U.S.NRC

United States Nuclear Regulatory Commission

Protecting People and the Environment

NUREG/CR-6952
Vol. 3, Part B
INL/EXT-05-00644

Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Vol. 3 Code Reference Manual – Part B

Office of Nuclear Regulatory Research

AVAILABILITY OF REFERENCE MATERIALS IN NRC PUBLICATIONS

NRC Reference Material

As of November 1999, you may electronically access NUREG-series publications and other NRC records at NRC's Public Electronic Reading Room at http://www.nrc.gov/reading-rm.html. Publicly released records include, to name a few, NUREG-series publications; *Federal Register* notices; applicant, licensee, and vendor documents and correspondence; NRC correspondence and internal memoranda; bulletins and information notices; inspection and investigative reports; licensee event reports; and Commission papers and their attachments.

NRC publications in the NUREG series, NRC regulations, and *Title 10, Energy*, in the Code of *Federal Regulations* may also be purchased from one of these two sources.

1. The Superintendent of Documents
   U.S. Government Printing Office
   Mail Stop SSOP
   Washington, DC 20402–0001
   Internet: bookstore.gpo.gov
   Telephone: 202-512-1800
   Fax: 202-512-2250
2. The National Technical Information Service
   Springfield, VA 22161–0002
   www.ntis.gov
   1–800–553–6847 or, locally, 703–605–6000

A single copy of each NRC draft report for comment is available free, to the extent of supply, upon written request as follows:

Address: U.S. Nuclear Regulatory Commission
Office of Administration
Mail, Distribution and Messenger Team
Washington, DC 20555-0001
E-mail: DISTRIBUTION@nrc.gov
Facsimile: 301–415–2289

Some publications in the NUREG series that are posted at NRC's Web site address http://www.nrc.gov/reading-rm/doc-collections/nuregs are updated periodically and may differ from the last printed version. Although references to material found on a Web site bear the date the material was accessed, the material available on the date cited may subsequently be removed from the site.

Non-NRC Reference Material

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. Such documents as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings may be purchased from their sponsoring organization.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at—

The NRC Technical Library
Two White Flint North
11545 Rockville Pike
Rockville, MD 20852–2738

These standards are available in the library for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from—

American National Standards Institute
11 West 42nd Street
New York, NY 10036–8002
www.ansi.org
212–642–4900

Legally binding regulatory requirements are stated only in laws; NRC regulations; licenses, including technical specifications; or orders, not in NUREG-series publications. The views expressed in contractor-prepared publications in this series are not necessarily those of the NRC.

The NUREG series comprises (1) technical and administrative reports and books prepared by the staff (NUREG–XXXX) or agency contractors (NUREG/CR–XXXX), (2) proceedings of conferences (NUREG/CP–XXXX), (3) reports resulting from international agreements (NUREG/IA–XXXX), (4) brochures (NUREG/BR–XXXX), and (5) compilations of legal decisions and orders of the Commission and Atomic and Safety Licensing Boards and of Directors' decisions under Section 2.206 of NRC's regulations (NUREG–0750).

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

U.S.NRC
United States Nuclear Regulatory Commission
Protecting People and the Environment

NUREG/CR-6952
Vol. 3, Part B
INL/EXT-05-00644

Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Vol. 3 Code Reference Manual – Part B

Manuscript Completed: October 2007
Date Published: September 2008

Prepared by
K.J. Kvarfordt, S.T. Wood, C.L. Smith

Idaho National Laboratory
Battelle Energy Alliance
Idaho Falls, ID 83415

D. O'Neal, NRC Project Manager

NRC Job Code N6203

Office of Nuclear Regulatory Research

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) is a software application developed for performing a complete probabilistic risk assessment (PRA) using a personal computer. SAPHIRE is funded by the U.S. Nuclear Regulatory Commission (NRC) and developed by the Idaho National Laboratory (INL). The INL's primary role in this project is that of software developer. However, the INL also plays an important role in technology transfer by interfacing and supporting SAPHIRE users comprised of a wide range of PRA practitioners from the NRC, national laboratories, the private sector, and foreign countries.

SAPHIRE can be used to model a complex system's response to initiating events, quantify associated damage outcome frequencies, and identify important contributors to this damage (Level 1 PRA) and to analyze containment performance during a severe accident and quantify radioactive releases (Level 2 PRA). It can be used for a PRA evaluating a variety of operating conditions, for example, for a nuclear reactor at full power, low power, or at shutdown conditions. Furthermore, SAPHIRE can be used to analyze both internal and external initiating events and has 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 risk in terms of release consequences to both the public and the environment (Level 3 PRA).

SAPHIRE includes a separate module called the Graphical Evaluation Module (GEM). GEM provides a highly specialized user interface with SAPHIRE that automates SAPHIRE process steps for evaluating operational events at commercial nuclear power plants. Using GEM, an analyst can estimate the risk associated with operational events in a very efficient and expeditious manner.

This reference guide will introduce the SAPHIRE Version 7.0 software. A brief discussion of the purpose and history of the software is included along with general information such as installation instructions, starting and stopping the program, and some pointers on how to get around inside the program. Next, database concepts and structure are discussed. Following that discussion are nine sections, one for each of the menu options on the SAPHIRE main menu, wherein the purpose and general capabilities for each option are furnished. Next, the capabilities and limitations of the software are provided.

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

EXECUTIVE SUMMARY

The U.S. Nuclear Regulatory Commission (NRC) has developed a powerful personal computer (PC) software application for performing probabilistic risk assessments (PRAs), called Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE). SAPHIRE capabilities for performing a PRA are summarized in this volume, and a more detailed description can be found in the individual sections of the documentation.

With the release of SAPHIRE versions 5 and 6, INL included a separate module called the Graphical Evaluation Module (GEM). GEM provides a highly specialized user interface with SAPHIRE, automating SAPHIRE process steps for evaluating operational events at commercial nuclear power plants. In particular, GEM implements many of the accident sequence precursor (ASP) program analysis methods. Using GEM, an analyst can estimate the risk associated with operational events very efficiently and expeditiously.

This SAPHIRE manual is designed to be an on-line reference guide to the program. Divided into two parts (A and B), this manual is a step-by-step approach to using the features of SAPHIRE. Introductions and installation instructions for the SAPHIRE are first presented, along with some information about the basic features of SAPHIRE. The next sections contain information about SAPHIRE data base concepts so that you can begin learning how SAPHIRE works. The sections following present a step-by-step approach to using the features of SAPHIRE. Later sections provide information about the capabilities and limitations of SAPHIRE, including some of the advanced features of SAPHIRE.

For readers who are familiar with previous versions of SAPHIRE, following are a list of some of the features that have been enhanced or are new to SAPHIRE version 7.

- Improved Graphical Editing
- Improved Fault Tree Logic Editing
- Event Tree Linking Rules
- Sequence Recovery Rules
- Fault Tree Recovery Rules
- Partition Rules

SAPHIRE automatically detects when the database schema for a project is different from the current version of the code. When SAPHIRE attempts to open a project created with an older version of the database schema, the Version Conflict dialog will appear. Once the data has been converted to version 7, this data cannot be used with older versions of SAPHIRE (IRRAS, SARA, GEM, or FEP).

Users will find that SAPHIRE has evolved with advances in computers. Previously, it was run in the DOS operating system. It consisted of a suite of modules as described in the NUREG/CR-6116, "System Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Version 5.0," series of volumes 1 - 8. The current versions in use, 6 and 7, run under the Windows environment. Certain modules have been integrated in the SAPHIRE environment; however, the Graphical Evaluation Module (GEM) remains a separate interface with the SAPHIRE code. GEM is a user-friendly interface that streamlines and automates select SAPHIRE inputs and processes for performing events assessments.

SAPHIRE has also evolved with users' needs. New features and capabilities have been added in Versions 6 and 7 for developing and using larger, more complex models. For example, Version 7 can solve up to 2 million sequences. In addition, enhancements have also been made in Version 7 for cut set slicing, event tree rule linkage rules, and reporting options.

To help assure the quality of new releases, SAPHIRE Versions 6 and 7 were used extensively with models created in earlier versions of SAPHIRE. The models were loaded into the current version of the software and results of the analyses were compared against SAPHIRE Version 5.0. Any discrepancies found were corrected in subsequent minor versions of the code. SAPHIRE has been validated in several ways.

ACRONYMS

DOE	Department of Energy
GEM	Graphical Evaluation Module
INL	Idaho National Laboratory
IRRAS	Integrated Reliability and Risk Analysis System
NRC	Nuclear Regulatory Commission
PRA	probabilistic risk analysis
SAPHIRE	Systems Analysis Programs for Hands-on Integrated Reliability Evaluations

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

Vol. 3 Code Reference Manual – Part B

11. UTILITY OPTIONS

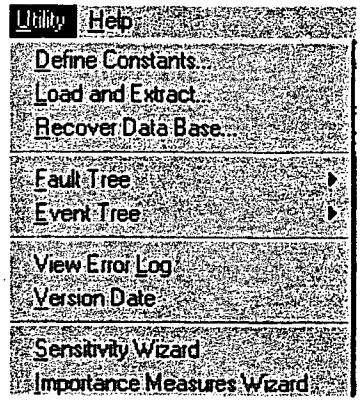
11.1 Utility Options

PURPOSE

This option allows you to perform routine functions that are required by SAPHIRE such as defining constants, recovering the database, and MAR D data exchange.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. The drop-down menu with available options will be displayed.



11.2 Defining Constants

11.2.1 General Tab

11.2.1.1 Defining Constants

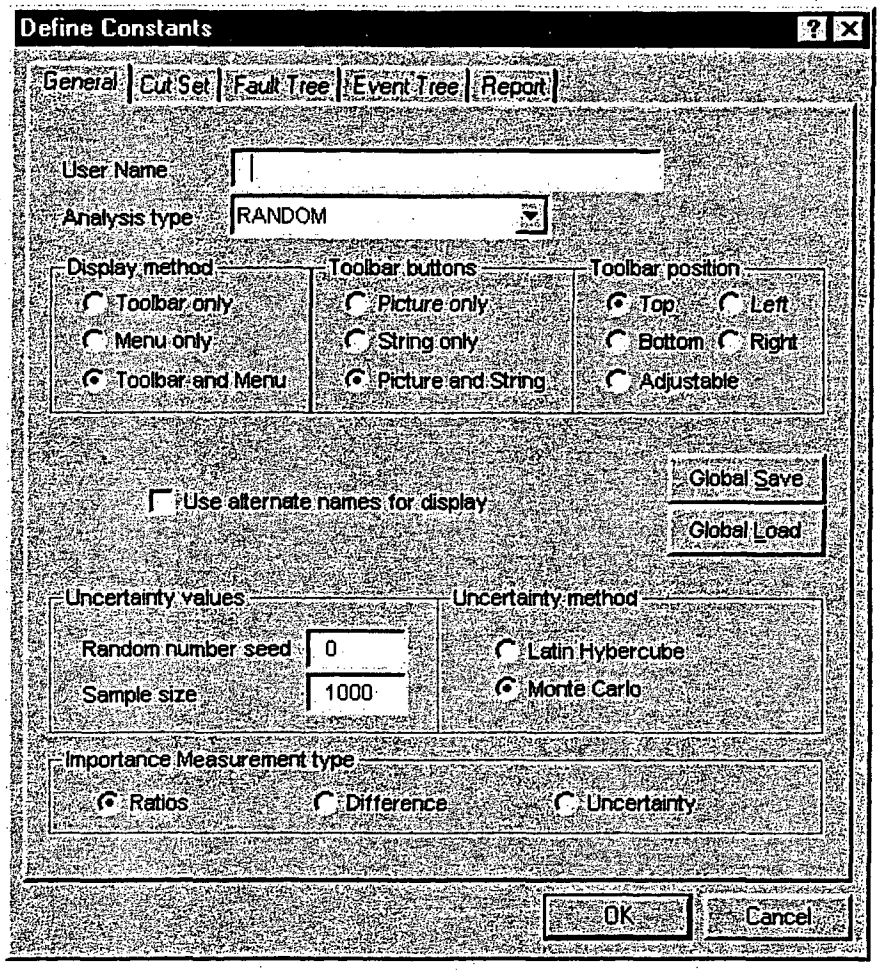
PURPOSE

This option provides the means to specify general information, cut set option defaults, and graphical editor defaults. The define constants dialog consists of four property sheets or "pages," each containing information pertaining to program defaults, grouped by related properties. Select the desired page by clicking on the tab.

STEPS

1. From the menu select **Utility**.
2. Choose **Define Constants** from the menu. The *Define Constants* dialog will be displayed.

Initially, the *General* constants are available when this option is selected.



After setting the general information you may choose one of the other pages: Cut Set , Fault Tree, Event Tree, or Report to change other default values.

Global Save - Save the current values as the SAPHIRE global defaults. Global defaults will be used as the initial constants each time a new project is created.
Global Load - Load the SAPHIRE global default values.
OK - Save the changes made and close the *Define Constants* dialog.
Cancel - Close the *Define Constants* dialog without saving changes.

11.2.1.2 User Name

This 36-character, alphanumeric field identifies the program user. This information is optional.

11.2.1.3 Analysis type

The default type of analysis to perform. Initially, the RANDOM analysis type is selected.

11.2.1.4 Display Method

Toolbar only - Shows only the toolbar in the main window.
Menu only - Shows only the menu in the main window.
Toolbar and Menu - Shows both the toolbar and menu on the main window.

11.2.1.5 Toolbar Buttons

These options are applicable if the toolbar is displayed in the main window.

Picture only - Shows only the bitmap (picture) on the toolbar buttons.
String only - Shows only the text (string) on the toolbar buttons.
Picture and String - Shows both the bitmap and text on the toolbar buttons.

11.2.1.6 Toolbar Position

These options are applicable if the toolbar is displayed in the main window.

Top - Positions the toolbar horizontally at the top of the main window.
Bottom - Positions the toolbar horizontally at the bottom of the main window.
Left - Positions the toolbar vertically on the left side of the main window.
Right - Positions the toolbar vertically on the right side of the main window.
Adjustable - Allows the toolbar to be resized.

11.2.1.7 Use alternate names

Choose the primary or alternate name field to display throughout SAPHIRE. If this check box is selected, the alternate names will be used throughout the system; otherwise the primary names will be used. This is useful when a project has been developed for dual languages.

11.2.1.8 Random number seed

The default 5-digit numeric field indicating the first random number in the seed to be used in the uncertainty calculation. The initial value, 0, indicates that the random number will be the current value of the real system clock. This value can be temporarily changed when performing uncertainty calculations.

11.2.1.9 Sample size

The default 6-digit numeric field indicating the number of uncertainty samples to be run. The sample size may range from 1 to 999999. The initial value is 1000. This value can be temporarily changed when performing uncertainty calculations.

11.2.1.10 Uncertainty method

Choose the default method for uncertainty analysis. The initial selection is Monte Carlo. This can be temporarily changed when performing uncertainty calculations.

11.2.1.11 Importance Measurement type

Choose the default importance measurement type.

Ratio -	Fussell-Vessely importance, risk reduction ratio , and risk increase ratio .
Difference -	Birnbaum importance, risk reduction interval , and risk increase interval .
Uncertainty -	Quantification of the contribution of each individual basic event's uncertainty to the total output uncertainty.

11.2.2 Cut Set Tab

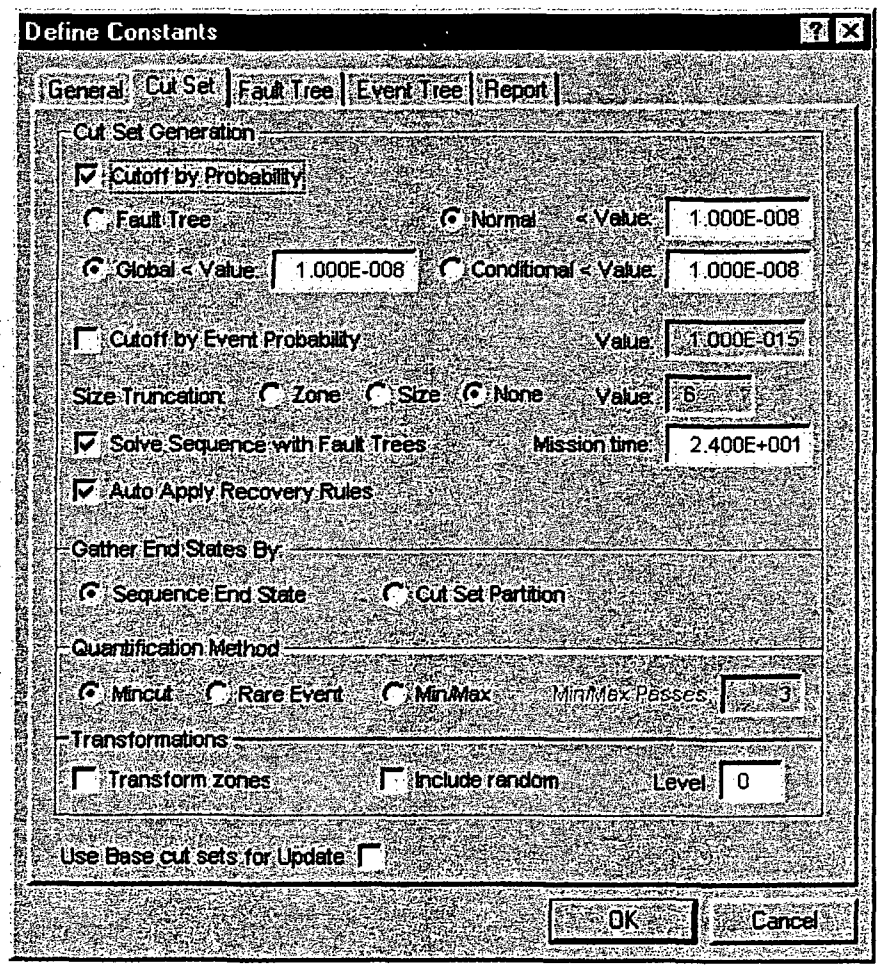
11.2.2.1 Cut Set Constants

PURPOSE

This option allows you to specify default values for cut set generation.

