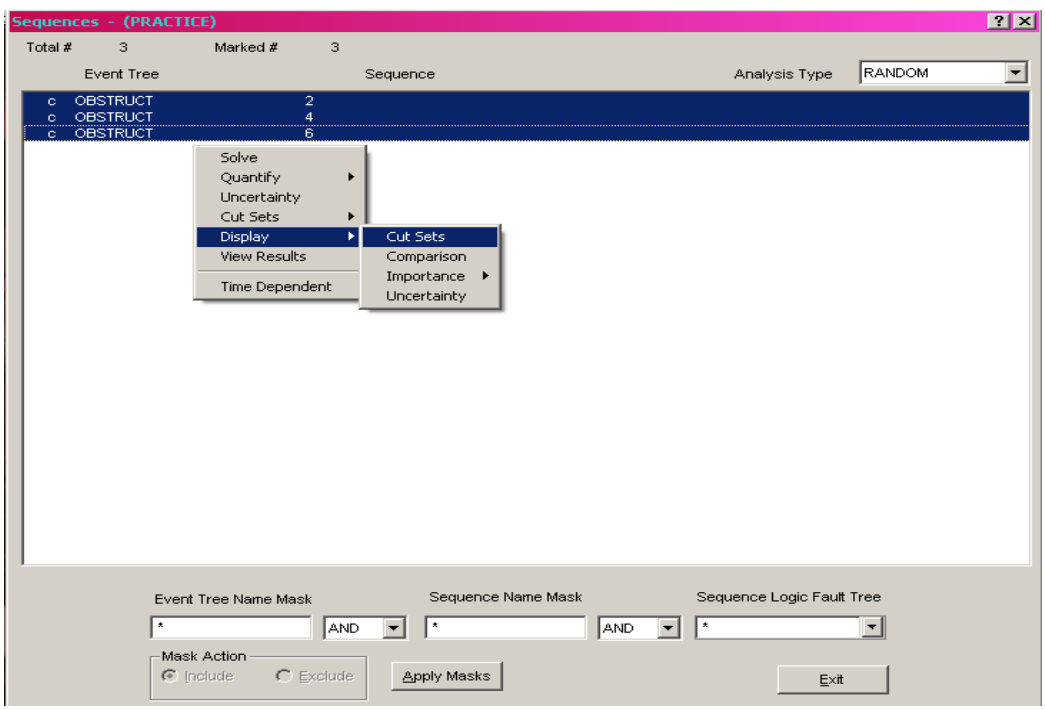


Right click to invoke the pop-up menu and select the **Solve** option.

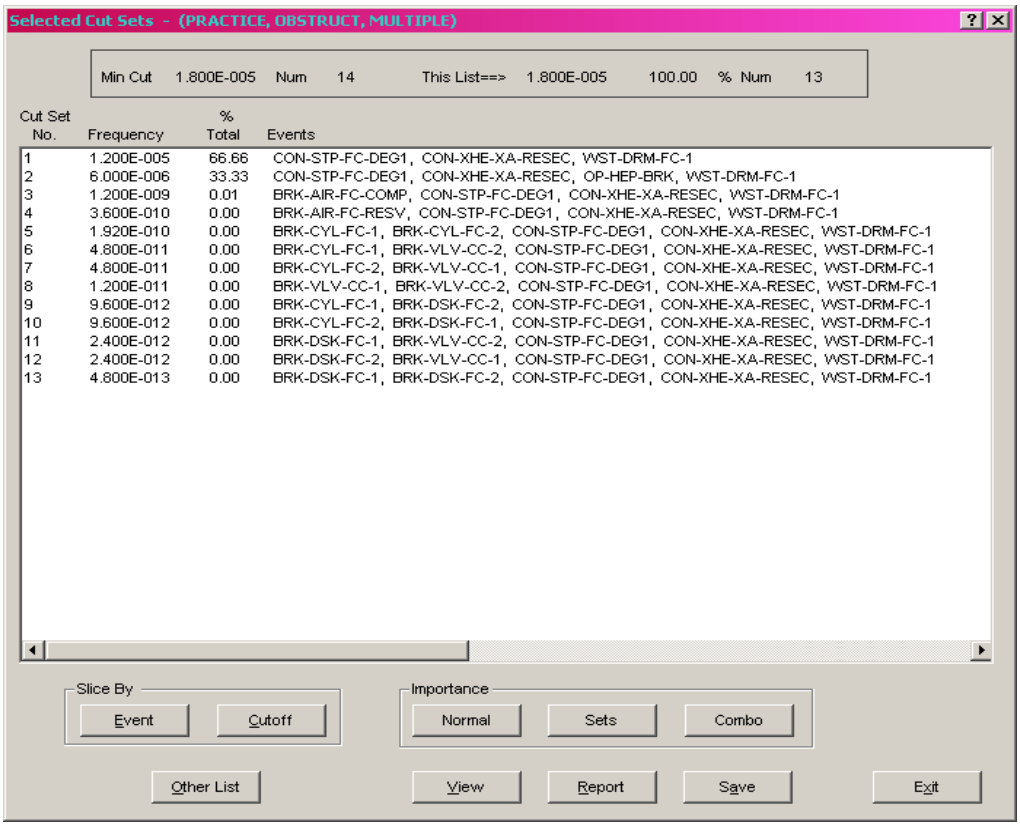
The **Cut Set Generation** window as shown below will be displayed. If the "Cutoff Cut Set Probability" box is check (default), then those cut sets with quantified value below the specified cutoff value (located in "Cutoff Value" entry box) will not be retained. It the box is not checked, then all cut sets will be retained. Click the OK button to accept the default cut set generation and truncation settings.



After the sequence cut sets have been generated, SAPHIRE returns to the **Sequences** screen. All three sequences should still be highlighted. If not, click on the Apply Masks button to highlight all sequences. To display the highlighted sequence cut sets, right click to invoke the pop-up menu. Select the **Display → Cut Sets** option as shown below.

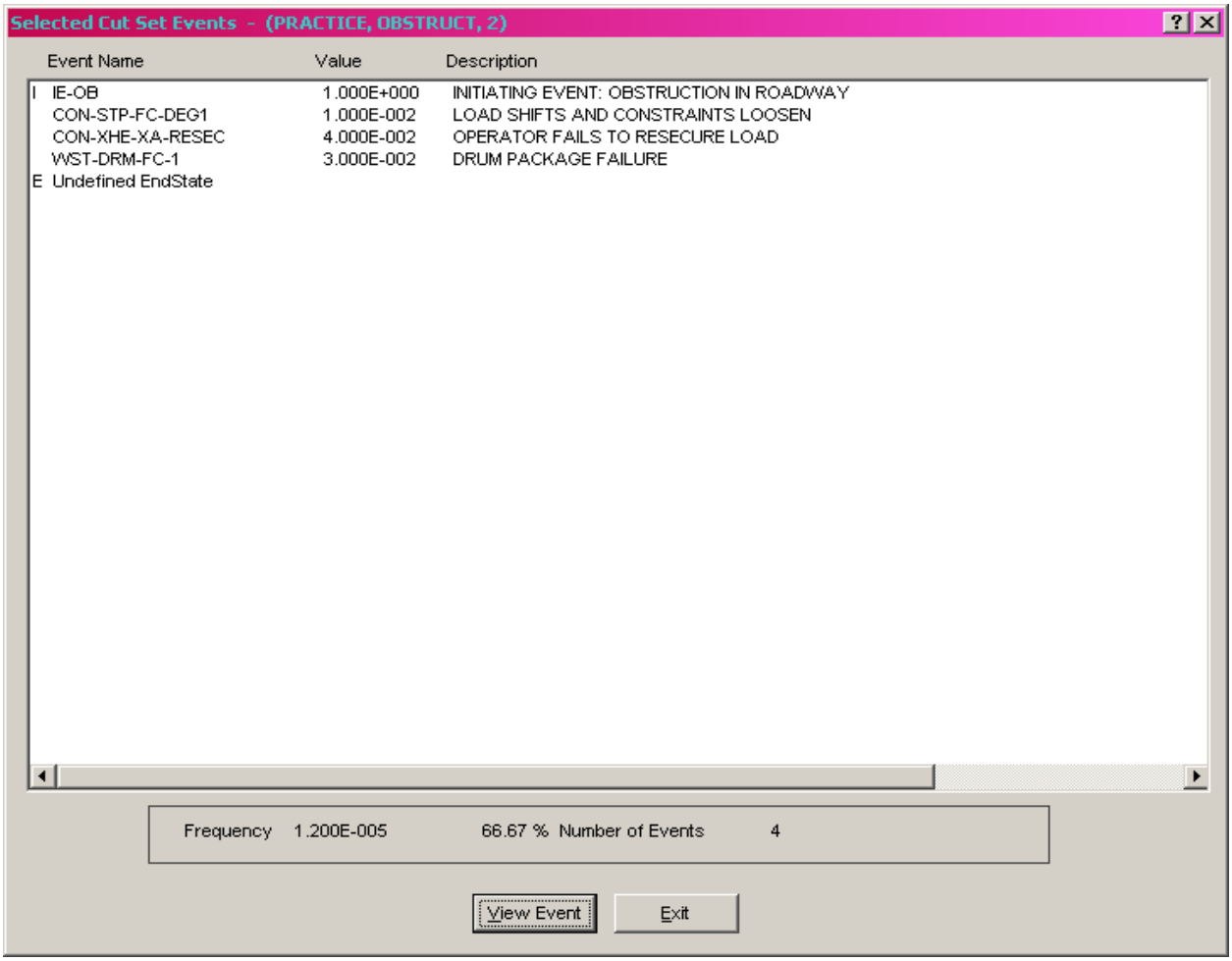


The Selected Cut Set screen is displayed as shown below.



To view the basic events and their respected failure probability, highlight the desired cut set (for example, highlight the first cut set) and click the View button

.