STEPS

1. From the menu select **Utility**.
2. Choose **Define Constants** from the menu. The *Define Constants* dialog will be displayed.
3. Select the **Cut Set** tab.



After setting the cut set defaults, you may choose one of the other pages: General , Fault Tree, Event Tree, or Report to change other default values.

Save - Save the changes made and close the *Define Constants* dialog.
Cancel - Close the *Define Constants* dialog without saving changes.

11.2.2.2 Fault Tree Probability Cutoff

If the **Cutoff by Probability** checkbox is selected, cut sets will be retained or discarded during fault tree cut set generation based on the cut set probability value. The default value is initially set to 1.0E-8. These settings can be temporarily changed when you solve.

If you select **Cutoff by Probability** checkbox and choose:

Fault Tree - Only those cut sets whose product for all of its event probabilities is greater than or equal to the fault tree's cutoff value will be kept. This value is entered in the **Modify | Fault Trees** option.

Global - Only those cut sets whose product for all of its event probabilities is greater than or equal to the value in the < *Global Cutoff Value* field will be kept.

All other cut sets will be removed.

If you deselect this check box, then the probability for the cut set will be irrelevant for determining if the cut set should be retained or discarded.

11.2.2.3 Sequence Probability Cutoff

If the **Cutoff by Probability** checkbox is selected, cut sets will be retained or discarded during sequence cut set generation based on the cut set probability value. The default value is initially set to 1.0E-8. These settings can be temporarily changed when you solve.

If you select **Cutoff by Probability** checkbox and choose:

Normal - Only those cut sets whose product for all of its event probabilities is greater than or equal to the value in the *<Cutoff Value* field will be kept.

Conditional - Only those cut sets whose product for all of its event probabilities, excluding the initiating event probability, is greater than or equal to the value in the *<Cutoff Value* field will be kept.

All other cut sets will be removed.

If you deselect this check box, then the probability for the cut set will be irrelevant for determining if the cut set should be retained or discarded.

11.2.2.4 Cutoff by Event Probability

The default event probability cutoff value. This option will check all cut sets that are below the probability cutoff and remove them only if they contain an event whose probability is below this value.

If you select this check box, then you must also choose the ***Cutoff by Probability*** check box and provide a value.

11.2.2.5 Size Truncation

The default size truncation method and value. When generating fault tree, sequence, or end state cut sets, this field is used for determining if cut sets should be retained or discarded based on the number of basic events in the cut sets. These settings can be changed temporarily when you solve or gather.

Size - The default maximum number of basic events allowed in cut set generation. The default value is initially set to six.

Zone - Only zone flagged events will be checked.

None - The number of events in a cut set is irrelevant for determining if the cut set will be retained.

11.2.2.6 Solve Sequences with Fault Trees

The default method for solving sequence cut sets. If checked then by default, the fault tree logic will be used to solve the sequence cut sets. If unchecked, previously generated fault tree cut sets will be used. This value can be temporarily changed in when solving sequence cut sets.

11.2.2.7 Mission Time

The default mission time to be used in the calculation of basic event probabilities, in hours. Initially, this value is 2.4E+01.

11.2.2.8 Auto Apply Recovery Rules

If checked, automatically apply recovery rules after cut set generation.

11.2.2.9 Gather End States by

The default method for gathering end state cut sets. This can be temporarily changed when gathering end state cut sets.

Seq End State - Gather cut sets by the end state assigned to each sequence.
Cut Set Partition - The rule-based (via the partition rules) end state of each cut set will be used for gathering.

11.2.2.10 Quantification Method

Choose the default quantification method to be used for quantifying fault tree, sequence, or end state cut sets.

Mincut - Use the Minimal Cut Set Upper Bound Approximation method.
Rare Event - Add together the probabilities for the cut sets of a top event.
Min/Max - Use the "exact" probability quantification algorithm.

11.2.2.11 Transform Zones

If checked, perform zone transformation during cut set generation.

11.2.2.12 Include Random

If checked, when performing location transformations, include random failures of the event in the transformations.

11.2.2.13 Level

An integer 0 -255 which indicates the default level of substitution for the transformations to be performed.

11.2.2.14 Use Base Case

If checked, use base case cut sets for fault tree, sequence, and end state update. Otherwise use current case cut sets.

11.2.3 Fault Tree Tab

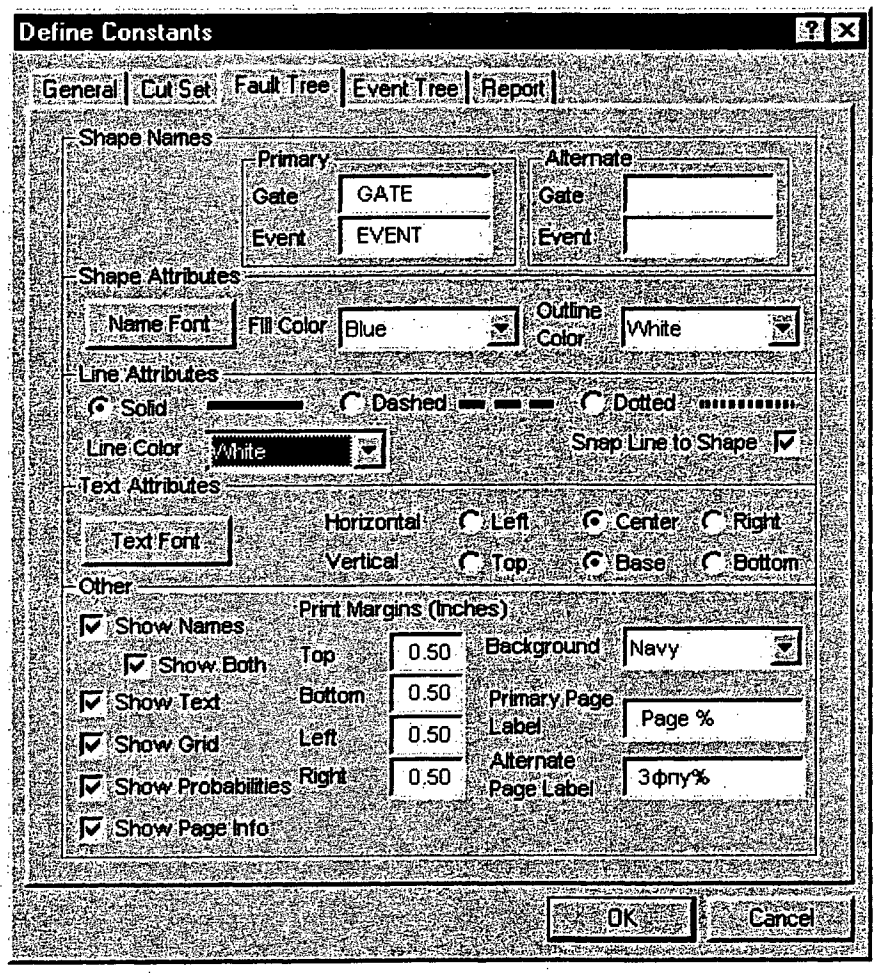
11.2.3.1 Fault Tree Constants

PURPOSE

This option allows you to specify default values for the graphical *Fault Tree Editor.*

STEPS

1. From the menu select **Utility**.
2. Choose **Define Constants** from the menu. The *Define Constants* dialog will be displayed.
3. Select the **Fault Tree** tab.



After setting the fault tree defaults, you may choose one of the other pages: General, Cut Set, or Event Tree, or Report to change other default values.

Save - Save the changes made and close the *Define Constants* dialog.
Cancel - Close the *Define Constants* dialog without saving changes.

11.2.3.2 Gate Name

10-character field for the fault tree default gate name. Each gate created in the fault tree editor will be given this name followed by a sequential number for each separate gate, e.g., GATE1, GATE2, etc. Both a default primary name and a default alternate name may be specified.

11.2.3.3 Alternate Gate Name

10-character field for the fault tree default alternate gate name. This feature allows names to be displayed and reported using a different naming scheme or another locale. For this name to be displayed in lists and on reports instead of the Primary gate name, choose **Use alternate names for display** in the **Define Constants** option. Each gate created in the fault tree editor will be given this alternate name followed by a sequential number for each separate gate, e.g., ALTGATE1, ALTGATE2, etc.

11.2.3.4 Event Name

10-character field for the fault tree default basic event name. Each event created in the fault tree editor will be given this name followed by a sequential number for each separate event, e.g., EVENT1, EVENT2, etc.

11.2.3.5 Alternate Event Name

10-character field for the fault tree default alternate basic event name. This feature allows names to be displayed and reported using a different naming scheme or another locale. For this name to be displayed in lists and on reports instead of the Primary event name, choose **Use alternate names for display** in the **Define Constants** option. Each event created in the fault tree editor will be given thisalternate name followed by a sequential number for each separate event, e.g., ALTEVNT1, ALTEVNT2, etc.

11.2.3.6 Name Font

The default font for displaying names. Change the name font attributes by choosing this button.

SEE ALSO

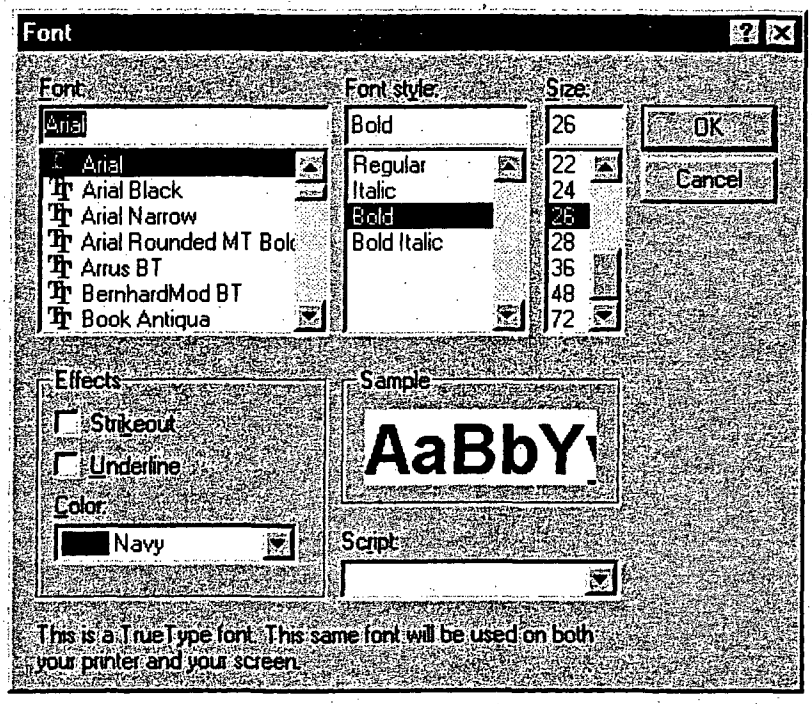
Font Selection

11.2.3.7 Font Selection

This option allows you to change default attributes for a text object font or shape name font. This option is accessible from either the *Fault Tree* page or the *Event Tree* page on the *Define Constants* dialog.

STEPS

1. From the menu select **Utility**.
2. Choose **Define Constants** from the menu. The *Define Constants* dialog will be displayed.
3. Select the **Fault Tree** or **Event Tree** tab.
4. Choose the desired **Font** button. The *Font* dialog will be displayed.
5. Select the desired font from the list and choose or change the style, size, color, etc.



Font -	List of available typeface names.
Font Style -	Bold, italic, bold italic, or regular styles.
Size -	Point size of the selected typeface.
Effects -	Underline, strikeout text color.
Sample -	Example of the appearance of the highlighted font.
Script -	Character set.
OK -	Close the *Select Font* dialog and change the font.
Cancel -	Close the *Select Font* dialog without changing the font.

11.2.3.8 Fill Color

The default color for filling in shapes (i.e., the interior color of the shape). Change the fill color by selecting from the drop-down list.

11.2.3.9 Outline Color

The default color for outlining shapes (i.e., the exterior color of the shape). Change the outline color by selecting from the drop-down list.

11.2.3.10 Line Style

Select the default line type. Connecting lines will be drawn using this style.

11.2.3.11 Line Color

The default color for drawing connecting lines. Change the line color by selecting from the drop-down list.

11.2.3.12 Text Font

The default font for writing text. Change the text font attributes by choosing this button.

11.2.3.13 Horizontal Justification

Select the default horizontal justification for text objects:

Left - The text is aligned along the left margin.
Center - The text is centered on the page. This is the initial value.
Right - The text is aligned along the right margin.

11.2.3.14 Vertical Justification

Select the default vertical justification for text objects:

Base - The text is aligned slightly above the bottom of the text box. This is the initial value.
Bottom - The text is aligned at the bottom of the text box.
Top - The text is aligned at the top of the text box.

11.2.3.15 Show Names

If selected, turns on the display of event and gate names. By default, this option is selected.

11.2.3.16 Show Both

If selected, turns on the display of both primary and alternate names for events and gates.

11.2.3.17 Show Text

If selected, turns on the display of descriptive text. By default, this option is selected.

11.2.3.18 Show Grid

If selected, turns the reference grid on. By default, this option is not selected.

11.2.3.19 Show Probabilities

If selected, turns on the display of basic event probabilites. By default, this option is not selected.

11.2.3.20 Show Page Information

If selected, turns on the display of page information. Information displayed includes the name and description of the fault tree or event tree, the current date, and the page number. By default, this option is selected.

11.2.3.21 Print Margins

Set the top, bottom, left, and right margins, in inches, for printing. Enter the values in real numbers. By default the margins are set to 0.50 (1/2 inch).

11.2.3.22 Background Color

The default background color for the graphical editor's window. Change the background color by selecting from the drop-down list.

11.2.3.23 Primary Page Label

If selected, turns on the page label display. The label entered is inserted in front of the page number. The symbol "%" adds a space between the label and the page number. By default, the label "Page%" is provided. If the label is left blank, then the page number will appear without any label. To omit both the label and the page number, use the carat (^) symbol.

11.2.3.24 Alternate Page Label

If selected, turns on the alternate page label display. The label entered is inserted in front of the page number. This label is used when alternate names are displayed and printed. The alternate name display feature is set in the **Define Constants** option.

11.2.4 Event Tree Tab

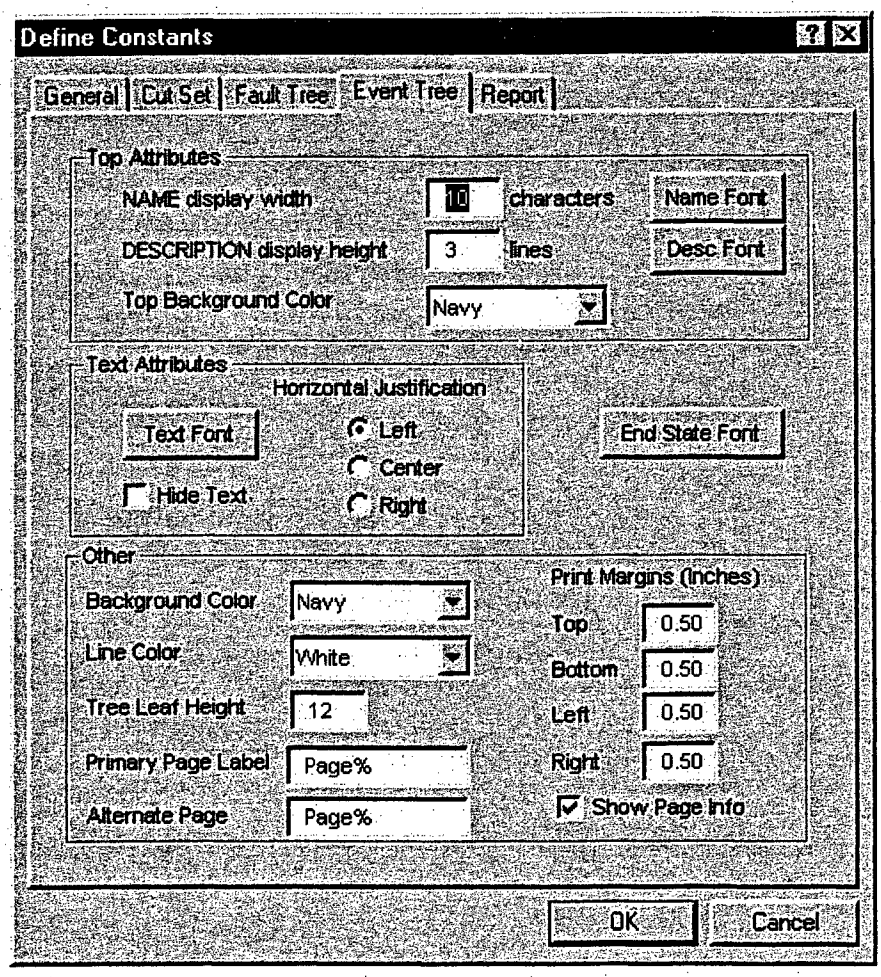
11.2.4.1 Event Tree Constants

PURPOSE

This option allows you to specify default values for the event tree graphical editor.

STEPS

1. From the menu select **Utility**.
2. Choose **Define Constants** from the menu. The *Define Constants* dialog will be displayed.
3. Select the **Event Tree** tab.



After setting the event tree defaults, you may choose one of the other constants pages: General, Cut Set, Fault Tree, or Report to change other default values.

OK - Save the changes made and close the *Define Constants* dialog.
Cancel - Close the *Define Constants* dialog without saving changes.

11.2.4.2 Name Display Width

The default number of characters in the name which will be displayed.

11.2.4.3 Name Font

The default font for displaying names. Change the name font attributes by choosing this button.

11.2.4.4 Description Display Height

The default number of lines of the description which will be displayed.

11.2.4.5 Description Font

The default font for displaying descriptions. Change the description font attributes by choosing this button .

11.2.4.6 Top Background Color

The default background color for the area where the top names are displayed in the graphical editor's window. Change the top background color by selecting from the drop-down list.

11.2.4.7 Text Font

The default font for writing text. Change the text font attributes by choosing this button.

11.2.4.8 Hide Text

If selected, the text will be hidden when displaying the tree.

11.2.4.9 Horizontal Justification

Select the default horizontal justification for text objects:

Left - The text is aligned along the left margin.
Center - The text is centered on the page. This is the initial value.
Right - The text is aligned along the right margin.

11.2.4.10 End State Font

The default font for displaying end state names. Change the end state name font attributes by choosing this button.

11.2.4.11 Background Color

The default background color for the graphical editor's window. Change the background color by selecting from the drop-down list.

11.2.4.12 Line Color

The default color of lines. Change the line color by selecting from the drop-down list.

11.2.4.13 Tree Leaf Height

The spacing, in relative units, between the branches.

11.2.4.14 Primary Page Label

If selected, turns on the page label display. The label entered is inserted in front of the page number. The symbol "%" adds a space between the label and the page number. By default, the label "Page%" is provided. If the label is left blank, then the page number will appear without any label. To omit both the label and the page number, use the carat (^) symbol.

11.2.4.15 Alternate Page Label

If selected, turns on the alternate page label display. The label entered is inserted in front of the page number. This label is used when alternate names are displayed and printed. The alternate name display feature is set in the **Define Constants** option.

11.2.4.16 Print Margins

Set the top, bottom, left, and right margins, in inches, for printing. Enter the values in real numbers. By default the margins are set to 0.50 (1/2 inch).

11.2.4.17 Show Page Information

If selected, turns on the display of page information. Information displayed includes the name and description of the fault tree or event tree, the current date, and the page number. By default, this option is selected.

11.2.5 Report Tab

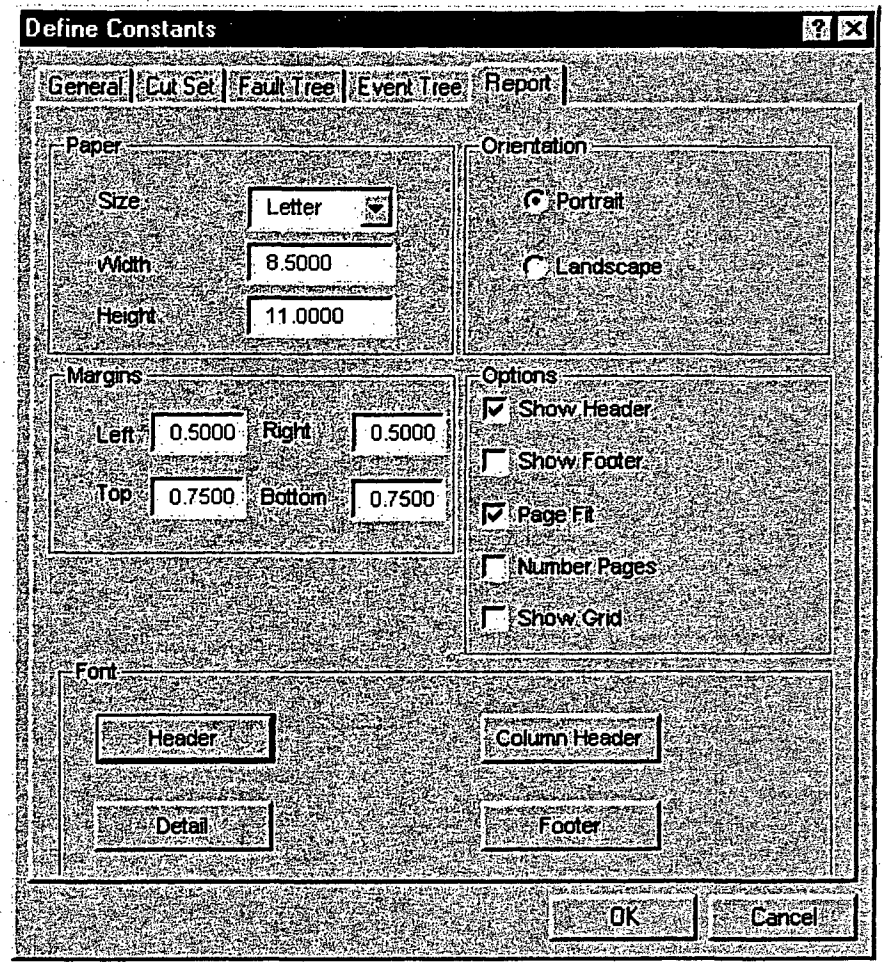
11.2.5.1 Report Constants

PURPOSE

This option allows you to specify default values for report output, where applicable.

STEPS

1. From the menu select **Utility**.
2. Choose **Define Constants** from the menu. The *Define Constants* dialog will be displayed.
3. Select the **Report** tab.



After setting the event tree defaults, you may choose one of the other constants pages: General, Cut Set, or Fault Tree to change other default values.

Paper -	Set the desired paper size, height, and width.
Orientation -	Select the desired paper orientation.
Options -	Turn the header, footer, page numbering, page fit, and grid options on (checked) or off (unchecked).
Margins -	Set the left, right, top, and bottom margins for the report.
Font -	Select the font characteristics for the header, column header, detail, and footer areas of the report.
OK -	Save the changes made and close the *Define Constants* dialog.
Cancel -	Close the *Define Constants* dialog without saving changes.

11.3 Load and Extract

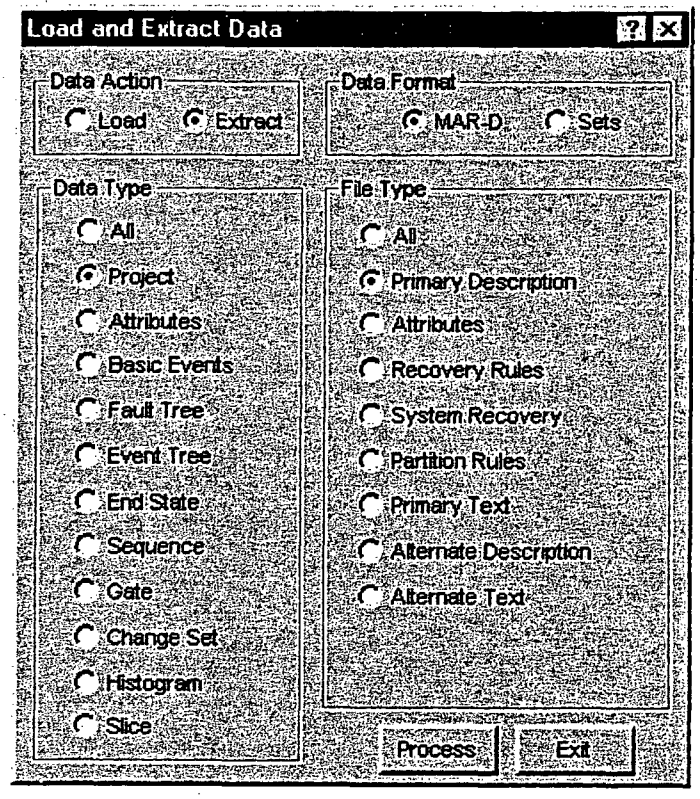
11.3.1 Loading and Extracting Data

PURPOSE

This option allows you to load probabilistic risk assessment data from the Models and Results Data Base (MAR D) or Set Equation Transformation System (SETS) database into SAPHIRE. Additionally, you can output SAPHIRE formatted data to the generic MAR-D format or SETS format.

STEPS

1. From the SAPHIRE menu select Utility.
2. Choose Load and Extract from the menu. The Load and Extract Data dialog will be displayed.
3. Select the desired Data Action, Data Format, Data Type, then File Type.
4. Choose the Process button to continue.



Data Action

Load - Load PRA data into the currently selected SAPHIRE project.

Extract - Output SAPHIRE formatted data of the currently selected project to a file in the current project directory.

Data Format

MAR-D - Use the Models and Results Data Base format.

SETS - Use the Set Equation Transformation System file format.

Data Type - The type of data to be loaded or extracted, e.g., Project, Basic Event, etc.

File Type - The subset data of the selected *Data Type*.

Process - Proceed with the loading or extracting operation.

Exit - Close the *Load and Extract Data* dialog.

11.3.2 Loading MAR-D Data

11.3.2.1 Loading MAR-D Data

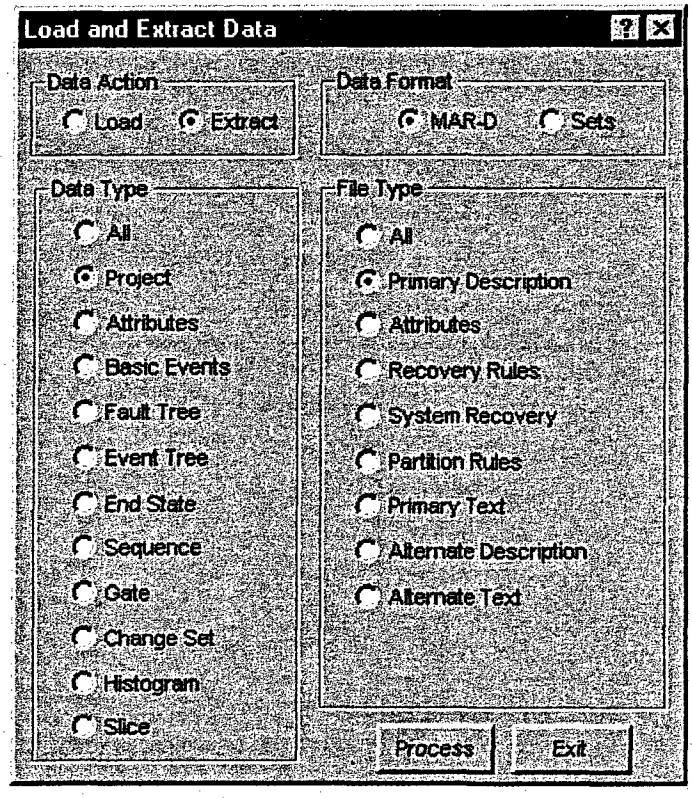
PURPOSE

This option facilitates loading of probabilistic risk assessment data from the Models and Results Data Base (MAR D). The process converts information from the generic format found in the MAR D database to the SAPHIRE database format. New data loaded into the database overwrites old data of the same name.

In general, the dialogs and operations are similar for each of the *Data Types*, therefore, a generic discussion is provided here.

STEPS

1. Before loading any data, a project must be created through the **Modify | Project** option.
2. Copy the files you wish to load into that project's folder (directory).
3. Select the project and verify that the project name displayed in the title bar of the SAPHIRE window is where the data should be loaded. If it is not correct, use the **File | Open Project** option again to choose the desired project.
4. Select the **Utility | Load and Extract** option. The *Load and Extract Data* dialog will be displayed.



As you click the different *Data Type* radio buttons, corresponding *File Type* options will appear to the right, listing the types of data that can be loaded.

After selecting the data to load, choose the **Process** button. The *Load* dialog will appear listing the files that contain the specified data and have the proper file extension.

5. Click on the *Data Type* in the graphic, above, for additional information on loading various types of data.

11.3.2.2 Load Dialog

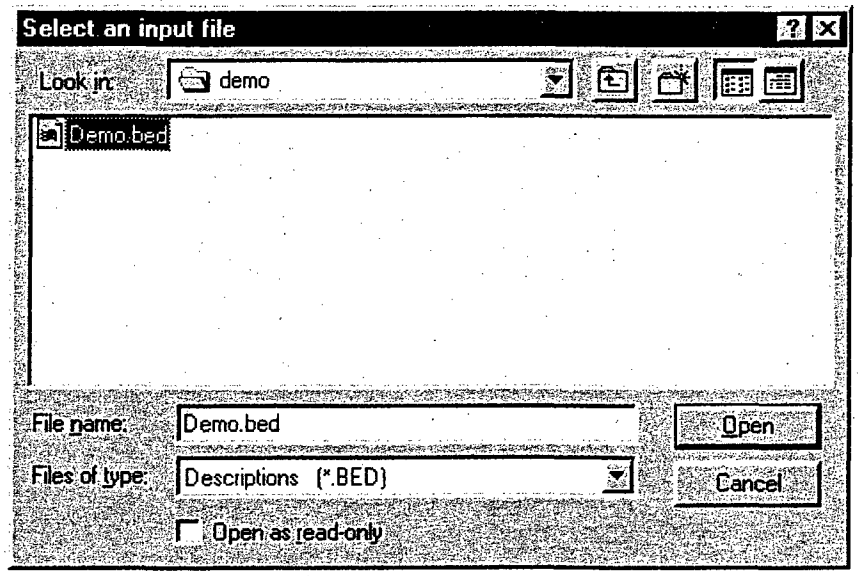
PURPOSE

Converts information from the generic format found in the MAR D database to the SAPHIRE database format.

STEPS

1. Choose the desired *Data Type* and *File Type* radio buttons along with the **Load** *Data Action* radio button.
2. Select the **Process** button. The *Select an input file* dialog will be displayed.
3. Select the desired folder, if necessary.
4. Highlight the desired file and choose the **Open** button.
   OR
5. Double click the desired file.

This dialog lists the existing files of the appropriate file type (depending on the *File Type* radio button previously selected) that can be loaded.



Open - Load the data in the selected file into the current project data base. If successful, the message, "Load complete," will be displayed in the status bar of the SAPHIRE window.
Cancel - Close the *Select an input file* dialog without loading data.

11.3.2.3 General Format Rules

The following general rules apply to the PRA Models and Results Data Base (MAR-D) data interchange format. These rules apply to all files unless specifically stated otherwise.

All name references (project names, event names, etc.) must be upper case alphanumeric. All lower case characters will be converted to upper case. Any alpha fields that are longer than the format specified will be truncated. No spaces are allowed in the middle of names.

Descriptions can have both upper-case and lower-case characters. No character checking will be done. No commas are allowed in the description.

Commas are used as field delimiters in most formats, and can be used as placeholders for unknown fields. Any number of leading and trailing field spaces can be inserted. Exceptions to this format are detailed as needed.

Text rules:

File is standard ASCII text, single spaced, upper and lower case.

First line of paragraph is indented 5 spaces, with a blank line between paragraphs.

^EOS signals the End of Section so that multiple names in the same project can be collected in one file.

11.3.2.4 Load Verification

PURPOSE

This provides consistency checking for names contained in multiple files. Data files that have no prerequisite loaded files (*.CTD, *.FTD, *.BED, etc.) are assumed to be correct and are loaded directly into the database without any field checking taking place. Data files having prerequisite loaded files contain dependent fields referenced in those data files (e.g., the *.BEI file contains event names also contained in the *.BED file).

MAR-D files load order dependencies.

Group	File	Prerequisite Files	Dependent Fields
Project[1]	.FAD		
	.FAT		
	.FAA		
	.FAY		
	.FAP		
	.FTT		
Attributes	.TTD		
	.CTD		
	.FMD		
	.LCD		
	.STD		
Basic Events	.BED		
	.BEI	.BED	Event Name
	.BEA	.BED	Event Name
		.CAD	Class Attr
		.CTD	Comp Type
		.FMD	Fail Mode
		.LCD	Location
		.STD	System Type
Fault Trees[2]	.FTD		
	.DLS		
	.FTL	.BED	Event Name
	.FTC	.FTD	Fault Tree Name
		.BED	Event Name
	.FTA	.FTD	Fault Tree Name
	.FTT	.FTD	Fault Tree Name
	.FTY		
	.PID		
Event Trees[3]	.ETD		
	.ETA	.ETD	Event Tree Name
		.BEI	Init Event
	.ETG	.ETD	Event Tree Name
	.ETL	.ETD	Event Tree Name
		.BEI	Init Event: unless Top

		.FTD	Fault Tree Name for TOPS
	.ETR		
	.ETY		
	.ETP		
	.ETT	.ETD	Event Tree Name
End States	.ESD		
	.ESI	????	????
	.EST	.ESD	End State Name
Sequences	.SQD	.ETD	Event Tree Name
	.SQC	.ETD	Event Tree Name
		.SQD	Sequence Name
	.SQA	.BED	Event Name
		.SQD	Sequence Name
		.ESD	End State Name
	.SQL		
	.SQY		
	.SQP		
	.SQT	.ETD	Event Tree Name
		.SQD	Sequence Name

NOTES:

Consistency checking for project names is active for all files.