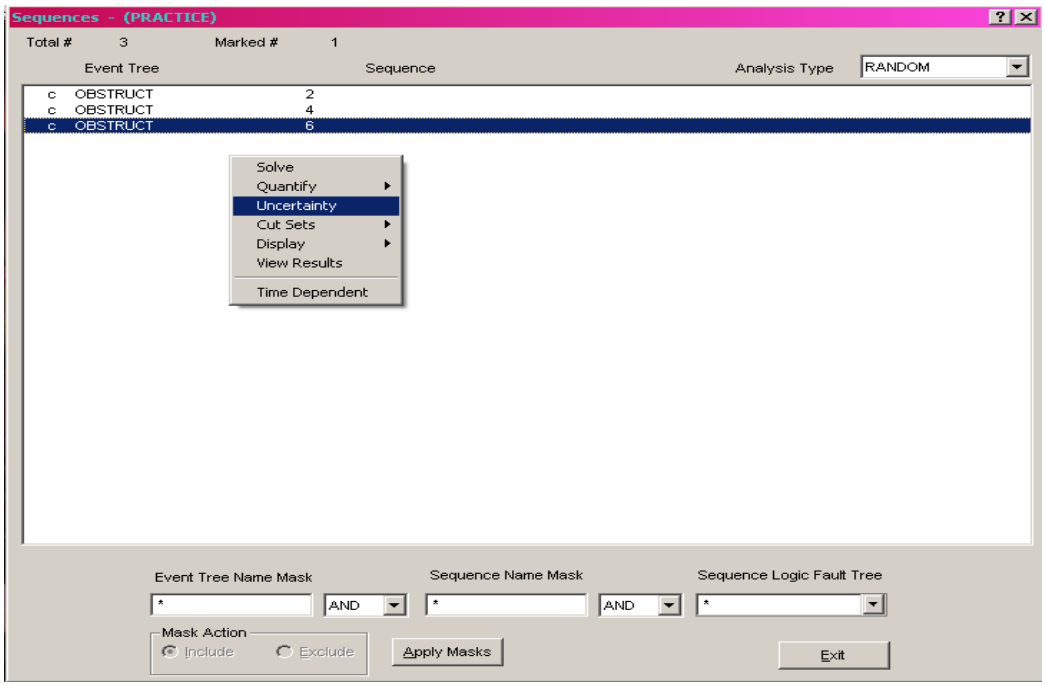
2.8.3 STEP 3: Performing Sequence Uncertainty Analysis

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:

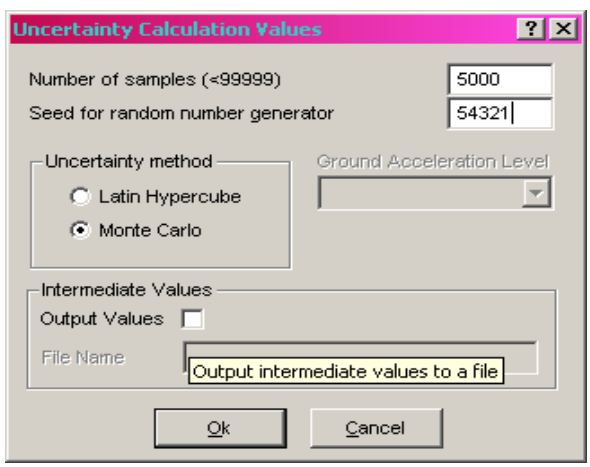
Select **Sequence** from the Graphical Tool Bar if the **Sequence** list screen is not shown.

Highlight the sequence OBSTRUCT 6. Right click to invoke the pop-up menu. Select Uncertainty as shown below.



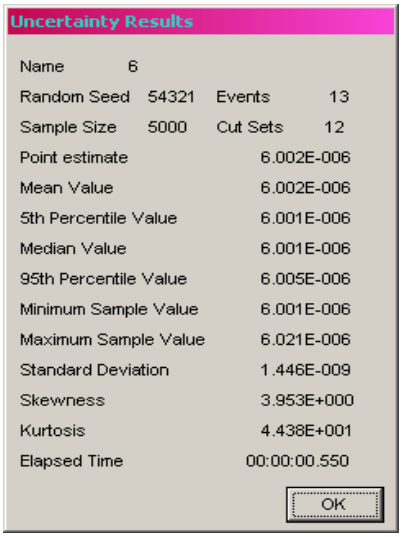
The **Uncertainty Calculation Values** screen will be displayed. Enter the uncertainty calculation valves given below.

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



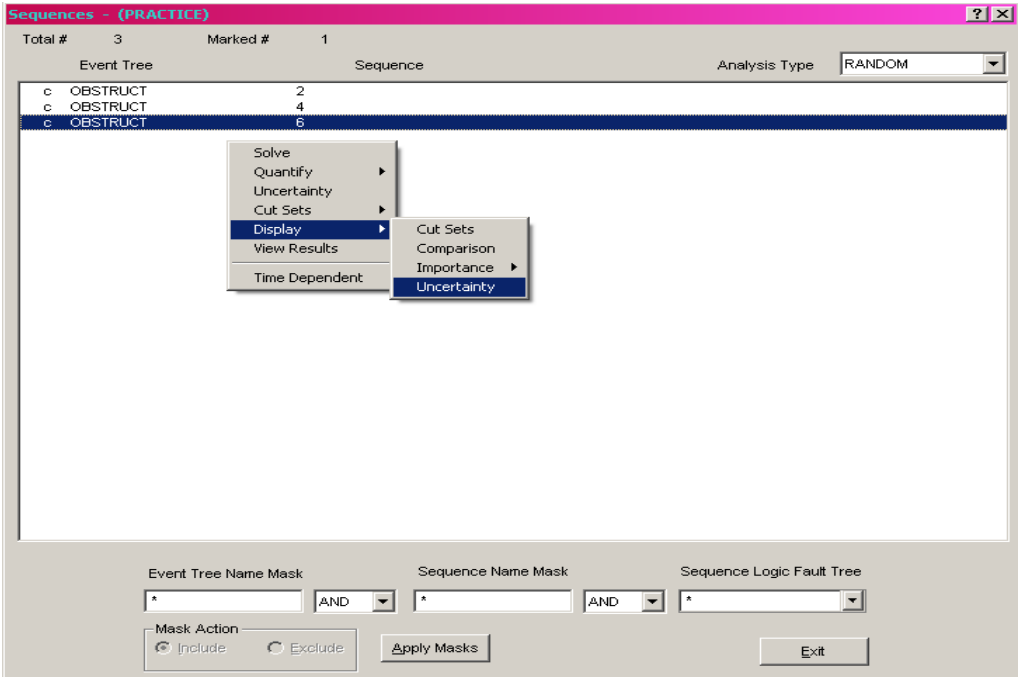
Click the Ok button when all the values are entered.

Uncertainty results will be displayed, as shown below, briefly on the computer screen following the completion of the uncertainty calculation.



Click the **Ok** button to continue.

Uncertainty results can also be displayed by highlighting sequence #6 and right clicking to invoke the pop-up menu. Select Display → Uncertainty to display the uncertainty results.



Click on **Exit** to leave the **Sequence Uncertainty** screen.

Click on **Exit** to leave the **Sequence** list screen.

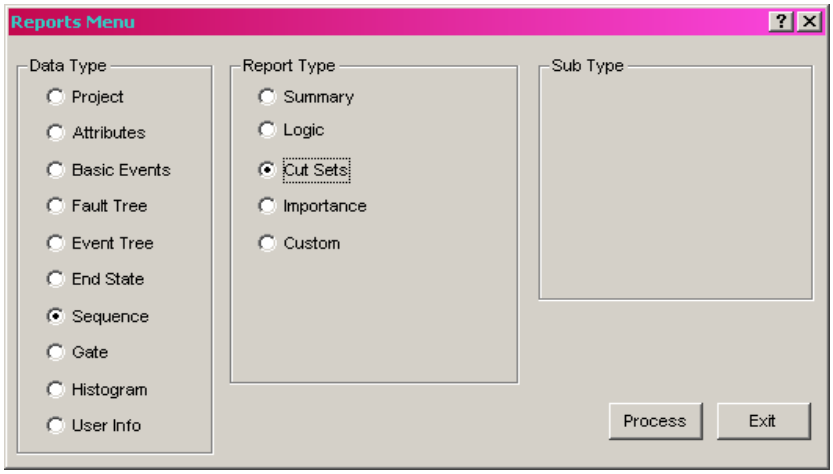
2.8.4 Reporting Sequence Cut Set Results

To produce a report of the sequence cut set results, perform the following steps:

Select Reports

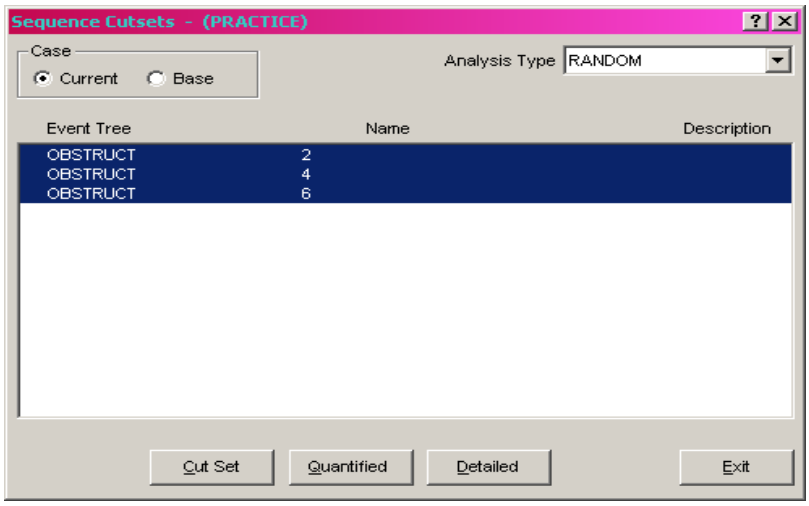
from the Graphical Tool Bar or the Menu Bar.

Select for the Data Type, the radio button located in front of the available option, **Sequences**. For Report Type, select **Cut Sets**, as shown below.



Click on the **Process** button to continue.

The Sequence Cut Sets screen will be displayed as shown below. Highlight all the sequences by left clicking the first sequence and dragging the mouse down the list of sequences until all are highlighted.



Click on the Cut Set button to continue.

Click on OK to accept the default Cut Set Report Options.

Click on radio button located in front of **Print (Default Printer)** and click on the Ok button to send the report to a printer.

(This page intentionally left blank)

3. ADVANCED LESSONS

The purpose of the Advanced Lessons section is to familiarize the user with some of advanced features and operation of SAPHIRE. 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 Change Sets and Flag Sets
This lesson describes the difference between a flag sets and a change sets and guides the user on how to implement 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. Recovery Rules
Lesson 12 introduces the user the "Recovery Rule" editor of SAPHIRE. The recovery rule editor allows you to create rules that affect existing cut sets in a "post-processing" manner.