.DLS and .FTL files are put into graphic relation instead of system relation. No cross relational cross checking available for graphics name.

.ETG is currently loaded as text. .ETL should look exactly like .ETG file.

If a dependent field does not exist in the database at load time, that field will be displayed in a confirmation/add menu. Thus, if an event name contained in a *.BEI file does not match any event name previously loaded by the *.BED file, the new name will be displayed with the option to add the event name or simply abort the load process. Note that the dependent field name cannot be edited at this point. If the name is not correct then exit, fix the mistake, and reload.

However, if you suspect your data files have multiple cross reference errors, load the files and allow the load procedure to find the errors for you. As an error is displayed on the dialog, write down the line number and file name and continue with the load. When the load is finished, make the necessary corrections to the data files, delete the database files for that project (i.e., the *.DAT, *.IDX, and *.BLK files), and restart the load procedure.

11.3.3 Extracting MAR-D Data

11.3.3.1 Extracting MAR-D Data

PURPOSE

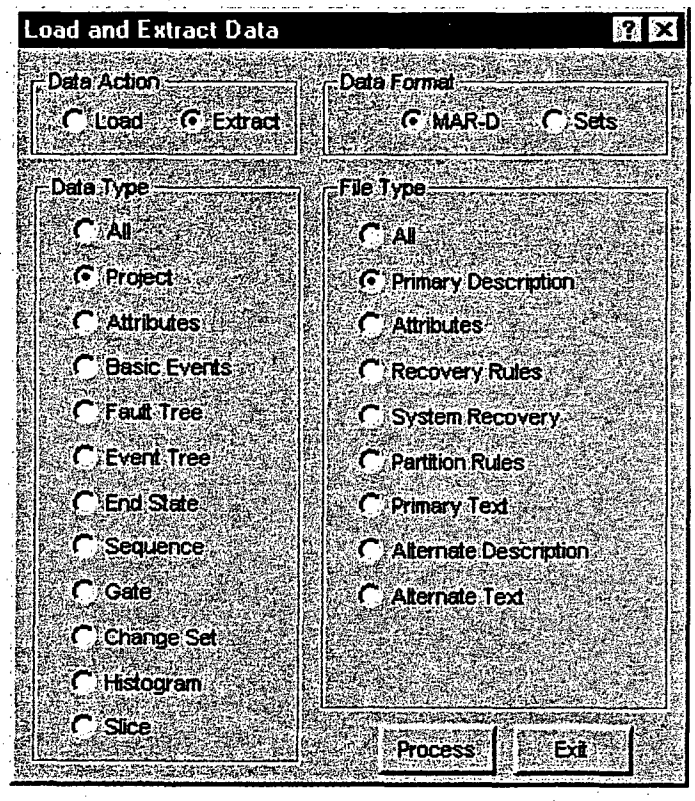
This option allows you to output data in MAR D file format. The extracted file is created in the current project folder (directory). The default output file name for description, information, and attribute data is the project name plus the appropriate extension. For MAR D cut set, logic, graphic, and textual information, the name of the file is generated based on the selected output option and some internal numbers.

NOTE: If a file with this name already exists it will be overwritten.

In general, the dialogs and operations are similar for each of the *Data Types*, therefore, a generic discussion is provided here.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose the **Load and Extract** option. The *Load and Extract Data* dialog will be displayed.



As you click the different *Data Type* radio buttons, corresponding *File Type* options will appear to the right, listing the types of data that can be loaded. Choose the data type and file type to output.

Usually, the *Extract* dialog will appear listing the records available for extraction. On occasion, when it is not necessary to select records, the *Output Destination* dialog will be displayed.

3. Click on the *Data Type* in the graphic, above, for additional information on various types of data.

11.3.3.2 Extract Dialog

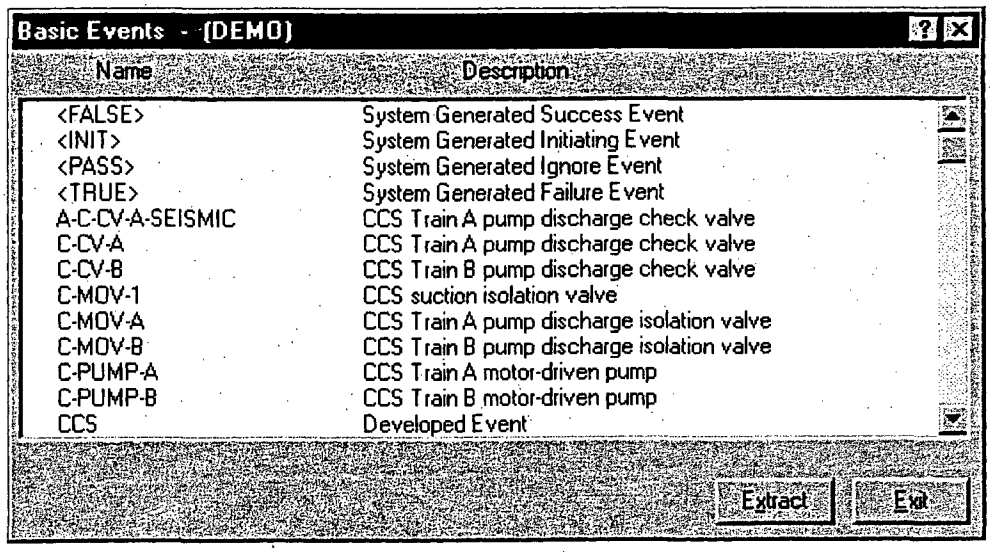
PURPOSE

Outputs information from the SAPHIRE database format to the generic format found in the MAR D database.

STEPS

1. Choose the desired *Data Type* and *File Type* radio buttons along with the **Extract** *Data Action* radio button.
2. Select the **Process** button. The *Extract* dialog will be displayed.
3. Highlight the desired record(s) and choose the **Extract** button.

This dialog lists the existing records in the current project database of the appropriate data type (depending on the *File Type* radio button previously selected) that can be extracted.



Extract - Output the data of the selected record(s) to a file in the current project subdirectory. After choosing this button, the *Output Destination* dialog will be displayed.

Exit - Close the *Extract* dialog without extracting data.

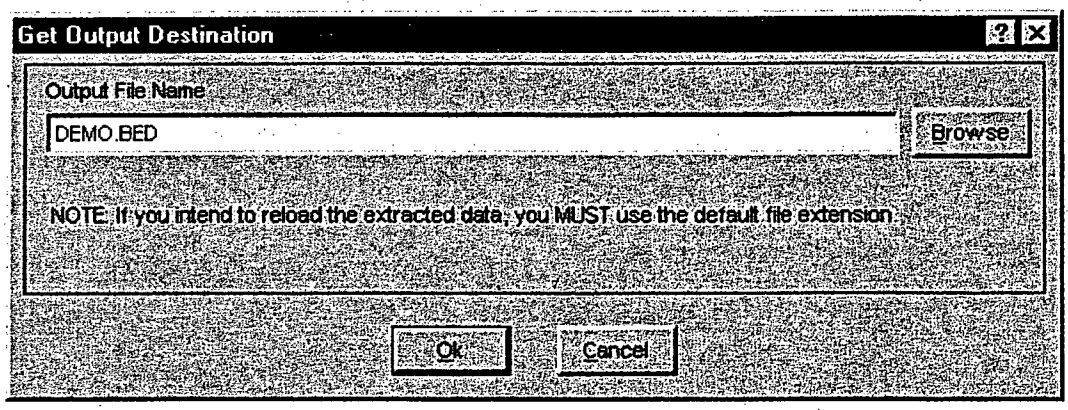
11.3.3.3 Output Destination Dialog

PURPOSE

Specify the output destination (file name) for extracted SAPHIRE data.

STEPS

1. Once the *Get Output Destination* dialog is displayed, enter the name of the file to which the data will be output.
2. Choose the **Ok** button.



Output File Name - The name of the file that will contain the extracted data. Each combination of data type and file type has a unique file extension. A default file name will be suggested. The file name can be changed; however, the default file extension should be used if the data is to be reloaded by SAPHIRE. If the output file name does not include a directory path, the output directory will be the same as the current project folder.

Browse – Select a folder and file name for the output.

Ok - Extract the data to the specified file. The file will be created in the current project directory. If successful, the message "Output complete", will be displayed in the status bar of the SAPHIRE window.

Cancel - Close the *Get Output Destination* dialog without extracting data.

Hint 1: By default, files will be extracted to the current project folder. However, it is recommended to extract files to a folder dedicated to load/extract files (a different dedicated folder for each project). In the **Browse** option, you can create a new folder by right clicking the file list and choosing **New | Folder**.

Hint 2: Once you have selected a load/extract folder that is different than the current project, that folder will be the initial directory selected each time you choose the **Browse** button. The most recently chosen folder will be remembered until SAPHIRE is closed.

11.3.4 All/Group Information

11.3.4.1 All/Group Data Type

11.3.4.1.1 Extract All Data Type

PURPOSE

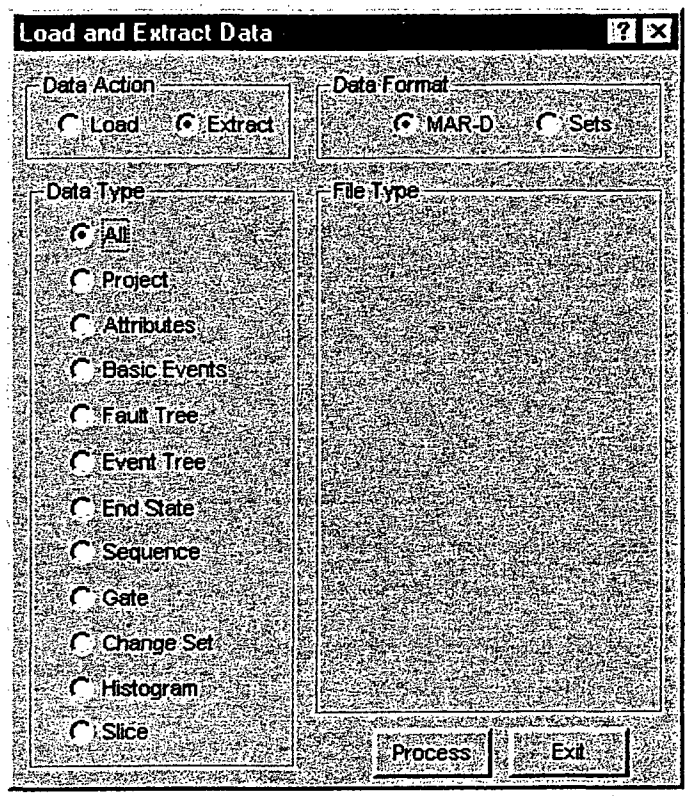
This option allows you to output all of the data contained in the current project into MAR D file format. This option is equivalent to selecting each combination of *Data Type* and *File Type*. The desired folder (directory) may be selected. The default output file names will be used. In addition, a master file will be created. This file will contain the names of all the files created. You may select the name of this file, but the default extension (.MARD) is expected if the data is to be reloaded into SAPHIRE.

NOTE 1: If the selected master file or any other files with default names already exist, they will be overwritten.

NOTE 2: Do not confuse this option with the **Project** *Data Type* option. The **All** option extracts every piece of data contained in the current project, whereas the **Project** option extracts only data directly related to the Project record, such as name, description, and project-wide rules.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose the **Load and Extract** option. The *Load and Extract Data* dialog will be displayed.
3. Select the **Extract** *Data Action*.
4. Select the **MAR-D** *Data Format*.
5. Select the **All** *Data Type*.



Process – Extract the current project data. The *Output Destination* dialog will be displayed.
Exit – Close the *Load and Extract Data* dialog.

*11.3.4.1.2 **Extract All** File Type*

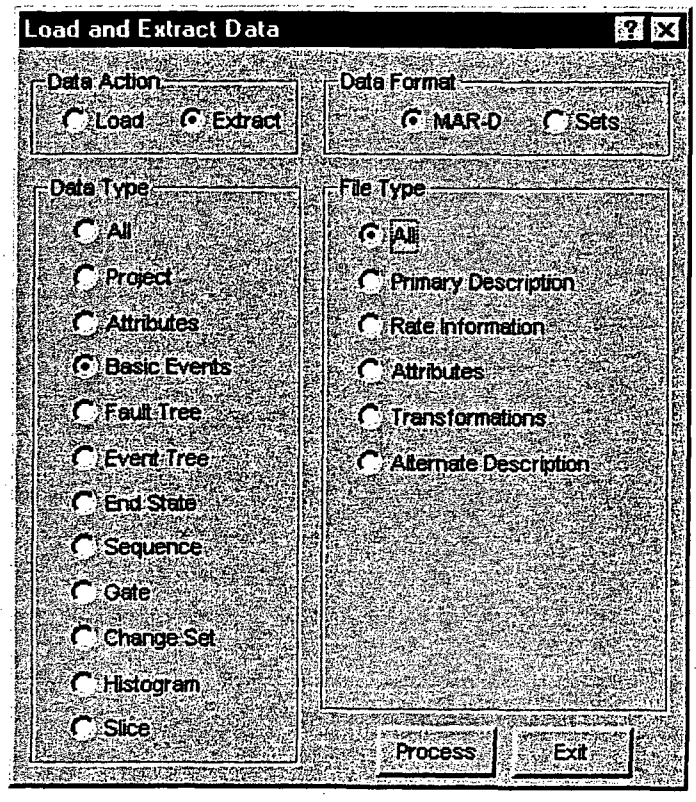
PURPOSE

This option allows you to output all of the data for a particular Data Type into MAR D file format. This option is equivalent to selecting each *File Type* option for the chosen *Data Type*. The desired folder (directory) may be selected. The default output file names will be used. In addition, a master file will be created. This file will contain the names of all the files created. You may choose the name of this file, but the default extension is expected if the data is to be reloaded into SAPHIRE.

NOTE: If the selected master file or any other files with default names already exist, they will be overwritten.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose the **Load and Extract** option. The *Load and Extract Data* dialog will be displayed.
3. Select the **Extract** *Data Action*.
4. Select the **MAR-D** *Data Format*.
5. Select the desired *Data Type*.
6. Select the **All** *File Type*.



Process – Extract the data. The *Output Destination* dialog will be displayed.
Exit – Close the *Load and Extract Data* dialog.

11.3.4.1.3 Load All/Group

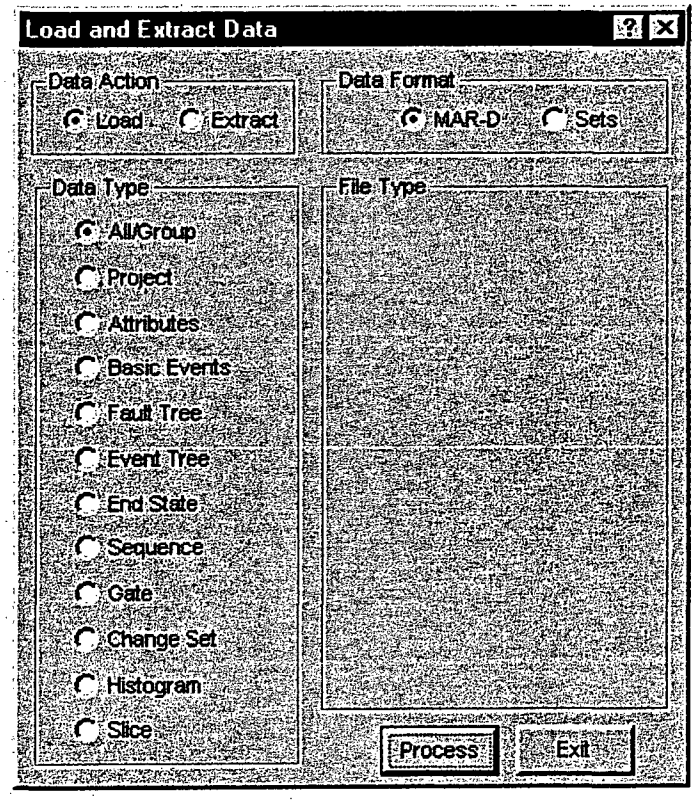
PURPOSE

This option facilitates loading of probabilistic risk assessment data from the Models and Results Data Base (MAR D). The process converts information from the generic format found in the MAR D database to the SAPHIRE database format. Use this option to load data that has been extracting using the corresponding **Extract All** *Data Type* option. You may also use this option to load data that was extracted using the **Extract All** *File Type* option, as well as for data extracted using the **Related Information** check box available in the **Fault Tree Logic** option.

NOTE: New data loaded into the database overwrites old data of the same name.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose the **Load and Extract** option. The *Load and Extract Data* dialog will be displayed.
3. Select the **Load** *Data Action*.
4. Select the **MAR-D** *Data Format*.
5. Select the **All/Group** *Data Type*.



Process – Load data into the current project. The Load dialog will be displayed.
Exit – Close the *Load and Extract Data* dialog.

11.3.4.1.4 Load All/Group Dialog

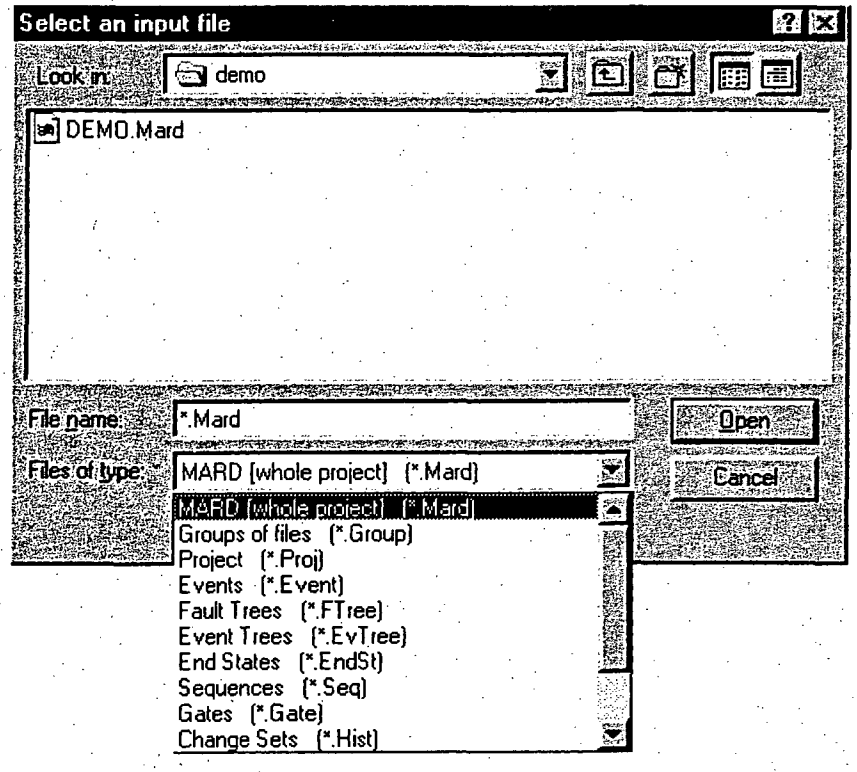
PURPOSE

Converts information from the generic format found in the MAR D database to the SAPHIRE database format. The *All/Group* option allows you to select a "master" file that contains a list of other MAR-D file formatted files to be loaded. SAPHIRE will load each of these files into the current project. If a file in the master list is not found, it will be ignored.

STEPS

1. Choose the **All/Group** *Data Type* and *File Type* radio buttons along with the **Load** *Data Action* radio button.
   OR
2. Choose the desired *Data Type* radio button and the **All** *File Type* radio button.
3. Select the **Process** button. The *Select an input file* dialog will be displayed.
4. Select the desired folder, if necessary.
5. Select the desired file type from the *Files of Type* drop down list, if necessary.
6. Highlight the desired file and choose the **Open** button.
   OR
7. Double click the desired file.

If you chose the **All/Group** *Data Type* option, the Files of Type drop down list will allow you to select any of the "master" list files. If you chose the **All** *File Type* option, only "master" list files for the selected *Data Type* will be available.



Open - Load the data from the selected "'master" file into the current project database. If successful, the message, "Load complete," will be displayed in the status bar of the SAPHIRE window.

Cancel - Close the *Select an input file* dialog without loading data.

11.3.5 Project Information

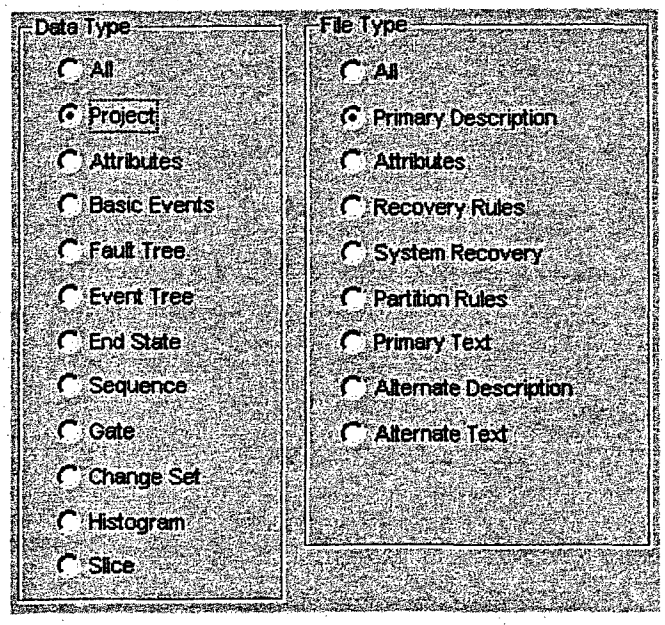
11.3.5.1 Project Information

PURPOSE

Project descriptions, attributes, and text can be loaded into the database or output from it. This information is currently stored in only the MAR D format. Note that each file contains data for only one project, and that a change in the data file project name will not change the database project name. When extracting, each file will contain information for the current project.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Load and Extract** from the menu. The *Load and Extract Data* dialog will be displayed.
3. Select the desired *Data Action*, **MAR-D** *Data Format*, **Project** *Data Type*, then the desired *File Type*.



Process - Continue the load or extract operation for the selected Data Type and File Type.
Cancel - Close the *Load and Extract Data* dialog.

11.3.5.2 Project Names and Descriptions

File Name:
xxxxxx.FAD

File Format:
name,description[,A]

where

name	24 character	Project name (first 8 characters must be unique).
description	60 character	Project description
A	1 character	If included indicates alternate description

11.3.5.3 Project Attribute File

File Name:
xxxxxx.FAA

File Format:
project=
name,mission,newSum,co,loc,type,design,vendor,AE,OpDate,QualDate

where

name	24 character	Project name
mission	Floating point	Default mission time in hours
newSum	Floating point	New sequence frequency sum
co	10 character	Company name
loc	16 character	Location name
type	3 character	Facility type

design	10 character	Facility design
vendor	5 character	Vendor name
AE	10 character	Architectural Engineer
OpDate	(yyyy/mm/dd)	Operational date
QualDate	(yyyy/mm/dd)	Qualification date

11.3.5.4 Project Recovery Rules

File Name:
xxxxxxxx.FAY

File Format:
project =

-- recovery rule text --

where

project	24 character	Project name

11.3.5.5 System Recovery Rules

File Name:
xxxxxxxx.FAS

File Format:
project =

-- recovery rule text --

where

project	24 character	Project name

11.3.5.6 Project Partition Rules

File Name:
xxxxxxxx.FAP

File Format:
project =

-- partition rule text --

where

project	24 character	Project name

11.3.5.7 Project Textual Information

File Name:
xxxxxx.FAT

File Format:
Project [,A] =

-- text --

where

project	24 character	Project name
A	1 character	If included indicates alternate description

11.3.6 Attributes Information

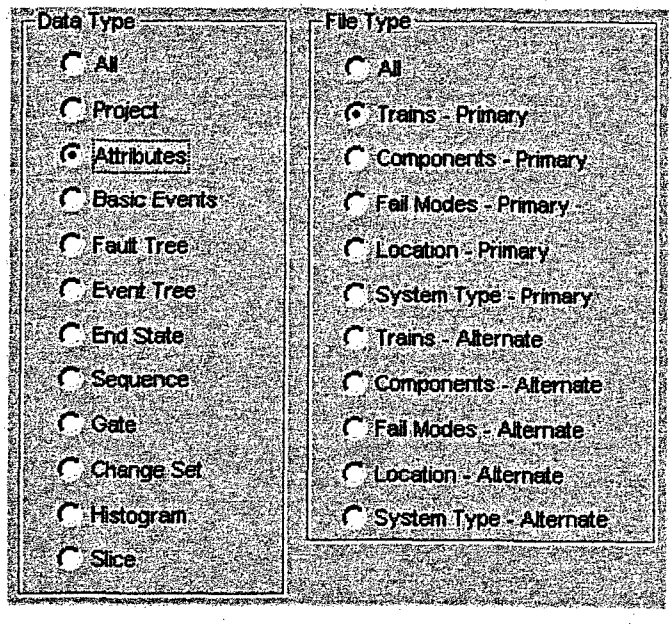
11.3.6.1 Event Attributes

PURPOSE

Basic event attributes descriptions (locations, failure modes, class attributes, system types and component types) can be loaded or extracted using the MAR D **Attributes** option. Event attributes will be output for the entire project.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Load and Extract** from the menu. The *Load and Extract Data* dialog will be displayed.
3. Select the desired *Data Action*, **MAR-D** *Data Format*, **Attributes** *Data Type*, then the desired *File Type*.



Process - Continue the load or extract operation for the selected Attributes File Type.
Cancel - Close the *Load and Extract Data* dialog.

11.3.6.2 Class Attribute Descriptions

File Name:
xxxxxx.TTD

File Format:
project =

attr,altAttr,description[,A]

. . . , . . .

where

project	24 character	Project name
attr	5 character	Class attribute primary name
altAttr	5 character	Class attribute alternate name
description	60 character	Class attribute description
A	1 character	If included indicates alternate description

11.3.6.3 Component Type Descriptions

File Name:

xxxxxx.CTD

File Format:

project =

comp, altComp, description [,A]

. . . , . . .

where

project	24 character	Project name
comp	5 character	Component type primary identifier
altComp	5 character	Component type alternate identifier
description	60 character	Component type description
A	1 character	If included indicates alternate description

11.3.6.4 Failure Mode Descriptions

File Name:

xxxxxx.FMD

File Format:

project =

fail,altFail,description[,A]

. . . , . . .

where

project	24 character	Project name
fail	5 character	Failure mode primary identifier
altFail	5 character	Failure mode alternate identifier
description	60 character	Failure mode description
A	1 character	If included indicates alternate description

11.3.6.5 Location Descriptions

File Name:

xxxxxx.LCD

File Format:

project =

loc,altLoc,description[,A]

. . . , . . .

where

project	24 character	Project name
loc	5 character	Location primary identifier
altLoc	5 character	Location alternate identifier
description	60 character	Location description
A	1 character	If included indicates alternate description

11.3.6.6 System Type Descriptions

File Name:

xxxxxx.STD

File Format:

project =

sys,altSys,description[,A]

. . . , . . .

where

project	24 character	Project name
sys	5 character	System primary identifier
altSys	5 character	System alternate identifier
description	60 character	System description
A	1 character	If included indicates alternate description

11.3.7 Basic Event Information

11.3.7.1 Basic Event Information

PURPOSE

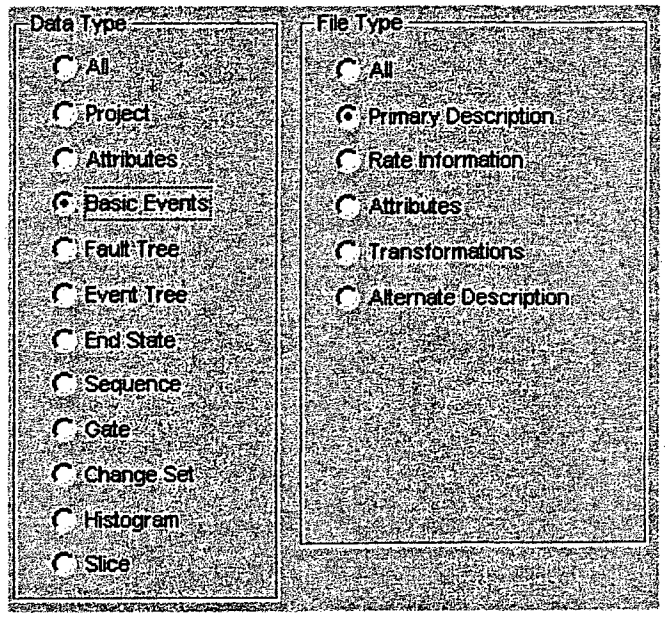
Event descriptions, failure rates, attributes, and transformations can be loaded into the database or extracted from it. When loading, the alternate name of the event defaults to the value of the primary name if an alternate name is not specified.

When extracting, if the *Use Alternate Basic Event Names* box is checked in the **Utility | Define Constants | Analysis** option, then the alternate name will be used instead of the primary name for descriptions and failure rates. Primary names will be output.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Load and Extract** from the menu. The *Load and Extract Data* dialog will be displayed.

3. Select the desired *Data Action*, **MAR-D** *Data Format*, **Basic Events** *Data Type*, then the *File Type*.



Process - Continue the load or extract operation for the selected Basic Events File Type.
Cancel - Close the *Load and Extract Data* dialog.

11.3.7.2 Basic Event Names and Descriptions

File Name:
xxxxxx.BED

File Format:
project =
name,description[,A]
. . . , . . .

where

project	24 character	Project name
name	24 character	Event primary name
description	60 character	Alphanumeric description
A	1 character	If included indicates alternate description

11.3.7.3 Basic Event Failure Rate Information

The basic event failure rates are stored in the Event relation.

File Name:
xxxxxx.BEI

File Format:

project =

name, calc, udC, udT, udV, prob, lambda, tau, mission, init, Flag, udV2

. . . , . . . , . . . , . . . , . . . , . . . , . . . , . . . , . . . , . . .

where

Project	24 character	Project name
Name	24 character	Basic event name
Calc	1 character	Calculation type

1	Probability
2	Lambda * Mission Time
3	1 Exp(Lambda * Mission Time)
4	Lambda * Min(Mission Time, Tau)
5	Operating component with full repair
6	Lambda * Tau / 2.0
7	1+(EXP(Lambda*Tau) 1.0)/(Lambda*Tau)
8	Base Probability * Probability
9	Base Probability * Probability
T	Set to House Event (Failed, Prob=1.0)
F	Set to House Event (Successful,Prob=0.0)
I	Set to ignore
S	Use fault tree mincut upperbound
G	Seismic event - Enter g level for screening
M	Seismic event - Use medium site hazard curve for screening

UdC	4 characters	Uncertainty correlation class Events in same class are 100% correlated.
UdT	1 character	Uncertainty distribution type

L	Log normal, error factor
N	Normal, standard deviation
B	Beta, b of Beta(a,b)
G	Gamma, a Gamma(a)
C	Chi-squared, degrees of freedom
E	Exponential, none
U	Uniform, Upper end pt.
H	Histogram
M	Maximum entropy

UdV	Floating point	Uncertainty distribution value
Prob	Floating point	Probability value
Lambda	Floating point	Basic event failure rate per hr.
Tau	Floating point	Time to repair in hours
Mission	Floating point	Mission time
init	Boolean	Initiating event flag (Y/N)
Flag	1-character	process flag

udV2	Floating point	Uncertainty distribution value #2

General Rules:

The name field is mandatory.

11.3.7.4 Basic Event Attribute Codes

Basic event attributes are stored in the Event relation.

File Name:
xxxxxx.BEA

File Format:
project =

name,Aname,type,sys,fail,loc,compID,Gname,train,att1,..,att16

. . . , . . . , . . . , . . . , . . . , . . . , . . . , . . . , . . . , . . .

where

project	24 character	Project name
name	24 character	Event name
Aname	24 character	Alternate event name
type	3 character	Event component type
sys	3 character	Event component system
fail	3 character	Failure mode
loc	3 character	Component location
compID	7 character	Component ID
Gname	24 character	Event group identifier
train	3 character	Train identifier
att1..att16	Class attribute flags	16 values of Y or N (yes or no) indicate whether the attribute described in the class attribute file is applicable.

General Rules:

The name field is mandatory.

11.3.7.5 Basic Event Transformations

Basic event attributes are stored in the Event relation.

Site Name:
xxxxxx.BET

File Format:
project =

name1,level,type

bename1, bename2, . . . ,

. . . , benameN

^EOS

name2,level,type

bename1, bename2, . . . ,

. . . , benameN

^EOS

where

project	24 character	Project name
name	24 character	Event name
Type	4 character	Transformation type
level	3 character	Transformation level
bename1..N	24 character	Event name

11.3.8 Fault Tree Information

11.3.8.1 Fault Tree Information

PURPOSE

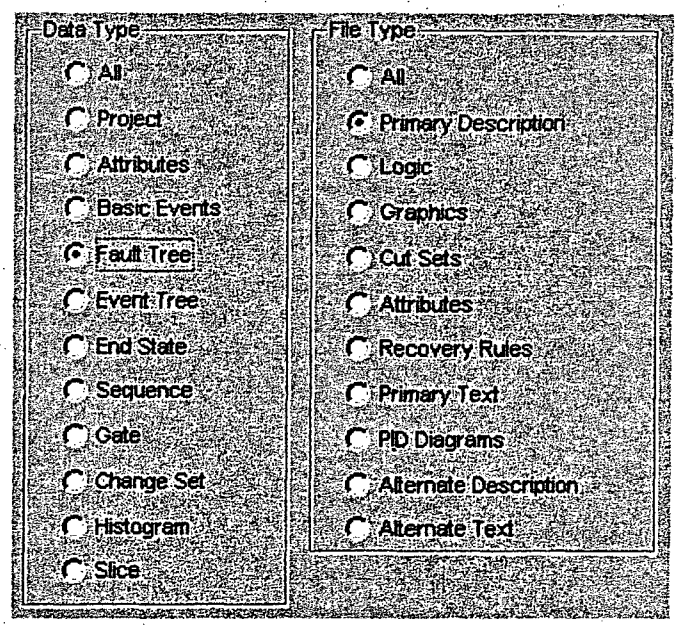
Fault tree descriptions, graphics, logic, cut sets, attributes, text and piping and instrumentation diagrams (P&IDs) can be loaded into the database or output to files.