3.1 Lesson 10 – Creating Change Sets and Flag Sets

In lesson 10, the user will learn the difference between change sets and flag sets and why they are used in SAPHIRE. An example change set will be created in the project PRACTICE and applied to specific basic events and the data re-generated.

Change sets are user-defined set of changes that will be applied to the base case data when event data is transferred to the current case data. Therefore, change sets adjust the base case data to a new user specified value and then when the data is "generated", it moves the new value into current case to be used in an analysis. Change sets can be used to change probability values and basic event calculation types, uncertainty parameters, and basic event attributes.

Flag sets are a special type of change set and can only contain either house flag changes to the calculation type of 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 Change Set

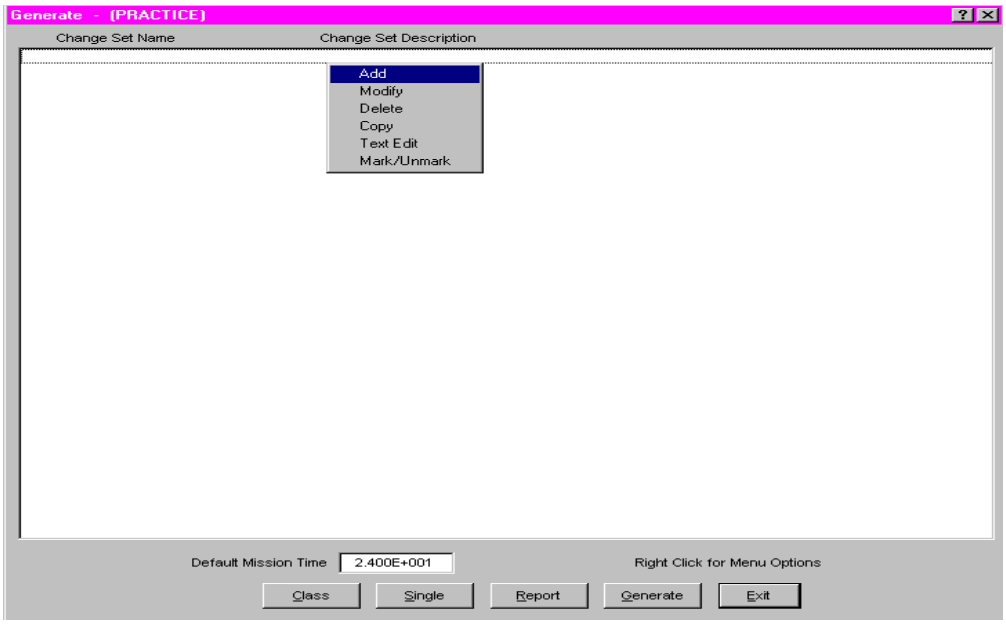
Creating a change set is performed in SAPHIRE under the Generate menu option. There is no limit to number of change sets that can be added to the database. Each change set must have a name and the name given is limited to 24 characters. There are two types of change sets: class change sets and individual change sets. Class change sets use a basic event attribute to search through the basic event list in the database for a class of basic events to which the defined change applies. Individual change sets only modify a user identified basic event. Once the basic event is selected, the user specifies the changes to be applied to the basic event. Within a change set, both class changes and individual changes can be applied. However, individual changes will overwrite the class change.

The following example will create a simple change set called "CHANGE1" that changes the basic event probability of BRK-AIR-FC-COMP from 1.0E-4 to 1.0E-1. The change set will then be applied to database and fault tree BRAKE-SYS will be re-quantified with the new value.

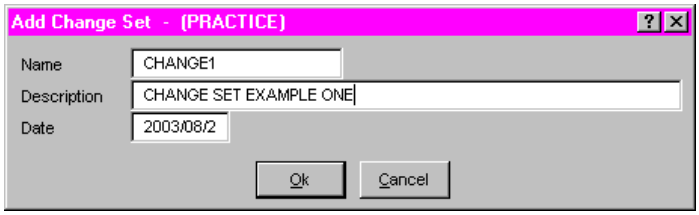
STEP 1: Creating and Naming a Change Set
To create a change set in SAPHIRE, perform the following steps:

Select the **Generate** button from the Graphical Tool Bar.

The **Generate** screen will be displayed as shown below. The screen list any change sets in the database (currently, there are no change sets in the project). Place the mouse cursor anywhere in the **Generate** screen and click the right mouse button to invoke the pop-up menu. Select **Add.**



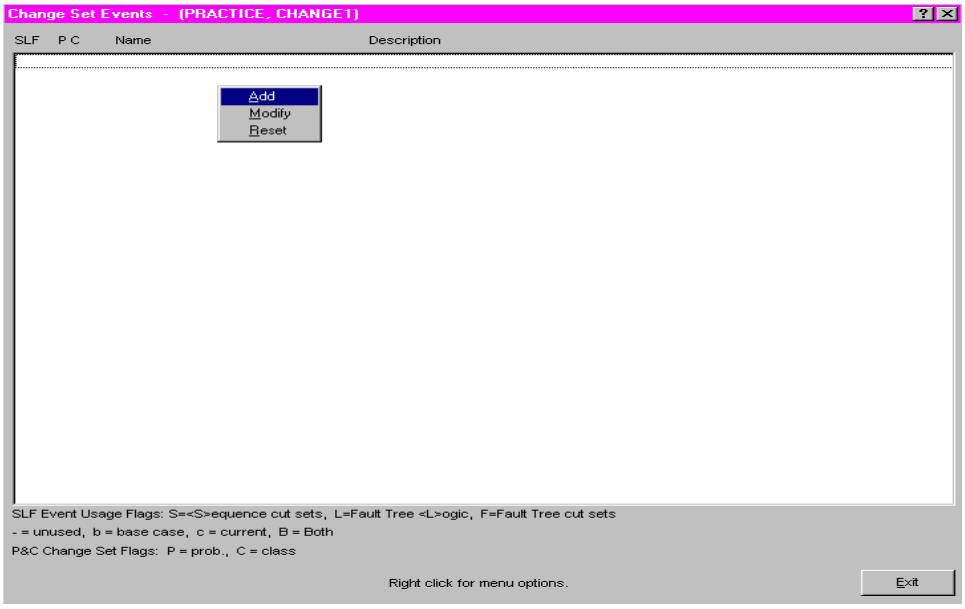
The **Add Change Set screen** appears as below. Type in the Name Box, CHANGE1 and type in the Description Box, Change Set Example One. Click on **Ok** when completed.



STEP 2: Modifying the Data in the Change Set
To modify the basic event failure and/or uncertainty data in the change set, perform the following steps:

The new change set, CHANGE1 has been added to the **Generate** screen. Select the change set CHANGE1 by clicking on with the left mouse button.

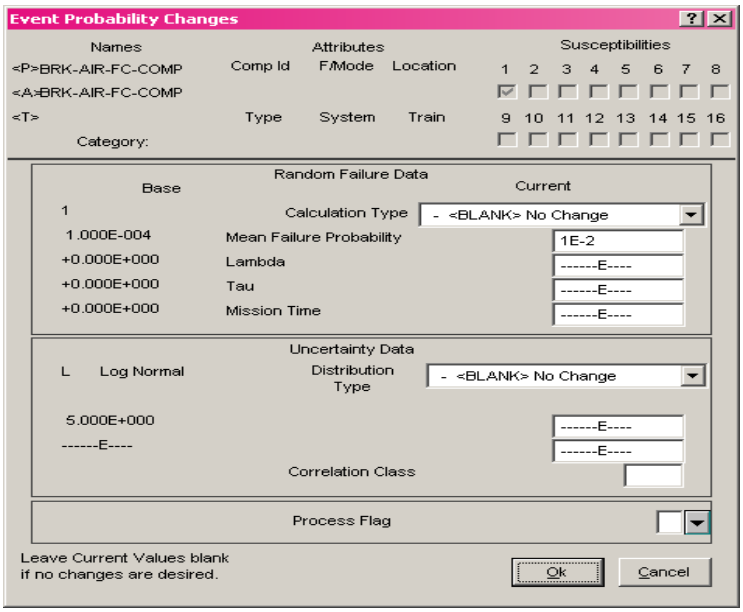
CHANGE1 will be an "individual" change set. Click on the **Single** Button. The Change Set Event screen will be displayed as below. No basic event exist in the change set therefore, the screen is blank. To add basic event to be changed, click the right mouse button to invoke the pop-up menu and select **Add**.



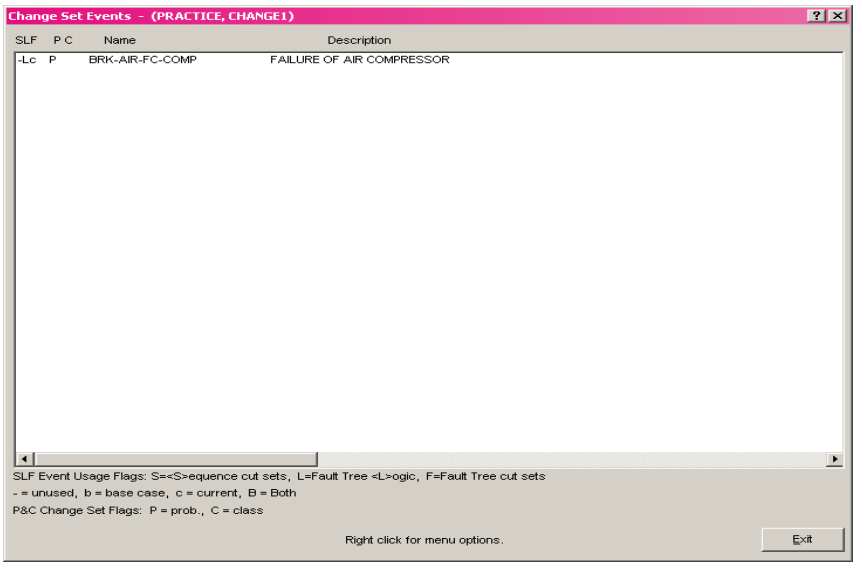
The **Select Change Set Event** screen is displayed. This lists includes all the basic events in the project. Select the basic event, BRK-AIR-FC-COMP and right click to invoke the pop-up menu. Select **Add** from the pop-up menu.

The Event Probability screen is displayed as shown below. The left side of the screen, under the heading **Random Failure Data** indicates the base case data for the selected event, BRK-AIR-FC-COMP. The right side of the screen contains the entry boxes for current case data that will be implemented after the data is generated.

For the **Mean Failure Probability**, type in the new value of 1.0E-2 in the data entry box as shown below. Click the **OK** button when completed.



The Change Set Events screen now has one basic event, BRK-AIR-FC-COMP, as shown below.



Click on the Exit button

.

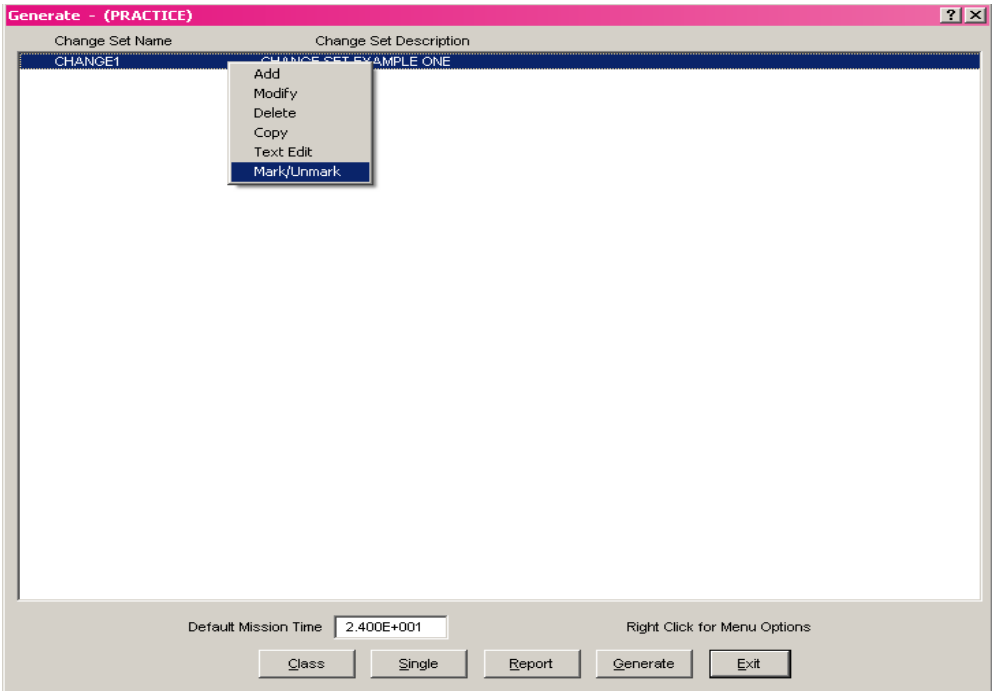
STEP 3: Marking the Change Set to be Used by SAPHIRE

Once the basic event BRK-AIR-FC-COMP has been added to the change set and rate information adjusted accordingly, the Generate screen will now have one change (the CHANGE1 change set) listed as shown below. However, the change set will NOT be implemented by SAPHIRE unless it is marked (i.e.

initialized). Marking a change set tells SAPHIRE to implement the data changes in the change set and copy the changed data into the current case data to be used for analysis.

To MARK a change set, perform the following steps:

Highlight the CHANGE1 change set and click the right mouse button to invoke the pop-up menu. Move cursor to Mark/Unmark to mark the change set. A change set can also be marked by *double-clicking* the left mouse button on the change set name in the **Generate** screen.



Once the change set, CHANGE1 is successfully marked, a number 1 is located in from of the change set name. This marking informs the user that the change set has been marked and will be the first change set implemented.

To invoke the change set and therefore moving the data changes into the current case, click on the Generate

button.

After the Generate button is activated, the data is moved into the current case. The data is now ready to re-analyze with the new data.

STEP 3: Analyzing with the Marked Change Set
To re-analyze the fault tree, BRAKE-SYS with the change set, CHANGE1 marked, perform the following steps. See Lesson 6 for a more detailed review of analyzing fault trees.

> **QUICK TIP**
>
> SAPHIRE is capable of implementing multiple change sets during the **Generate** process. However, the order of "marking" the change sets is important. The first change set selected, which will be marked by a number "1" will be applied first during Generate. If a second change set is marked, it will be marked with a number "2". This change set will be applied second and override any changes in the first change set if they happened to change the same basic event.

Select the Fault Tree button from the Graphical Tool Bar.

Highlight the fault tree, BRAKE-SYS and click the right mouse button to invoke the pop-up menu. **Select Solve**.

The Cut Set Generation screen is displayed. Accept the default truncation levels by clicking on the **Ok** button.

The Cut Set Generation Results screen is displayed. The new results show an increase to 5.05E-1 for calculated fault tree failure probability. Click on the **Ok** button to exit the Cut Set Generation Results screen.

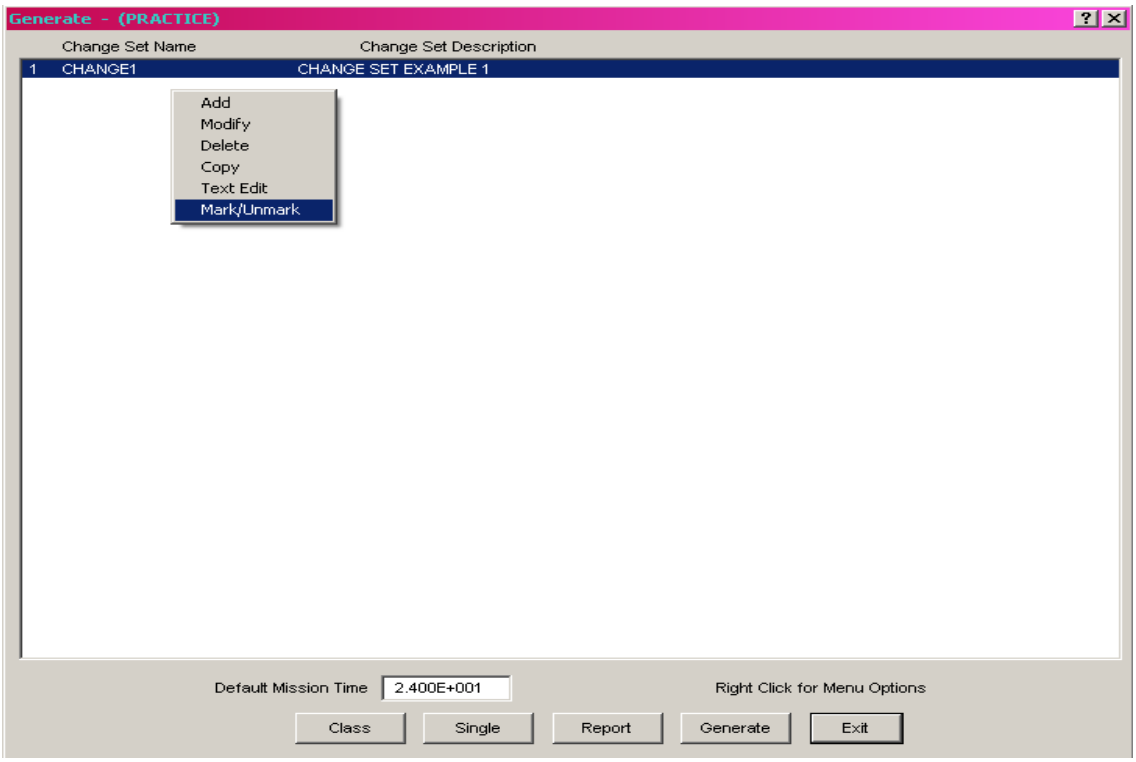
STEP 4: Unmarking the Change Set
The changes to the base case data in the current case will remain in effect until the Change Set is unmarked and the data regenerated. This takes the original data of base case and moves it back into the current case.

To unmark (or deactivate) a change set, perform the following steps:

Select the Generate

Button from the Graphical Tool Bar.

Highlight the Change Set, CHANGE1 and right click to invoke the pop-up menu and click on Mark/Unmark. This action de-selects the change set. [*A second method to unmark a change set simple to double click on the marked change set.*]



Click on the **Generate** button located at the bottom of the screen to move the original base case data back to the current case.

3.1.2 Creating a Flag Set

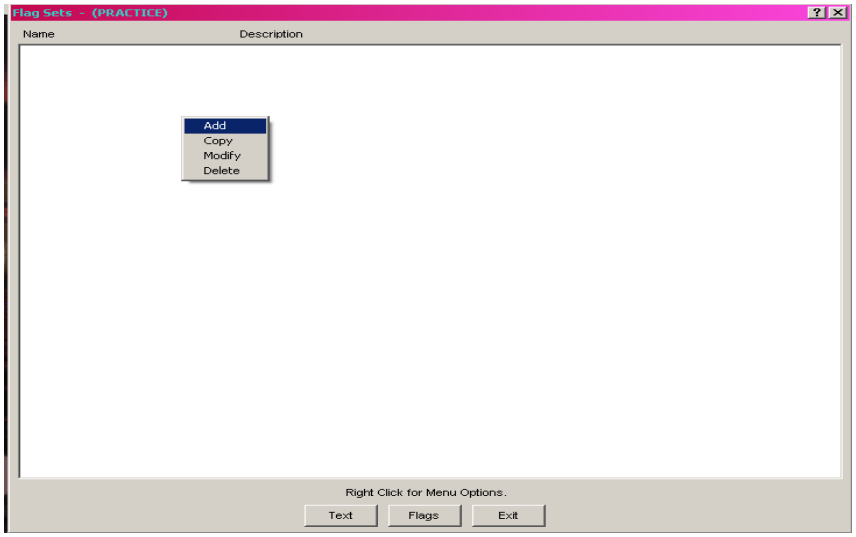
Even though a flag set is a special type of change set, flag sets are not created in the Generate Menu. Flag sets are created in **Modify** menu option and are applied to the database by directly "attaching" them to fault trees or event tree sequences.

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.