Fault tree cut sets and attributes data can be loaded into either the base case (permanent) or current (temporary) field areas. MAR-D database applications will normally use the base case load option because only permanent data should be loaded into the database. Analysis software such as SAPHIRE will use current case fields for comparisons of changed value results. Any basic event found within the fault tree logic or cut sets will be added to the Event relation.

For some of the *File Types*, if more than one of the fault trees are selected for extraction, the *File Output* dialog is displayed, allowing you to choose the manner in which files are created for output.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Load and Extract** from the menu. The *Load and Extract Data* dialog will be displayed.
3. Select the desired *Data Action*, **MAR-D** *Data Format*, **Fault Tree** *Data Type*, then the desired *File Type*.

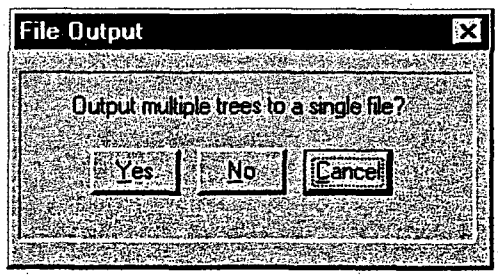


Process - Continue the load or extract operation for the selected Fault Tree File Type.
Cancel - Close the *Load and Extract Data* dialog.

11.3.8.2 File Output Dialog

PURPOSE

This option provides you with the opportunity to choose the method for extracting multiple fault trees, event trees, or end states.



Yes - The output for all selected list items will be saved in one file, with the items separated by ^EOS. The *Get Output Destination* dialog will be displayed.

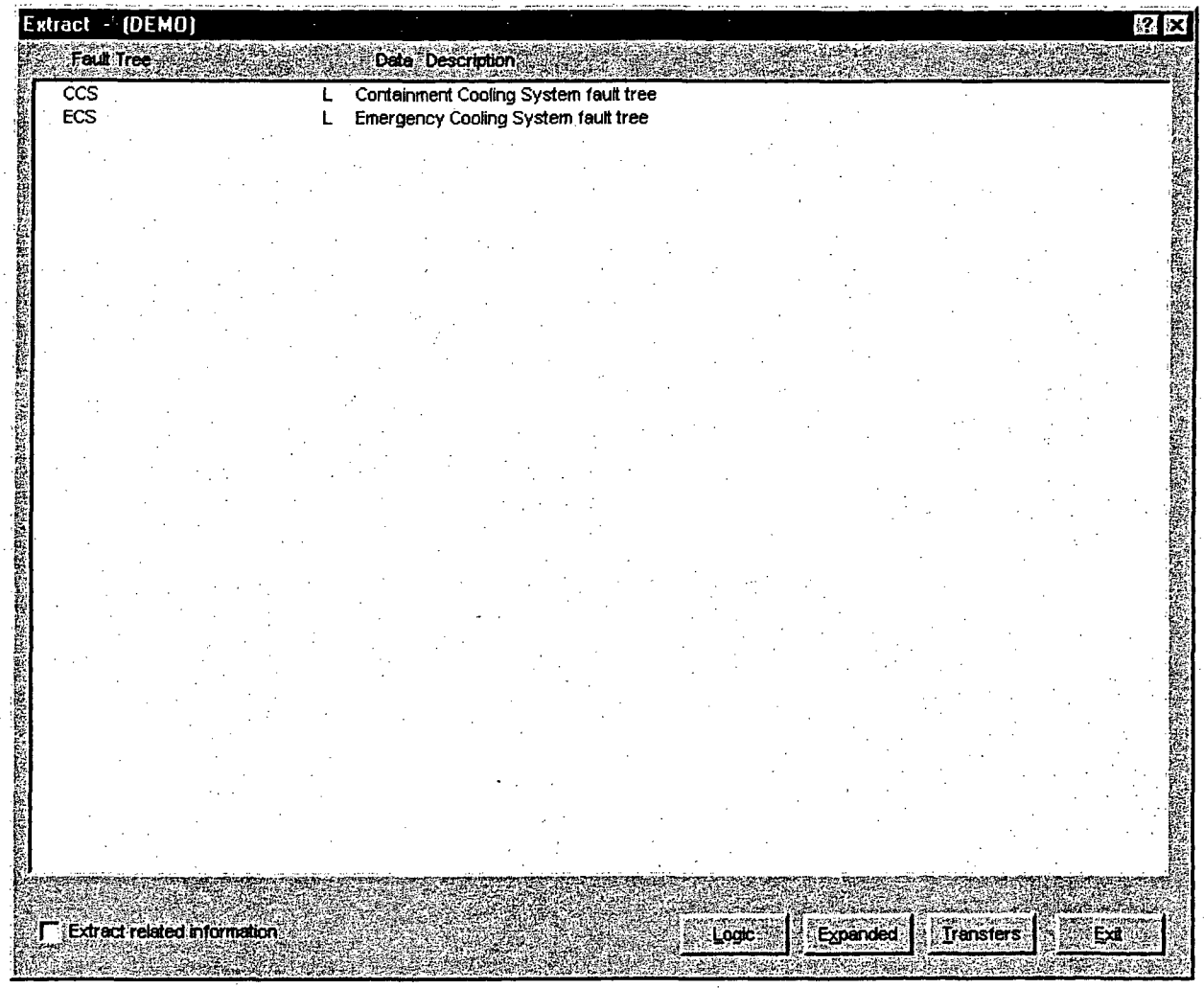
No -

Cancel - Close the *File Output* dialog without extracting data.

11.3.8.3 Fault Tree Logic Options

PURPOSE

This option provides you with the opportunity to choose the way the logic will be extracted.



When you choose a Fault Tree Logic extraction method, either the *Get Output Destination* or the Select Directory dialog will be displayed. The File Output dialog may also be displayed.

Extract related information – When this is checked, related basic event and gate information for the selected fault trees are also extracted into their respective MAR-D file formats. A .Group master file is created, which contains a list of all files created. This .Group file can be used to load all of the related information back into SAPHIRE at once.

Logic – Extract the logic "as is" for all selected items. Transfer gates will not be expanded.

Expanded – Extract the expanded logic for all selected items. All transfer gates will be replaced with the complete transfer information. Paging information will be lost.

Transfers – Extract the logic for all selected items, preserving transfer gates, but automatically extracting the transfer fault trees, whether they were explicitly marked or not. This option is a hybrid of the Logic and Expanded options. It both preserves paging information, and includes the transfer information.

Exit - Close the *Extract* Fault Trees dialog without extracting data.

11.3.8.4 Fault Tree Names and Descriptions

File Name:

xxxxxx.FTD

File Format:

project =

name,description[,s][,A]

. . . , . . .

where

project	24 character	Project name
name	24 character	Fault tree primary name
description	60 character	Fault tree description
s	1 character	If included indicates fault tree is a sub-tree
A	1 character	If included indicates alternate description

11.3.8.5 Fault Tree Logic

Fault tree logic is stored in the block data file of the System relation.

File Name:

xxxxxx.FTL

File Format:

project, fault tree =

* gatename1,description

gatename1 gatetype input1 input2 . . . inputn

. . . . . .

* gatenamen,description

gatenamen gatetype input1 input2 . . . inputn

. . .

where

project		24 character	Project name
fault tree		24 character	Fault tree name
gatename		24 character	Gate name
gatetype		4 character	Gate type
	AND		logical AND
	OR		logical OR
	TBL		table of events
	TRAN		transfer followed by a 24-character fault tree name
	NAND		logical NOT AND
	NOR		logic NOT OR
	N/M		N out of M logic gate

CONT		continuation of inputs to the previous gate
input	24 character	inputs to the gate (event or gate names)
description	60 character	gate name descriptions included as comment

General Rules:

A gate definition cannot exceed 255 characters.

A line beginning with an asterisk (*) is a comment.

For each gate name a comment should be included giving the gate description.

11.3.8.6 Fault Tree Graphics

Fault tree graphics are stored in the block data file of the System relation. The MAR D file (.DLS) is a display list sequence for the graphics in a binary format. It is loaded and output as is with no conversion performed.

File Name:

xxxxxx.DLS

File Format:

IRRAS 2.5/4.0/5.0, SAPHIRE 6.0 and 7.0 Fault Tree Graphics file (DLS format)

11.3.8.7 Fault Tree Cut Sets

The fault tree cut sets are stored in the System relation in the block data file.

File Name:

xxxxx.FTC

File Format:

project, fault tree, analysis =

eventname * eventname +

eventname * eventname * eventname *

eventname +

eventname * eventname.

^EOS

project, fault tree2 =

where

project	24 character	Project name
fault tree	24 character	Fault tree name
analysis	1 character	Analysis type
1		Random
2		Fire
3		Flood
4		Seismic

5 through 8			Reserved
9 through 16			user-defined
eventname	24 character	Event names in the cut set	

General Rules:

An asterisk (*) separates cut set events. Spaces are ignored.

A plus sign (+) separates cut sets.

A period (.) denotes the end of a sequence.

A slash (/) precedes complemented events.

Event names are a maximum of 4 characters including the "/".

A line beginning with an asterisk (*) is a comment.

11.3.8.8 Fault Tree Attributes

File Name:

xxxxx.FTA

File Format:

project, analysis =

name,level,mission,mincut,proCut,sample,seed,sizCut,sys,cuts, events,value1,..,value9

. . . , . . . , . . . , . . . , . . . , . . . , . . . , . . . , . . . , . . . ,

where

project	24 character	Project name	
analysis	1 character	Analysis type	
1			Random
2			Fire
3			Flood
4			Seismic
5 through 8			Reserved
9 through 16			user-defined
name	24 character	Fault tree name	
level	Integer 2	0 = top level tree	
mission	Floating point	Mission time	
mincut	Floating point	Mincut upper bound	
proCut	Floating point	Probability cut off value	
sample	Integer 4	Sample size	
seed	Integer 8	Random number seed	
sizecut	Integer 2	Size cut off value	

sys	3 character	System identifier
cuts	Integer 5	Base number of cut sets
events	Integer 5	Base number of events
value	Floating point	Base uncertainty values

11.3.8.9 Fault Tree Recovery Rules

File Name:
xxxxxxxx.FTY

File Format:
project =

-- recovery rule text --

where

project	24 character	Project name

11.3.8.10 Fault Tree Textual Information

File Name:
xxxxxx.FTT

File Format:
project, fault tree [,A]=

-- text --

^EOS

project, fault tree2 =

. . .

where

project	24 character	Project name
fault tree	24 character	Fault tree name
A	1 character	If included indicates alternate text

11.3.8.11 Fault Tree Graphical P&ID

File Name:
xxxxxxxx.PID

File Format:
IRRAS 4.0/5.0, SAPHIRE 6.0 P&ID Graphics file (PID Format)

11.3.9 Event Tree Information

11.3.9.1 Event Tree Information

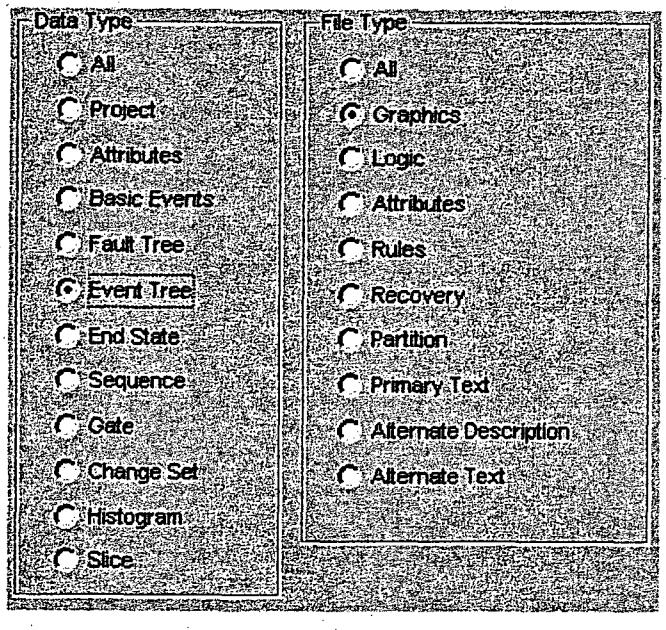
PURPOSE

Event tree descriptions, graphics, logic, attributes, rules, and text can be loaded into the database or output from it.

For some of the *File Types*, if more than one of the event trees are selected for extraction, the *File Output* dialog is displayed, allowing you to choose the manner in which files are created for output.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Load and Extract** from the menu. The *Load and Extract Data* dialog will be displayed.
3. Select the desired *Data Action*, **MAR-D** *Data Format*, **Event Tree** *Data Type*, then the desired *File Type*.

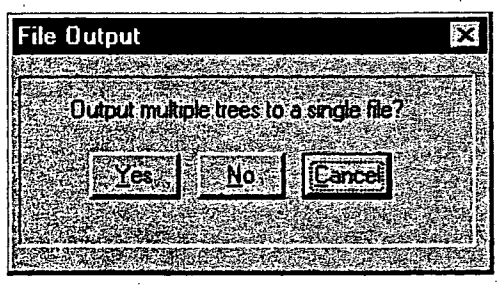


Process - Continue the load or extract operation for the selected Event Tree File Type.
Cancel - Close the *Load and Extract Data* dialog.

11.3.9.2 File Output Dialog

PURPOSE

This option provides you with the opportunity to choose the method for extracting multiple fault trees, event trees, or end states.



Yes - The output for all selected list items will be saved in one file, with the items separated by ^EOS. The *Get Output Destination* dialog will be displayed.

No -

Cancel - Close the *File Output* dialog without extracting data.

11.3.9.3 Event Tree Names and Descriptions

File Name:

xxxxxx.ETD

File Format:

project =

name,description[,s][,A]

. . . , . . .

where

Project	24 character	Project name
Name	24 character	Event tree name
Description	60 character	Event tree description
S	1 character	If included indicates event tree is a fault tree
A	1 character	If included indicates alternate description

11.3.9.4 Event Tree Graphics

The SAPHIRE Event Tree Graphics file (*.ETG) is a display list sequence for the graphics. Its format and contents are the same as the Event Tree Logic File.

File Name:

xxxxxx.ETG

File Format:

See file format for the Event Tree Logic

11.3.9.5 Event Tree Logic

File Name:

xxxxxx.ETL

File Format:

project, event tree, init event [,T] =

^TOPS

```
* 1 | 2 | 3 | 4 | 5 | this is a comment
ABCDE BCDEF CDEFG DEFGH EFGHI
^LOGIC
+1 +2 3 +4 +5
   5
  4 5
 2 +3 4 5
  3 +4 +5
   5
  4 5
1 +2 3 4 5
 2 +3 +4 5
  4 5
  4 5
  4 5
  3 4 5

^SEQUENCES
 Y/N, header#1,      Y/N, header#2,      Y/N, header#3,     Y/N,header#4
 Y/N, sequence#1,    Y/N, end state#1,   Y/N, xdata1#1,    Y/N,xdata2#1
 Y/N, sequence#2,    Y/N, end state#2,   Y/N, xdata1#2,    Y/N,xdata2#2
 Y/N, sequence#3,    Y/N, end state#3,   Y/N, xdata1#3,    Y/N,xdata2#3
 Y/N, sequence#4,    Y/N, end state#4,   Y/N, xdata1#4,    Y/N,xdata2#4
 Y/N, sequence#5,    Y/N, end state#5,   Y/N, xdata1#5,    Y/N,xdata2#5
 Y/N, sequence#6,    Y/N, end state#6,   Y/N, xdata1#6,    Y/N,xdata2#6
 Y/N, sequence#7,    Y/N, end state#7,   Y/N, xdata1#7,    Y/N,xdata2#7
 Y/N, sequence#8,    Y/N, end state#8,   Y/N, xdata1#8,    Y/N,xdata2#8
 Y/N, sequence#9,    Y/N, tran file#9,   Y/N, xdata1#9,    Y/N,xdata2#9, T
 Y/N, sequence#10,   Y/N, end state#10,  Y/N, xdata1#10,   Y/N,xdata2#10
 Y/N, sequence#11,   Y/N, end state#11,  Y/N, xdata1#11,   Y/N,xdata2#11
 Y/N, sequence#12,   Y/N, end state#12,  Y/N, xdata1#12,   Y/N,xdata2#12
 Y/N, sequence#13,   Y/N, end state#13,  Y/N, xdata1#13,   Y/N,xdata2#13

^TEXT
SIZE s
JUST j
COLOR j
XY xvalue,yvalue
"60 character line of text"
```

XY xvalue, yvalue

"60 character line of text"
"60 character line of text"

^PARMS

START yvalue

WINDOW x1,y1,x2,y2

HEADER x1,x2,x3,x4

^EOS

project, event tree2 =

(additional event trees)

where

project	24 character	Project name
name	24 character	Event tree name
init event	24 character	Initiating Event
[,T]	1 character	Optional flag indicating init event name is a Top event fault tree
TOPS	24 character	Top event/fault tree names
Y/N	Boolean	End state text displayed?
header	24 character	Sequence header
sequence	24 character	Sequence name
endstate	24 character	End state name
tran file	24 character	Name of transfer file
xdata1	24 character	Information (optional)
xdata2	24 character	Information (optional)

General Rules:

A line beginning with an asterisk (*) is a comment.

Literal "^TOPS", "^LOGIC", "^SEQUENCES" labels must be present.

Logic is built according to the position of the top event in the definition.