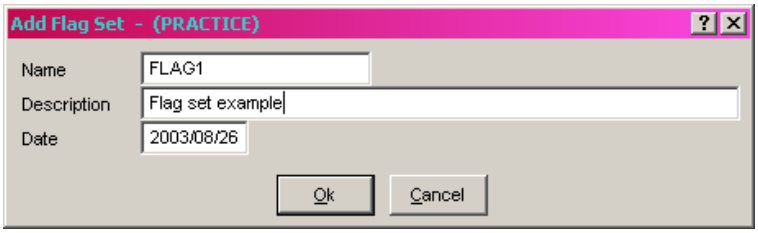
STEP 1: Creating and Naming a Flag Set
To create a flag set in SAPHIRE, perform the following steps:

Select the **Modify** button from the Graphical Tool Bar, then select the **Flags** button (or from the Menu Bar Select **Modify → Flags**).

In the **Flag Set** screen as shown below, click the right mouse button to invoke the pop-up menu. Select **Add**.



In the Add Flag Set screen, enter in the **Name** box FLAG1. A description is not required.



Click the **Ok** button to add the flag set.

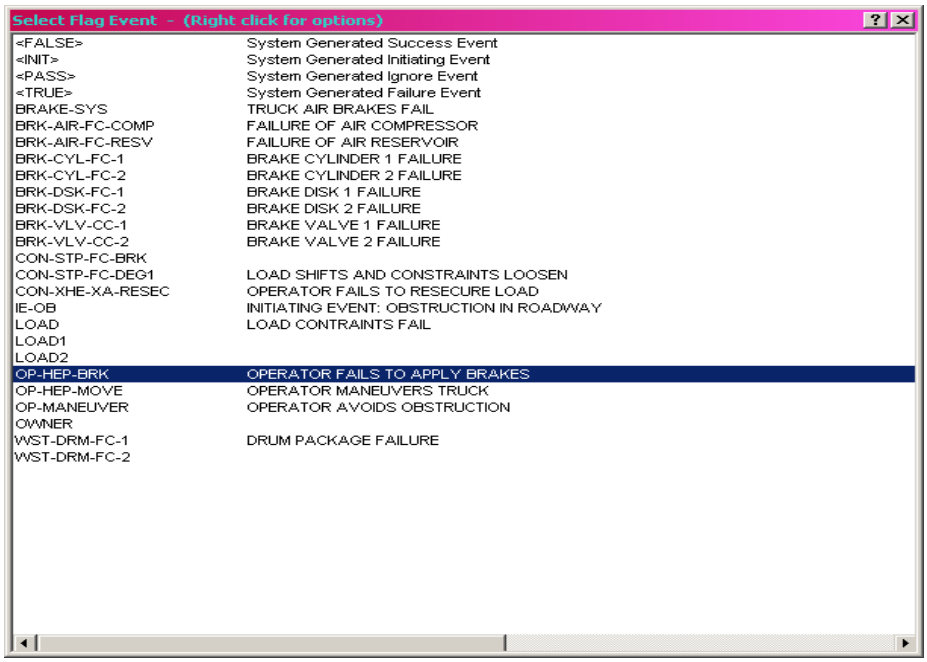
STEP 2: Modifying the Calculation Type in the Flag Set
To modify the basic event Calculation Type in the flag set, perform the following steps:

In the **Flag Set** screen, as shown below, **highlight** the flag set, FLAG1 and click the Flags

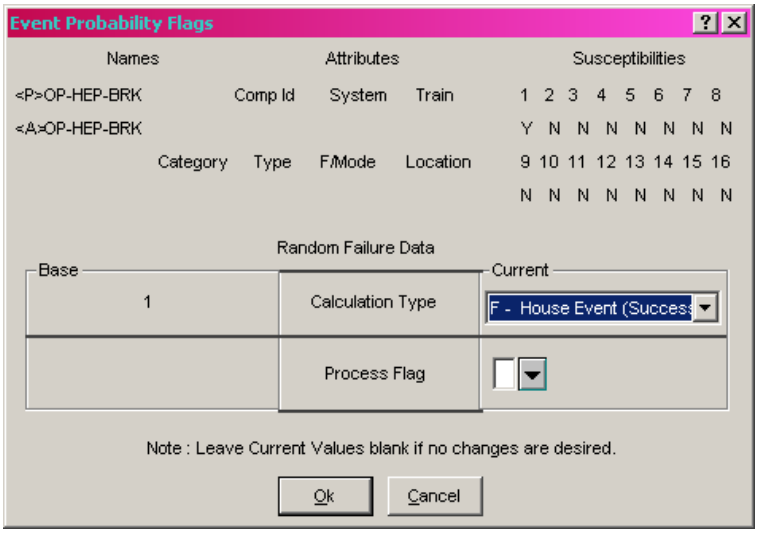
button located at the bottom of the **Flag Set** screen.

The Flag Set Events screen is displayed. To add events to the flag set that will be modified, right-click to invoke the pop-up menu and select **Add.**

The **"Select Flag Event"** screen is displayed as shown below. Select the basic event, OP-HEP-BRK by double clicking on the basic event or right clicking and selecting **Add** from the pop-up menu.



The **"Event Probability Flags"** screen is displayed as below. The flag set data screen allows you to enter changes on the right side of the screen. Note that only the Calculation Type or a Process Flag can be modified in Flag Set. Under the **Current** case drop down menu, select "F- House Event (Success)" and click on the **Ok** button.



Click on the **Exit** button to leave the "**Flag Set Events**" screen.

Click on the **Exit** button to leave the "**Flag Sets**" screen. The Flag Set, FLAG1 is now ready to use in the database. Click on the Return button to return to main SAPHIRE menu screen.

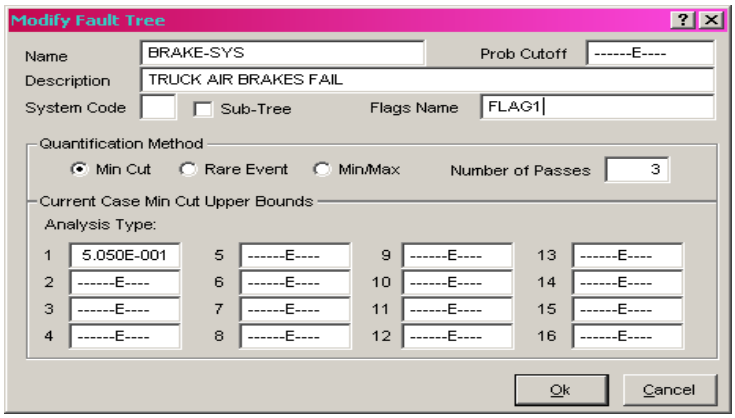
STEP 3: Using the Flag Set

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:

Click on the **Modify** button and then click on the **Fault Tree** button (or select **Modify → Fault Trees** from the menu bar.

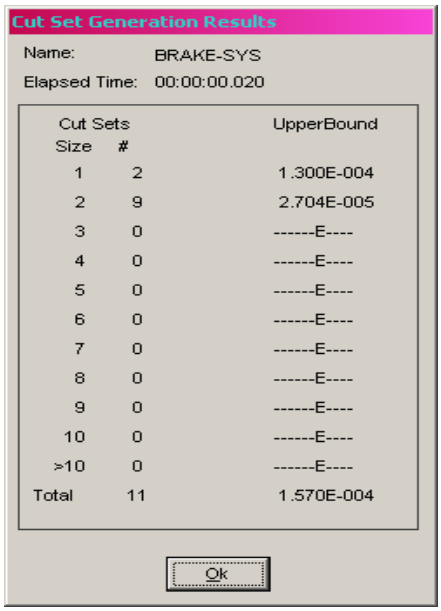
Highlight the BRAKE-SYS fault tree and right click to invoke the pop-up menu. Select **Modify**.

The **Modify Fault Tree** Screen appears as below. To apply the Flag Set, type in the Flag Set name ("FLAG1") in the box field labeled **Flags Name**.



Click the **Ok** button when complete. The Flag Set is now assigned to the fault tree BRAKE-SYS and will be applied when the fault tree permanently until it is removed from the "Flags Name" box.

To solve the fault tree, BRAKE-SYS with the flag set applied, select **Fault Tree** <highlight the BRAKE-SYS fault tree and right click> **→ Solve** <click Ok to accept the default truncation values> The new results with the operator action set to FALSE is shown below at 1.57E.



Cut Set Generation Results

Name: BRAKE-SYS

Elapsed Time: 00:00:00.020

Cut Sets Size	#	UpperBound
1	2	1.300E-004
2	9	2.704E-005
3	0	------E----
4	0	------E----
5	0	------E----
6	0	------E----
7	0	------E----
8	0	------E----
9	0	------E----
10	0	------E----
>10	0	------E----
Total	11	1.570E-004

Ok

STEP 4: Removing the Flag Set

To remove the flag set from the fault tree, BRAKE SYSTEM:

Click on the **Modify** button and then click on the **Fault Tree** button (or select **Modify → Fault Trees** from the menu bar.

Highlight the BRAKE-SYS fault tree and right click to invoke the pop-up menu. Select **Modify**.

The **Modify Fault Tree** Screen appears as below. To remove the Flag Set, delete the flag set name (FLAG1) in the box field labeled **Flags Name**.