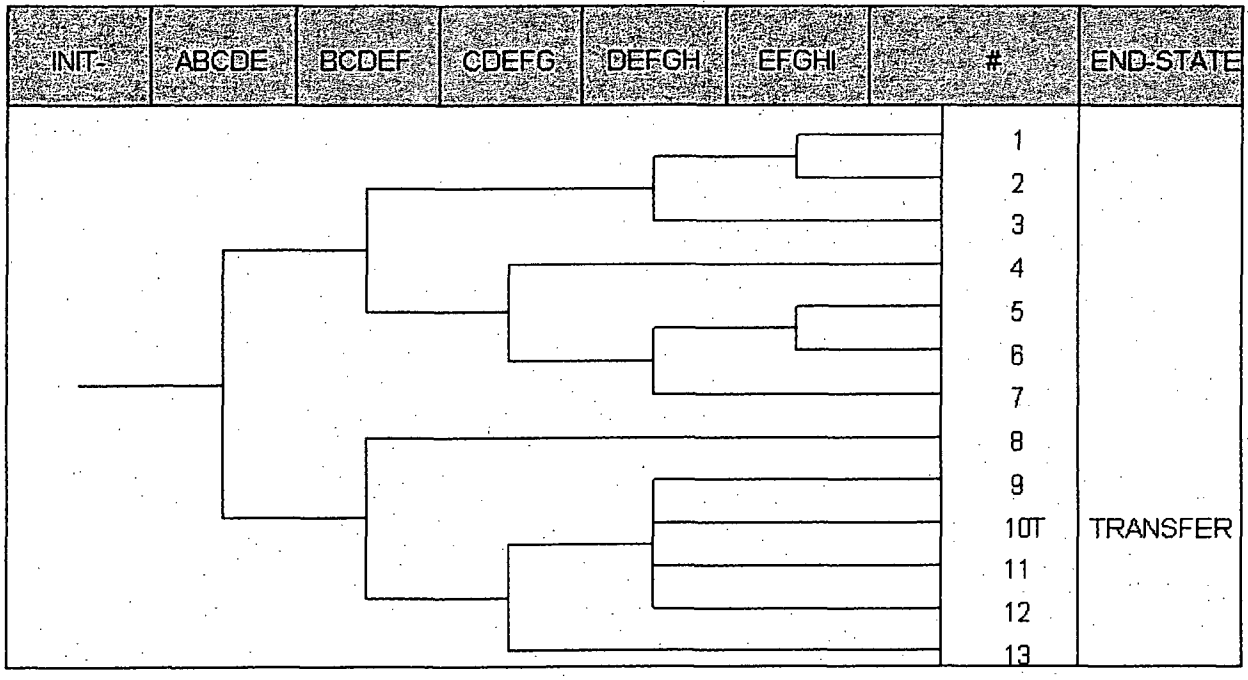
Plus sign (+)---the specified top event succeeded.
Minus sign ( )---the specified top event failed.
Blank ( )---the response of the indicated top event did not matter.

Header, Sequence name, End State name, Xdata1, Xdata fields associated with each sequence. "Y/N" indicates whether the specified field is visible. A "T" at the end indicates the sequence transfers to another tree.

User text is input following the ^TEXT command. Parameters include the size, justification, color, and location of the text block.

The ^PARMS command allows input of program control parameters.

11.3.9.6 Sample Graphical Event Tree



11.3.9.7 Event Tree Attributes

File Name:

xxxxxx.ETA

File Format:

project =

name,init

. . . , . . .

where

project	24 character	Project name
name	24 character	Event tree name
init event	24 character	Initiating Event

11.3.9.8 Event Tree Rules

File Name:

xxxxxxxx.ETR

File Format:

project, event tree =

-- event tree rule text

. . .

^EOS

project, event tree2

where:

project	24 character	Project name
name	24 character	Event tree name
tops	24 character	Top event/fault tree names

11.3.9.9 Event Tree Recovery Rules

File Name:

xxxxxxxx.ETY

File Format:

project, event tree =

-- recovery rule text --

^EOS

project, event tree2 =

where

project	24 character	Project name
event tree	24 character	Event tree name

11.3.9.10 Event Tree Partition Rules

File Name:

xxxxxxxx.ETP

File Format:

project, event tree =

-- partition rule text --

^EOS

project, event tree2 =

where

project	24 character	Project name
event tree	24 character	Event tree name

11.3.9.11 Event Tree Textual Information

File Name:

xxxxxx.ETT

File Format:

project, event tree [,A]=

-- text --

^EOS

project, event tree2 =

-- text --

where

project	24 character	Project name
event tree	24 character	Event tree name
A	1 character	If included indicates alternate description

11.3.10 End State Information

11.3.10.1 End State

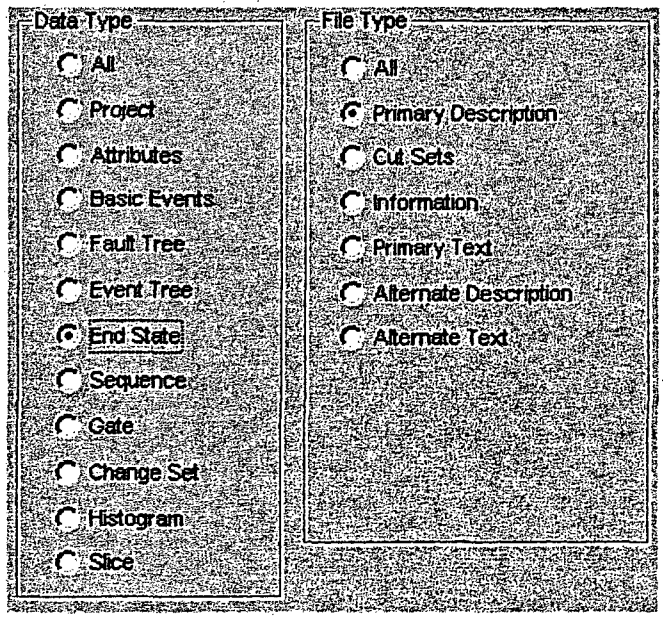
PURPOSE

End state names, descriptions, cut sets, and text can be loaded into the database or output from it. Each sequence can be tied to a single plant damage state. The cut sets for a sequence can be partitioned to map to separate end state. The name and description data are loaded with the *.PDS file.

For some of the *File Types*, if more than one of the end states are selected for extraction, the *File Output* dialog is displayed, allowing you to choose the manner in which files are created for output.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Load and Extract** from the menu. The *Load and Extract Data* dialog will be displayed.
3. Select the desired *Data Action*, **MAR-D** *Data Format*, **End State** *Data Type*, then the desired *File Type*.



Process - Continue the load or extract operation for the selected End State File Type.
Cancel - Close the *Load and Extract Data* dialog.

11.3.10.2 End State Names and Descriptions

File Name:

xxxxxx.ESD

File Format:

project =

name,description[,A]

. . . , . . .

where

project	24 character	Project primary name
name	24 character	End state primary name
description	60 character	End state description
A	1 character	If included indicates alternate description

11.3.10.3 End State Cut Sets

The end state cut sets are the minimal cut sets for end state logic as derived from the fault tree logic. The cut sets are stored in the block data file of the Endstate relation.

The MAR D end state cut sets are in a format similar to that of the fault tree cut sets.

File Name:

xxxxxx.ENC

File Format:

project, event tree, end state =

eventname * eventname +

eventname * eventname * eventname *

eventname +

eventname * eventname.

^EOS

project, event tree2, end state =

where

project	24 character	Project name
event tree	24 character	Event tree name
end state	24 character	End state name
eventname	24 character	Event names in the cut set

General Rules:

An asterisk (*) separates events in a cut set. Spaces are ignored.

A plus sign (+) separates cut sets.

A period (.) denotes the end of the sequence.

A slash (/) precedes complemented events.

Event names have a maximum of 16 characters including the "/" character for complemented events.

A line beginning with an asterisk (*) is a comment.

11.3.10.4 End State Information

File Name:

xxxxxx.ESI

File Format:

project =

project =

Name, E-QMethod, E-QPasses, R-QMethod, R-QPasses,

. . . . . , ... , ... , ... , ... ,

where

project	24 character	Project name
name	24 character	End state name
e-Qmethod	1 character	End state default quantification method
e-Qpasses	Integer 3	End state default min/max quantification passes
r-QMethod	1 character	Quantification method used for current results
r-Qpasses	Integer 3	Min/max quantification passes used for current results

11.3.10.5 End State Textual Information

A separate file is created for each end state selected.

File Name:

end-state.EST

where

end-state	24 character	End state name

File Format:

project, end state[, A]=

-- text --

where

project	24 character	Project name
end state	24 character	End state name
A	1 character	If included indicates alternate description

11.3.11 Sequence Information

11.3.11.1 Sequence

PURPOSE

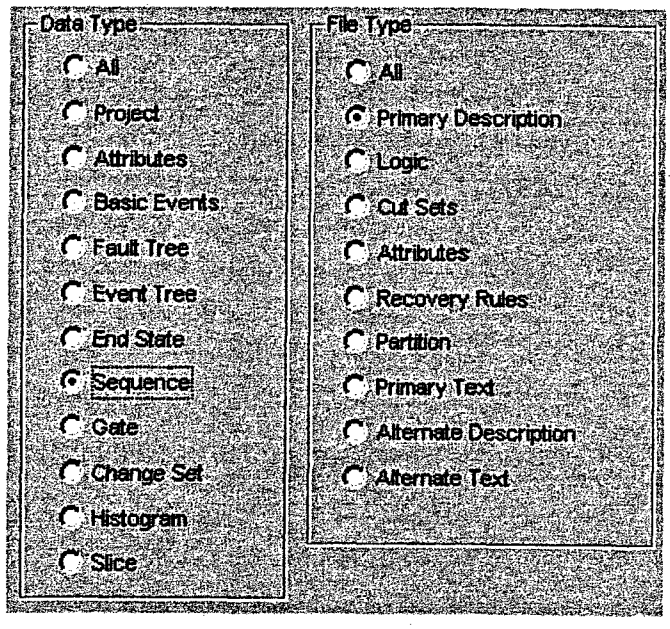
Sequence cut sets, descriptions, attributes, text, and logic for MAR D file formats can be loaded or output.

With cut sets and attributes, data can be loaded into either the base case or current field areas. The event tree initiating event must have been entered into the database prior to loading sequence cut sets and, therefore, does not need to be included; any found will be removed from the cut set data.

When extracting, if more than one of the sequences are selected, the *Output* dialog is displayed, allowing you to choose the manner in which files are created for output.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Load and Extract** from the menu. The *Load and Extract Data* dialog will be displayed.
3. Select the desired *Data Action*, **MAR-D** *Data Format*, **Sequence** *Data Type*, then the desired *File Type*.

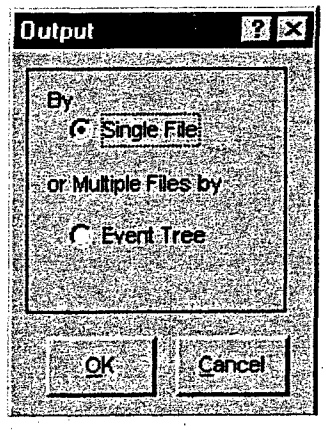


Process - Continue the load or extract operation for the selected Sequence File Type.
Cancel - Close the *Load and Extract Data* dialog.

11.3.11.2 Output Dialog

PURPOSE

This option provides you with the opportunity to choose the method for extracting multiple sequences. When extracting descriptions, attributes, or text, the **Multiple Files by Sequence** option is not available.



Single File - The output for the selected sequences will be saved in one file. The file name will be the first eight letters of the project name with the appropriate file extension.

Event Tree - The selected sequences will be grouped by event tree and output to individual event tree files. The names of the event tree files will be based on the event tree names and the appropriate file extension.

OK - Extract the data to the specified file(s). The file will be created in the current project directory. If successful, the message, "Output complete," will be displayed in the menu of the SAPHIRE window.

Cancel - Close the *Output* dialog without extracting data.

11.3.11.3 Sequence Names and Descriptions

File Name:

xxxxxx.SQD

File Format:

project,eventree =

name,description[,A]

. . . , . . .

^EOS

where

project	24 character	Project name
event tree	24 character	Event tree name
name	24 character	Sequence name
description	60 character	Sequence description
A	1 character	If included indicates alternate description

11.3.11.4 Sequence Logic

File Name:

xxxxxxxx.SQL

File Format:

project, event tree, sequence=

sys1 sys2 /sys3 sys4

. . .

^EOS

project, event tree2, sequence2=

where

project	24 character	Project name
event tree	24 character	Event tree name
sequence	24 character	Sequence name
sys	24 character	Fault tree name

General Rules:

Complemented fault trees are prefixed with "/".

11.3.11.5 Sequence Cut Sets

The sequence cut sets are the minimal cut sets for sequence logic as derived from the fault tree logic. The cut sets are stored in the block data file of the Sequence relation.

The MAR D sequence cut sets (.SQC) are in a format similar to that of the fault tree cut sets..

File Name:

xxxxxx.SQC

File Format:

project, event tree, sequence, analysis =

eventname * eventname +

eventname * eventname * eventname *

eventname +

eventname * eventname.

^EOS

project, event tree2, sequence2 =

where

project	24 character	Project name	
event tree	24 character	Event tree name	
sequence	24 character	Sequence name	
analysis	1 character	Analysis type	
1			Random
2			Fire
3			Flood
4			Seismic
5 through 8			Reserved
9 through 16			user-defined
eventname	24 character	Event names in the cut set	

General Rules:

An asterisk (*) separates events in a cut set. Spaces are ignored.

A plus sign (+) separates cut sets.

A period (.) denotes the end of the sequence.

A slash (/) precedes complemented events.

Event names have a maximum of 24 characters including the "/" character for complemented events.

A line beginning with an asterisk (*) is a comment.

11.3.11.6 Sequence Attributes

File Name:

xxxxxx.SQA

File Format:

project, event tree, analysis =

name,endstate,mincut,mission,procut,sample,seed,size,cuts,

events,value1, . . . ,value9,default flags, used flags

. . . , . . . , . . . , . . . , . . . , . . . , . . . , . . . , . . .

^EOS

project, event tree2 =

where

project		24 character	Project name
event tree		24 character	Event tree name
analysis		1 character	Analysis type
	1		Random
	2		Fire
	3		Flood
	4		Seismic
	5 through 8		Reserved
	9 through 16		user-defined
name		24 character	Sequence name
endstate		24 character	End State name
mincut		Floating point	Mincut upper bound
mission		Floating point	Mission time in hours
procut		Floating point	Probability cut off value
sample		Integer 4	Sample size
seed		Integer 8	Random number seed
size		Integer 2	Size cut off value
cuts		Integer 5	Base number of cut sets

events	Integer 5	Base number of events
value	Floating point	Base uncertainty values
value1		5th percentile
value2		Median
value3		Mean
value4		95th percentile
value5		Minimum sample
value6		Maximum sample
value7		Standard deviation
value8		Skewness
value9		Kurtosis
Default flags	24 character	Default flag set for this sequence
Used flags	24 character	Flag set used to generate these cut sets

11.3.11.7 Sequence Recovery Rules

File Name:

xxxxxxxx.SQY

File Format:

project, event tree, sequence =

-- recovery rule text --

^EOS

project, event tree, sequence2 =

where

project	24 character	Project name
event tree	24 character	Event tree name
sequence	24 character	Sequence name

11.3.11.8 Sequence Partition Rules

File Name:

xxxxxxxx.SQP

File Format:

project, event tree, sequence =

-- partition rule text --

^EOS

project, event tree, sequence2 =

where

project	24 character	Project name
event tree	24 character	Event tree name
sequence	24 character	Sequence name

11.3.11.9 Sequence Textual Information

File Name:

xxxxxx.SQT

File Format:

project, event tree, sequence[, A]=

--- text ---

^EOS

project, event tree2, sequence2=

--- text ---

where

project	24 character	Project name
sequence	24 character	Sequence name
event tree	24 character	Event tree name
A	1 character	If included indicates alternate description

11.3.12 Gate Information

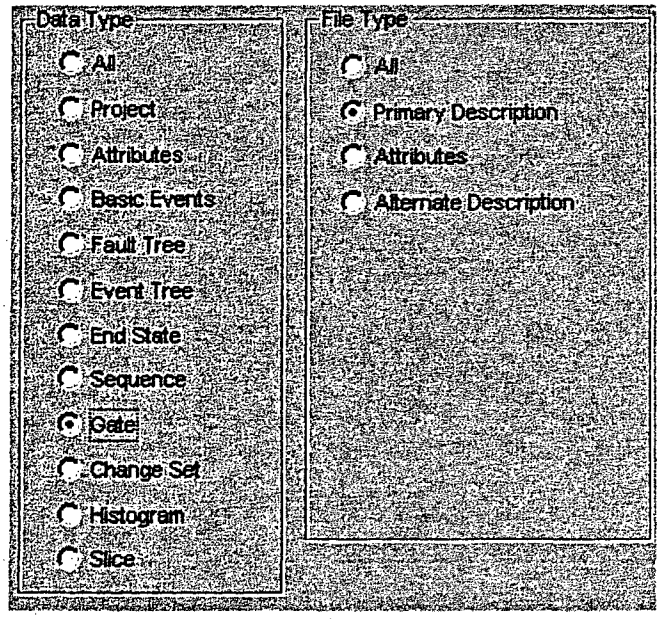
11.3.12.1 Gate

PURPOSE

Gate names and descriptions and types can be loaded for use in graphics conversion or output from the database.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Load and Extract** from the menu. The *Load and Extract Data* dialog will be displayed.
3. Select the desired *Data Action*, **MAR-D** *Data Format*, **Gate** *Data Type*, then the desired *File Type*.