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:

Table 2. SAPHIRE Linking Rule Search Criteria Examples

Search Criteria	**Example**	**Meaning of Search Criteria**
A	If A then	If "A" occurs then do this…
/A	If /A then	If "A" does not occur then do this…
~A	If ~A then	If "A" is not present then do this…
A * B	If A * B then	If "A" and "B" occurs then do this…
A + B	If A + B then	If "A" or "B" occurs then do this…
(A + B)* C	If (A + B) * C then	I "A" or "B" occurs and C occurs then do this…
always	If always then	The search criteria are always met so do this…

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 rule 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 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
```

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).

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 20 illustrates the new fault tree to be substituted in the event tree OBSTRUCT. Create the following fault tree in SAPHIRE by performing the following steps:

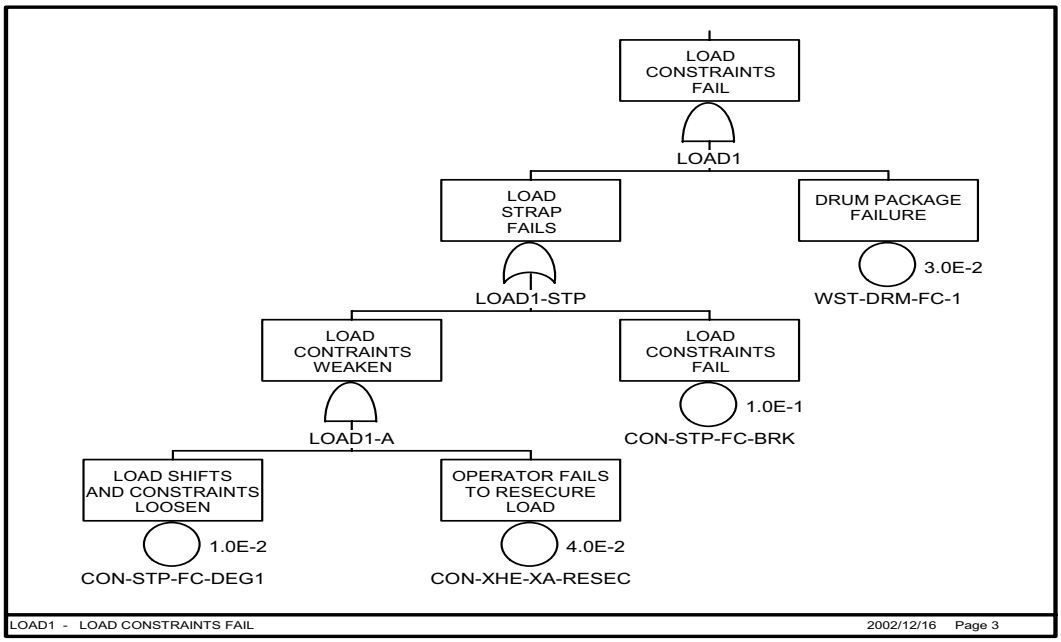
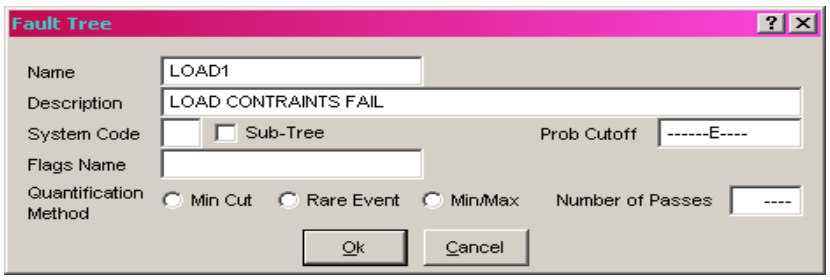


Figure 20 LOAD1 Substitution Fault Tree

Access the Fault Tree Graphical editor by clicking on the **Fault Tree** button and then right clicking to invoke the pop-up menu. Select **Add Fault Tree**.

In the **Fault Tree** screen, type in the name of new fault, LOAD1 as shown below. Click the **Ok** button when complete.



Highlight the new fault tree, LOAD1 and right click to invoke the pop-up menu. Select Edit Graphics. Create the fault tree shown in Figure 20 using the graphical editor. Click the **Save** button and exit the graphical editor.

Click the **Exit** button to leave the Fault Tree List Screen.

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.

Go to **Modify → Basic Events** and add this data to the basic event CON-STP-FC-BRK. 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-MANEUVE is successful.

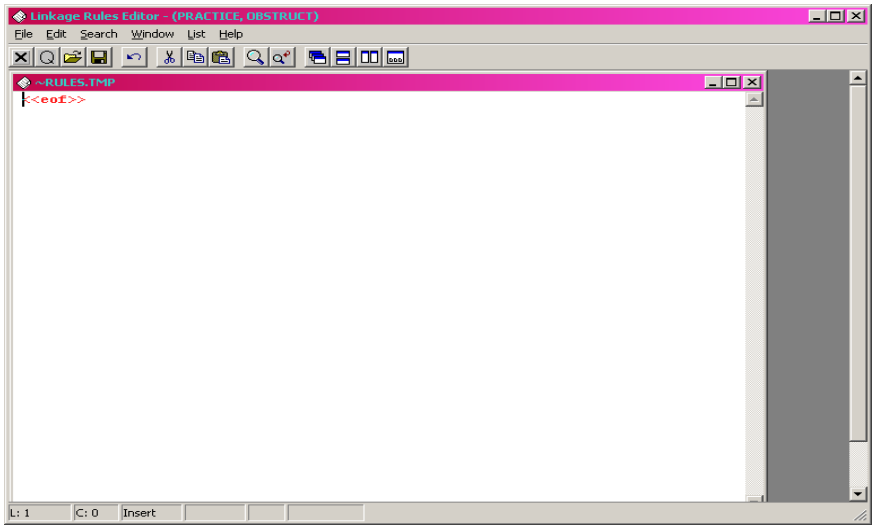
Linking rules are entered into SAPHIRE through the **Event Tree** menu button

. To write a rule that accomplishes the desired substitution, perform the following steps:

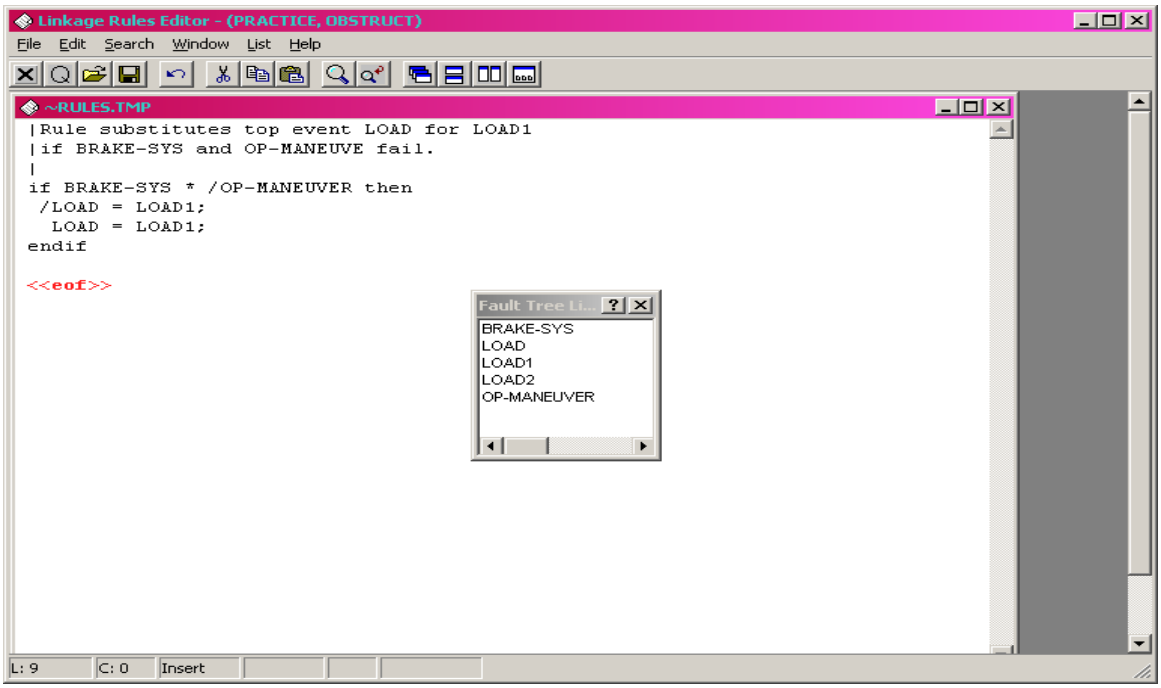
Select the Event Tree button in the Graphical Tool Bar.

Highlight the OBSTRUCT event tree and right click to invoke the pop-up menu. Highlight **Edit Rules**.

The Linkage Rule Editor is displayed as below.



The substitution requires a rule that replaces LOAD with LOAD1 when BRAKE-SYS and OP-MANEUVE both fail (i.e. down braches on the event tree). To accomplish this, type in the following rule in the rule editor shown below.



Note: The **Fault Tree List** box shown above is obtained by clicking on **List** in the Linkage Rule Editor and then **Systems (Fault Trees).**

To exit the linkage rule editor, click on the **Exit** button and click on **Yes** to save the linkage rule. This now a "+" symbol next to event tree name, OBSTRUCT. This indicates to the user that there are linkage rules associated with the event that have successfully been compiled (a "-" sign would indicate that there are rules associated with the event tree but, they were not successfully compiled).

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 logic sequences generated. To link the event and generate new sequence logic, perform the following steps (for more detail on linking, see Lesson 8):

Select the Event Tree Button from the Graphical Tool Bar.

From the Event Tree List Window, highlight the OBSTRUCT event and right-click to invoke the pop-up menu. Select Link Trees option.

The Event Tree – Sequence Logic Generate Screen is displayed. To view the new logic and confirm that the substitution took place, click the radio button in Report Options in front of Send Report to Screen. The following report will be displayed.

```
Report Viewer - SAPHTEMP.000
File   Edit   Options   Help
 Sequence Name          !       Fault Tree            !        Fault Tree          !
 ______________________ !_____________________________ !____________________________
Event Tree Name: OBSTRUCT
6                       ! BRAKE-SYS                   !  OP-MANEUVER              !
                        ! LOAD                        !                           !
   end state: RELEASE
4                       ! BRAKE-SYS                   !  /OP-MANEUVER             !
                        ! LOAD                        !                           !
   substitutes          ! LOAD1                       !                           !
   end state: RELEASE
2                       ! /BRAKE-SYS                  !  LOAD                     !
   end state: RELEASE
Saved Sequences:      3  Valid Sequences:       3  Processed:       3
```

From this report, it can be confirmed that for sequence #4, LOAD1 has been substituted for LOAD. Click the exit button located in the upper right corner to exit this report.

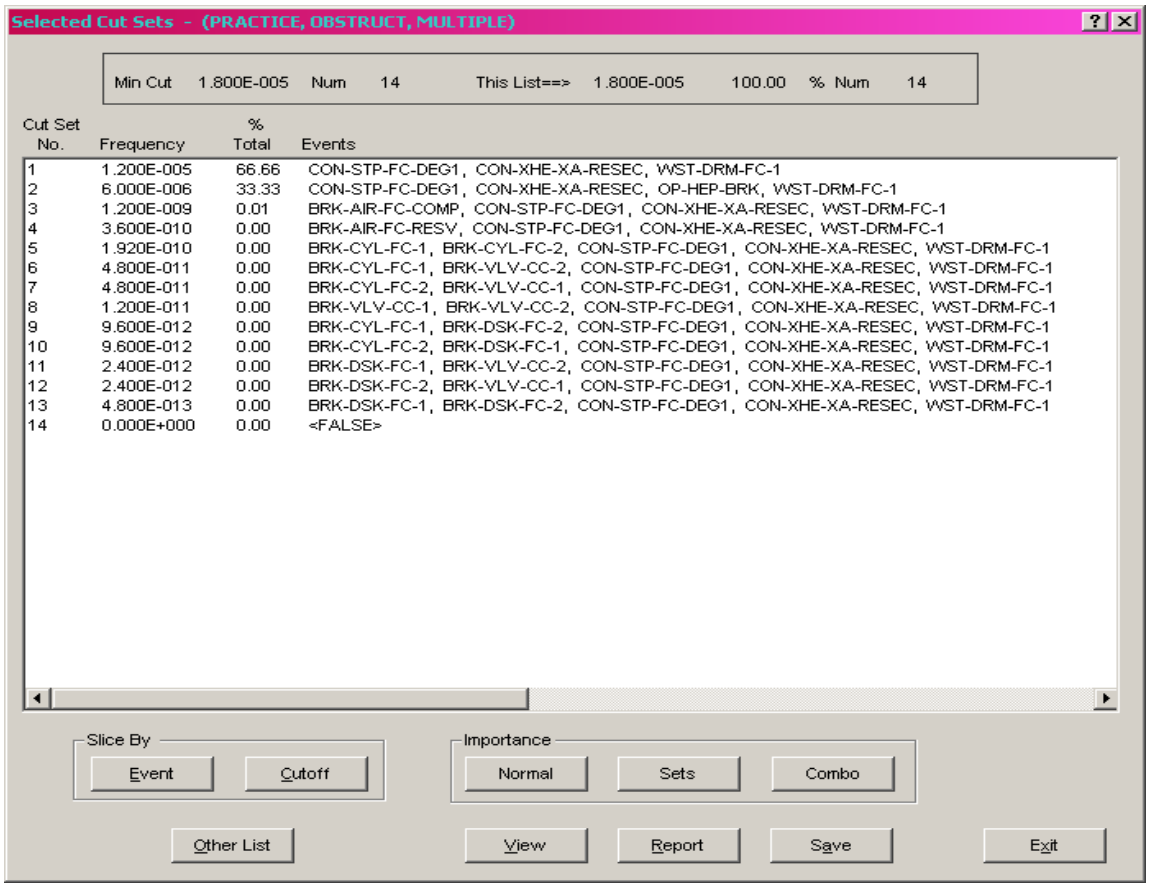
3.2.4 STEP 3: Re-solve For Sequence Cut Sets

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:

Select the Sequence button from the Graphical Tool Bar.

Highlight all the sequences. Right click to invoke the pop-up menu and select the **Solve** option. Click **Ok** to accept the default truncation values.

Right click to invoke the pop-up menu and select Display → Cut Sets. The new cut sets will be displayed as shown below.



3.3 Lesson 12 – Recovery Rules

In Lesson 12, the user will be introduced to the "Recovery Rule" editor. This editor allows rules to be written that affect existing cut sets in a "post-processing" fashion. The SAPHIRE Recovery Rules are logic rules that allow for the alteration or deletion of fault tree or sequence cut sets. The recovery 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 Recovery Rule editor uses standard programming (i.e. FORTRAN, BASIC) line format and structure. Symbols, nomenclature and structure for the Recovery 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 Recovery Rules search existing cut sets as a search criteria. 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 the recovery 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 cuts as directed by the user (i.e. add basic events to the cut set, delete basic events from the cut set or delete the cut altogether). Typical search criteria examples used by the linking rule editor are:

Table 3. SAPHIRE Recovery Rule Search Criteria Examples

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

The recovery rule editor is implemented after the cut sets are generated using the **Solve** command (i.e. searches on pre-existing cut sets). If the recovery 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 recovery 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 recovery 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 Recovery Rule

The recovery rule example will be applied to fault tree, LOAD1. Therefore, this recovery rule is "fault tree recovery 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 recovery rule" (found under the **Sequence** menu option). Fault tree recovery rules are entered into SAPHIRE through the Fault Tree menu button

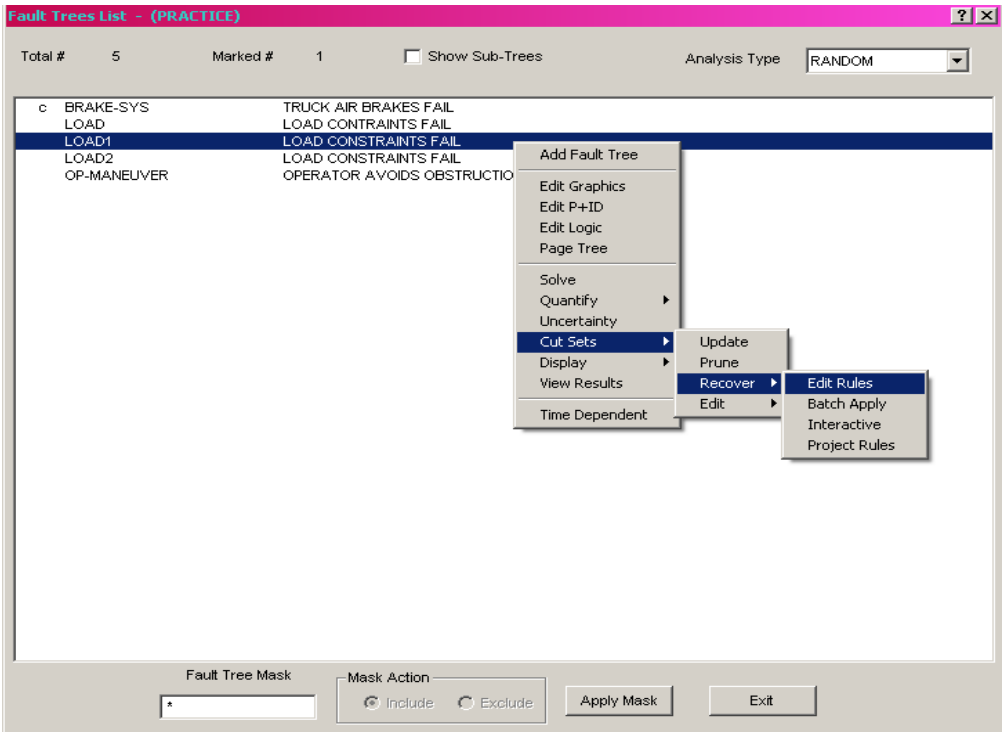
.

The recovery 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:

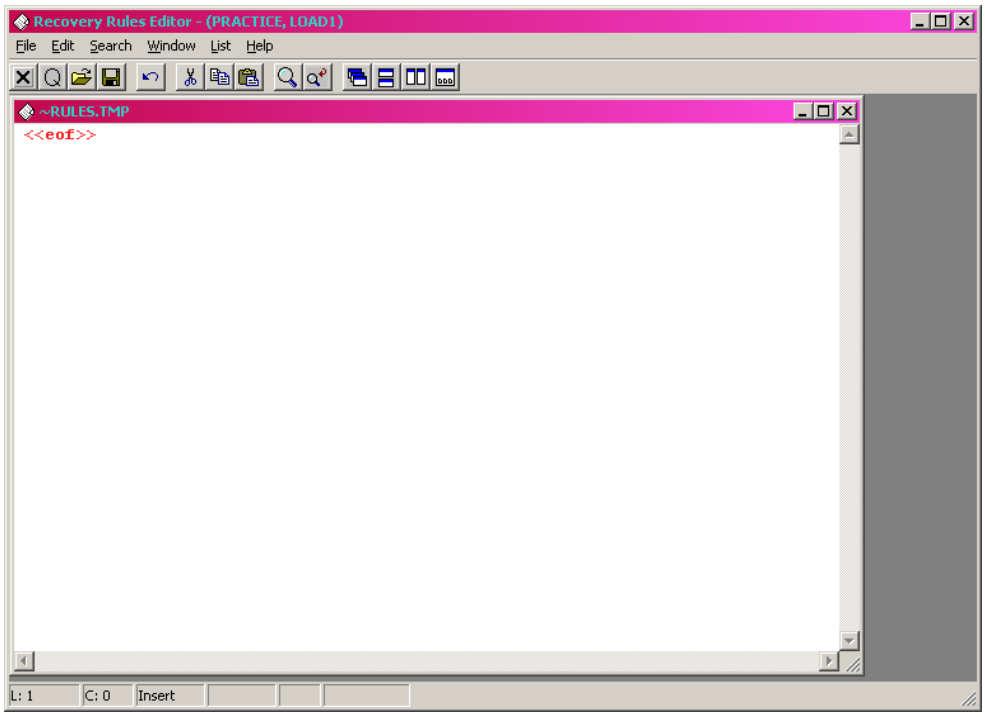
Select the Fault Tree button

from the Graphical Tool Bar.

Highlight the fault tree, LOAD1 and right click to invoke the pop-up menu. Select **Cut Sets → Recover → Edit Rules** as shown below.

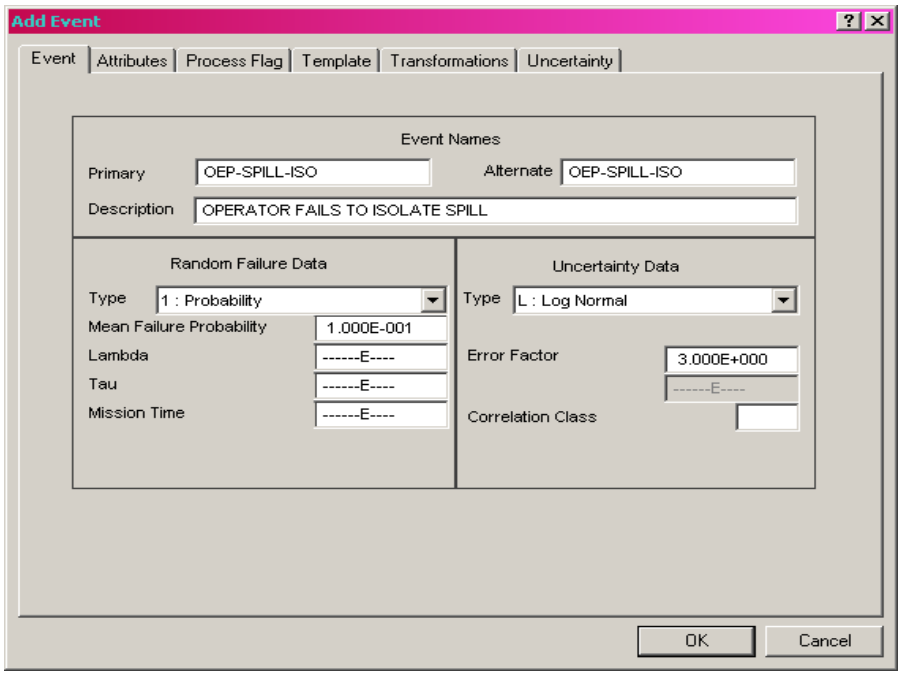


The Recovery Rule is displayed as shown below.



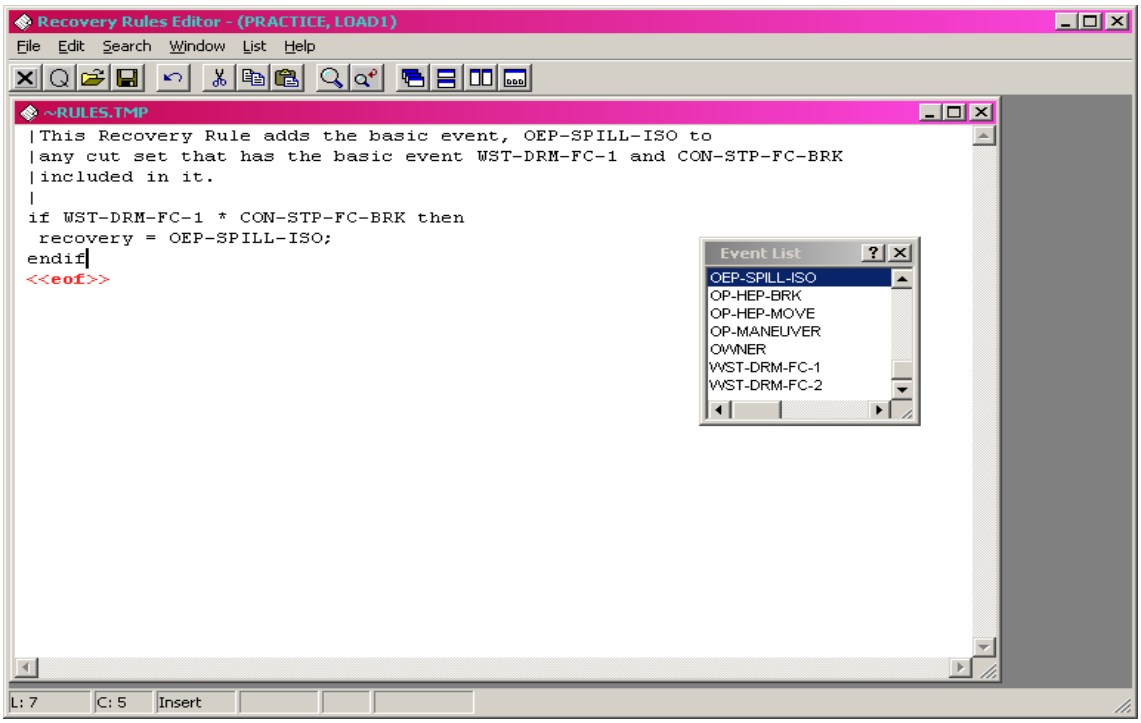
The recovery rules require that the basic event OEP-SPILL-ISO be added to the database. This can be directly in the rule editor by selecting **List** from the rule editor's menu bar and then selecting **Events**.

The Event List Box will be displayed. Place the mouse cursor inside the box and right-click to invoke the pop-up menu. Select **Add**. The Add Event screen is displayed. Enter as the Primary Name, OEP-SPILL-ISO and a **Mean Failure Probability** of 1.0E-1 with a **Lognormal distribution** and an **Error Factor** of 3, as shown below.



Click on the **Ok** button when completed.

Type in the recovery rule shown below. Double clicking on a basic event in the Event List box places that event inside the rule editor in the last location of the cursor.



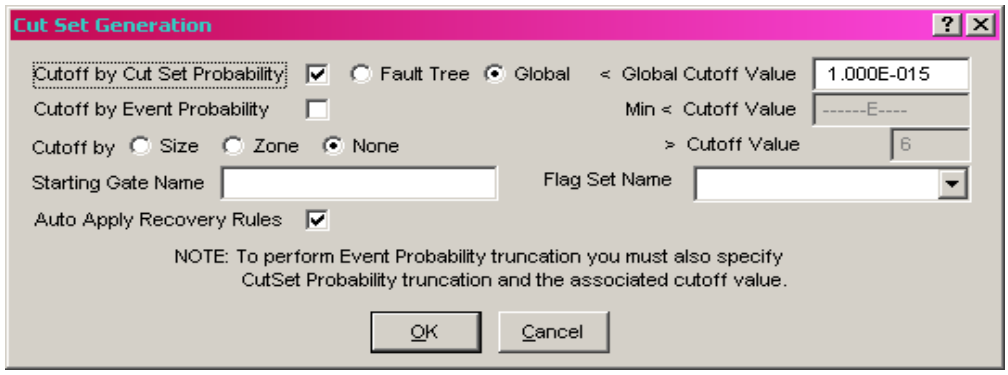
To exit the recovery rule editor, click on the **Exit** button and click on **Yes** to save the recovery rule.

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

The fault tree, LOAD1 can now be solved with the fault tree recovery rule being implemented. To solve for cut sets, perform the following steps:

If the LOAD1 fault tree isn't highlighted, click on the LOAD1 fault tree and right-click to invoke the pop-up menu. Select the **Solve** option.

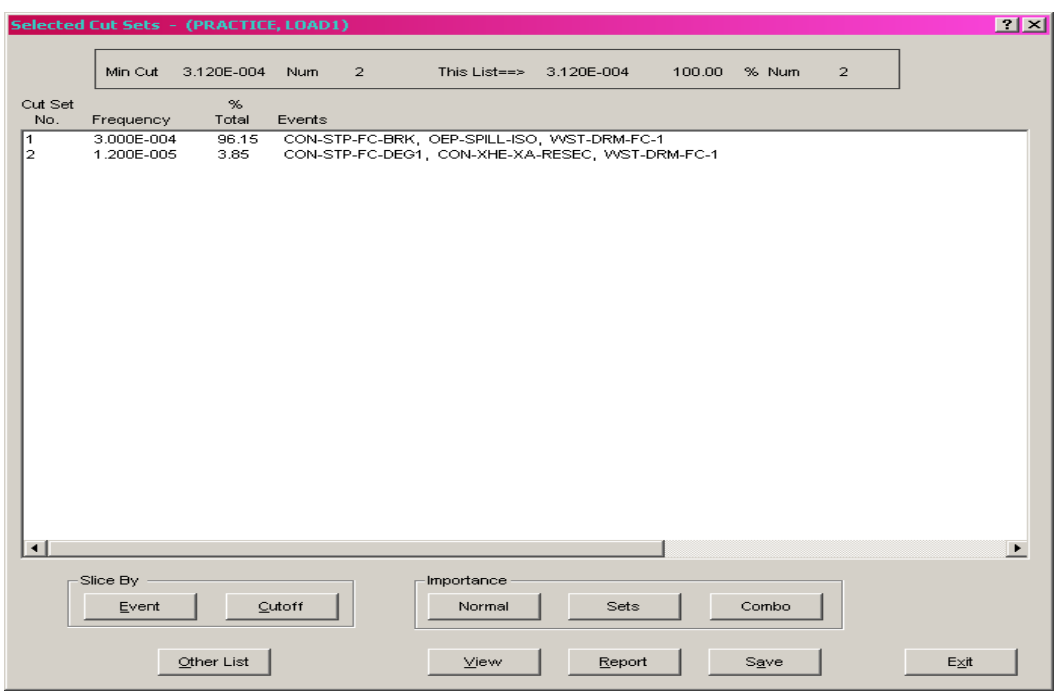
The **Cut Set Generation** screen is shown as below. Make certain that "Auto Apply Recovery Rules" box is check. This automatically applies the recovery rules during the **Solve** command. Click **Ok** to continue.



Note: If the user wanted to apply recovery rules manually, then un-click the "Auto Apply Recovery Rules" box. Apply the recovery rule by right-clicking on the LOAD1 fault tree and selecting **Cut Sets → Recover → Apply Batch**. This command will apply the recovery rule after the cut sets have been solved.

Click the Ok button in the Cut Set Generation Results screen.

To display the cut set with the recovery event applied, select the LOAD1 fault tree and right- click to invoke the pop up menu. **Select Display → Cut Sets**. The **Selected Cut Sets** screen is shown as below. Notice now 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.



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, 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

BIBLIOGRAPHIC DATA SHEET

(See Instructions on the reverse)

1. REPORT NUMBER (Assigned by NRC, Add Vol., Supp., Rev., and Addendum Numbers, if any.)

NUREG/CR-6952
INL/EXT-05-00654

2. TITLE AND SUBTITLE

Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Vol. 4 Tutorial

3. DATE REPORT PUBLISHED

MONTH	YEAR
September	2008

4. FIN OR GRANT NUMBER

N6203

5. AUTHOR(S)

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

6. TYPE OF REPORT

Technical

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.)

Idaho National Laboratory
Battelle Energy Alliance
P.O. Box 1625
Idaho Falls, ID 83415-3850

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.)

Division of Risk Analysis
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001

10. SUPPLEMENTARY NOTES

D. O'Neal, NRC Project Manager

11. ABSTRACT (200 words or less)

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 system. In this document, a series of lessons are provided that guide the user through basic steps common to most analyses preformed 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 software.

12. KEY WORDS/DESCRIPTORS (List words or phrases that will assist researchers in locating the report.)

SAPHIRE, software, reliability, risk, safety, PRA, tutorial, fault tree event tree

13. AVAILABILITY STATEMENT

Unlimited

14. SECURITY CLASSIFICATION

(This page)
Unclassified
(This report)
Unclassified

15. NUMBER OF PAGES

16. PRICE

NRC FORM 335 (2-89)