Process - Continue the load or extract operation for the selected Gate File Type.
Cancel - Close the *Load and Extract Data* dialog.

11.3.12.2 Gate Description

File Name:
xxxxxx.GTD

File Format:
project=
name,description[,A]

where

project	24 character	Project name
name	24 character	Gate name
description	60 character	Gate description
A	1 character	If included indicates alternate description

11.3.12.3 Gate Attributes

File Name:
xxxxxx.GTA

File Format
project=
name,attribute

where

project	24 character	Project name
name	24 character	Gate name
attribute	4 character	Gate type

11.3.13 Change Set Information

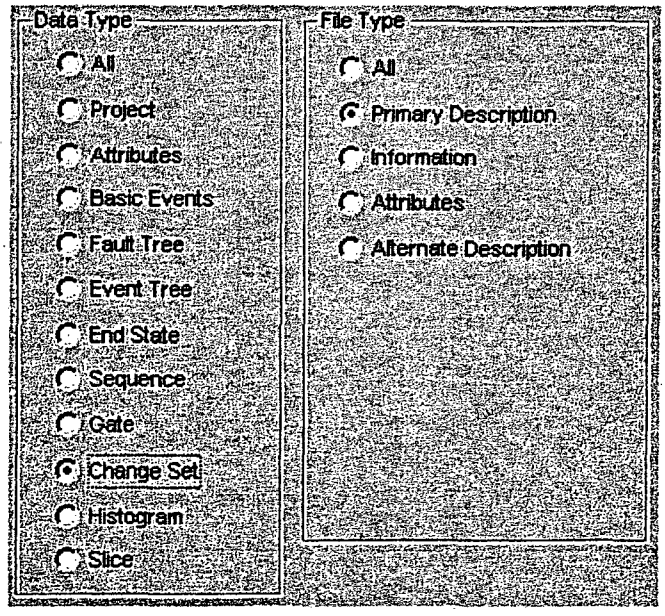
11.3.13.1 Change Sets

PURPOSE

All change set information used in the **Generate** option can be loaded via this file.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Load and Extract** from the menu. The *Load and Extract Data* dialog will be displayed.
3. Select the desired *Data Action*, **MAR-D** *Data Format*, **Change Set** *Data Type*, then the desired *File Type*.



Process - Continue the load or extract operation for the selected Change Set File Type.
Cancel - Close the *Load and Extract Data* dialog.

11.3.13.2 Change Set Description

File Name:

xxxxxx.CSD

File Format:

project=

name,description[,A]

...,...

where

project	24 character	Project name
name	24 character	Change set name
description	60 character	Change set description
A	1 character	If included indicates alternate description

11.3.13.3 Change Set Information

File Name:

xxxxxx.CSI

File Format:

project,change=

^PROBABILITY

eventname,calc,udT,prob,lambda,tau,udV,udC,mission,init

^CLASS

eventname,group,compType,compId,system,location,failMode,train,init,att1,..att16

calcType,udT,prob,lambda,tau,udV,udC,mission,init

^EOS

project,change2=

where

change	24 character	change set name
eventname	24 character	name mask
group	24 characters	event group mask
compType	7 characters	component type mask
compId	3 characters	component ID mask
system	3 characters	system mask
location	3 characters	location mask
failMode	2 characters	failure mode mask
train	2 characters	train mask
init	1 character	initiating event (Y/N)
att1..att16	Class attribute flags	16 values of Y or N (yes or no) indicate whether the attribute described in the class attribute file is applicable.
calc	1 character	Calculation type

1	Probability
2	Lambda * Mission Time
3	1 Exp(-Lambda * Mission Time)
4	Lambda * Min(Mission Time, Tau)
5	Operating component with full repair
6	Lambda * Tau / 2.0
7	1+(EXP(Lambda*Tau) 1.0)/(Lambda*Tau)
8	Base Probability * Probability
9	Base Probability * Probability
T	Set to House Event (Failed, Prob=1.0)

F		Set to House Event (Successful,Prob=0.0)
I		Set to ignore
S		Use fault tree mincut upperbound
G		Seismic event - Enter g level for screening
L		Use low site hazard curve
M		Seismic event - Use medium site hazard curve for screening
H		Use high site hazard curve

udT	1 character	Uncertainty distribution type

L	Log normal, error factor
N	Normal, standard deviation
B	Beta, b of Beta(a,b)
G	Gamma, a Gamma(a)
C	Chi-squared, degrees of freedom
E	Exponential, none
U	Uniform, Upper end pt.
H	Histogram
M	Maximum entropy

prob	Floating point	Probability value
lambda	Floating point	Basic event failure rate per hr.
tau	Floating point	Time to repair in hours
udV	Floating point	Uncertainty distribution value
udC	4 characters	Uncertainty correlation class. Events in same class are 100% correlated.
mission	Floating point	Mission time
init	Boolean (T/F)	Initiating event

11.3.13.4 Change Set Attributes

File Name:

xxxxxx.CSA

File Format:

project=

name,altName

...,...

where

project	24 character	Project name
name	24 character	Change set primary name
altName	24 character	Change set alternate name

11.3.14 Histogram Information

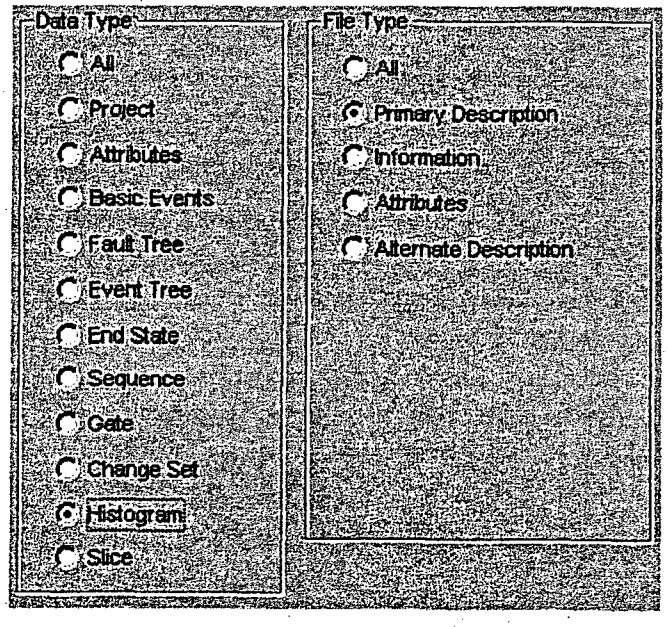
11.3.14.1 Histograms

PURPOSE

Histogram descriptions and information can be loaded into or extracted from the database.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Load and Extract** from the menu. The *Load and Extract Data* dialog will be displayed.
3. Select the desired *Data Action*, **MAR-D** *Data Format*, **Histogram** *Data Type*, then the desired *File Type*.



Process - Continue the load or extract operation for the selected Histogram File Type.
Cancel - Close the *Load and Extract Data* dialog.

11.3.14.2 Histogram Description

File Name:
xxxxxxxx.HID

File Format:
project =

name, type, subtype, description[, A]

where
project 24 character Project name

name	24 character	Histogram primary name	
type	1 character	Histogram type	
H			Hazard
U			Uncertainty
F			Fragility
subtype	1 character	Histogram subtype	
P			Percent
A			Area
R			Range
H			Hazard
Description	60 character	Histogram description	
A	1 character	If included indicates alternate description	

11.3.14.3 Histogram Information

File Name:

xxxxxxxx.HII

File Format:

project, name1=

type, subtype

bin1 value1, bin1 value2

bin2 value1, bin2 value2

...

bin20 value1, bin20 value2

^EOS

project, name2 =

where

Project	24 character	Project name	
NameN	24 character	Histogram primary name	
Type	1 character	Histogram type	
H			Hazard
U			Uncertainty
F			Fragility
Subtype	1 character	Histogram subtype	
P			Percent
A			Area
R			Range
H			Hazard
bin value1	Exponential	first value for bin	
bin value2	Exponential	second value for bin	

11.3.14.4 Histogram Attributes

File Name:

xxxxxxxx.HII

File Format:

project =

name, type, subtype, altName

where

project	24 character	Project name	
name	24 character	Histogram primary name	
type	1 character	Histogram type	
H			Hazard
U			Uncertainty
F			Fragility
subtype	1 character	Histogram subtype	
P			Percent
A			Area
R			Range
H			Hazard
altName	24 character	Histogram alternate name	

11.3.15 Slice Information

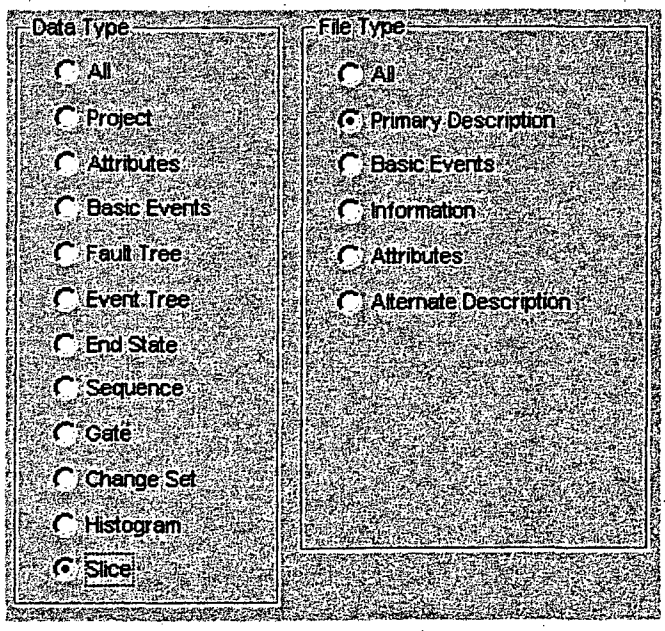
11.3.15.1 Slice

PURPOSE

Slice descriptions and basic event information can be loaded into or extracted from the database.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Load and Extract** from the menu. The *Load and Extract Data* dialog will be displayed.
3. Select the desired *Data Action*, **MAR-D** *Data Format*, **Slice** *Data Type*, then the desired *File Type*.



Process - Continue the load or extract operation for the selected Slice File Type.
Cancel - Close the *Load and Extract Data* dialog.

11.3.15.2 Slice Description

File Name:
xxxxxxxx.SLD

File Format:
project =

name, description[, A]

where

project	24 character	Project name
name	24 character	Slice name
description	60 character	Slice description
A	1 character	If included indicates alternate description

11.3.15.3 Slice Basic Events

File Name:
xxxxxxxx.SLB

File Format:
project, slice =

eventname + eventname + eventname + .

^EOS

project, slice2 =
where

project	24 character	Project name
slice	24 character	Slice name
eventname	24 character	Event names in the slice
+ or *	1 character	Slice logic: +=or , *= and

General Rules:

A plus symbol (+) or asterisk (*) between event names represent the logic in a slice. Spaces are ignored. All logic must be the same in a slice.

A period (.) denotes the end of the slice.

A slash (/) precedes complemented events.

Event names have a maximum of 24 characters including the "/" character for complemented events.

A line beginning with an asterisk (*) is a comment.

11.3.15.4 Slice Information

File Name:

xxxxxxxx.SLI

File Format:

project, slice =

eventname , delta, factor

...,...

^EOS

project, slice2 =

where

project	24 character	Project name
slice	24 character	Slice name
eventname	24 character	Event names in the slice
delta	Floating point	Delta value that is factored
factor	1 character	Factor flag: F=multiply, Blank=add

11.3.15.5 Slice Attributes

File Name:

xxxxxx.SLA

File Format:

project=

name,altName

...,...

where

project	24 character	Project name
name	24 character	Slice primary name
altName	24 character	Slice alternate name

11.3.16 Extracting SETS Data

11.3.16.1 Basic Events Data Format

11.3.16.1.1 SETS Basic Event Descriptions

File Name:

xxxxxxx.DES.

File Format:

name $ description $

name $ description $

where

name	event name
name list	description of event

11.3.16.1.2 SETS Basic Event Failure Rates

File Name:

xxxxxxx.VBK.

File Format:

VALUE BLOCK$ value-block-name
prob $ name-list$
prob $ name-list$

where

prob	point value probability estimate
name list	list of event names separated by commas

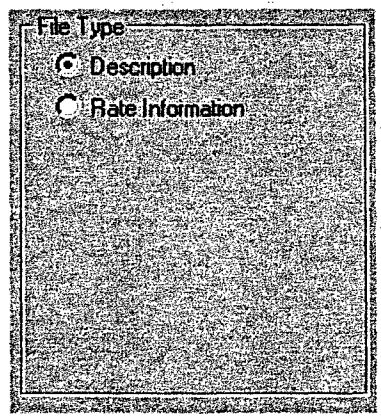
11.3.16.1.3 SETS Basic Events Data Format

PURPOSE

Event descriptions, and failure rates for the Set Equation Transformation System (SETS) data format can be loaded into the database or extracted from it.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Load and Extract** from the menu. The *Load and Extract Data* dialog will be displayed.
3. Select the desired *Data Action*, **SETS** *Data Format*, **Basic Events** *Data Type*, then the desired *File Type*.



Process - Continue the load or extract operation for the selected Basic Events File Type.
Exit - Close the *Load and Extract Data* dialog.

11.3.16.2 Fault Tree Data Format

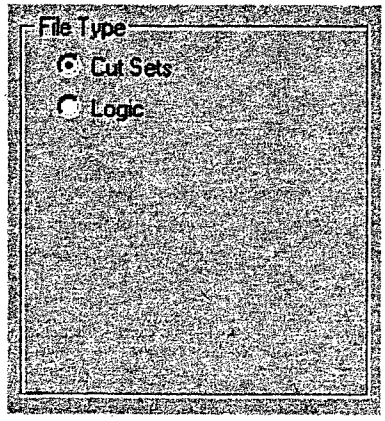
11.3.16.2.1 SETS Fault Tree Data Format

PURPOSE

Fault tree cut sets and logic for the Set Equation Transformation System (SETS) data format can be loaded into the database or output to files.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Load and Extract** from the menu. The *Load and Extract Data* dialog will be displayed.
3. Select the desired *Data Action*, **SETS** *Data Format*, **Fault Tree** *Data Type*, then the desired *File Type*.



Process - Continue the load or extract operation for the selected Fault Tree File Type.
Cancel - Close the *Load and Extract Data* dialog.

11.3.16.2.2 SETS Fault Tree Cut Sets

The fault tree cut sets are stored in the System relation in the block data file. The format of the cut set file (.DNF) is given in the SETS Sequence Cut Sets.

11.3.16.2.3 SETS Fault Tree Logic

File Name:

xxxxxx.SET.

File Format:

FAULT TREE$ fault tree name.

COMMENT$ descriptive material $

gate type $ gate name. IN$ input 1, input 2, . . . , input n.

OUT$ output 1, output 2, . . . , output n.

event type $event name. OUT$ output 1, . . . , output n.

where

fault tree name	The name of the fault tree.
gate type	The type of gate being defined.
AG	= AND gate
OG	= OR gate
EOR	= Exclusive OR gate (converted to SG)
EAG	= Exclusive AND gate (converted to SG)
SG	= Special Gate
gate name	The name of the gate being defined (16 characters) input n
	The names of the gates or primary events that are the immediate inputs to the gate being defined (16 characters)
output n	The names of the gates that are the immediate outputs of the gate or primary event being defined (16 characters).
event type	The type of primary event being defined.
BE	= Basic Event
CE	= Conditional Event
UE	= Undeveloped Event
DE	= Developed Event
EE	= External Event
COMMENT$	Defines a comment. Must follow a "." delimiter.

11.3.16.3 Sequence Data Format

11.3.16.3.1 SETS Sequence Cut Sets

File Name:

xxxxxx.DNF.

The format of the SETS output cut sets file (.DNF) is dependent upon the command issued within SETS. The factored form is

A * (B + C)

The disjunctive normal form is

A * B + A * C.

ONLY the disjunctive normal form is accepted by the MAR D at this time.

File Format:

sequence-name =

eventName * eventName +

eventName * eventName.

where

. . . . .

General Rules:

An asterisk (*) separates event names. Spaces are ignored.

A plus sign (+) separates cut sets.

A period (.) denotes the end of a sequence.

An asterisk (*) in the first column denotes a comment.

11.3.16.3.2 SETS Sequence Data Format

PURPOSE

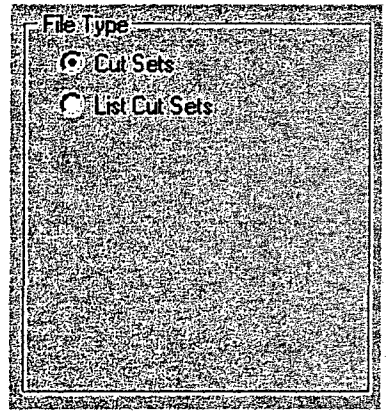
Sequence cut sets, and list cut sets for the Set Equation Transformation System (SETS) data format can be loaded or output.

The event tree initiating event will be included in each cut set term for SETS *.DNF format output.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Load and Extract** from the menu. The *Load and Extract Data* dialog will be displayed.

3. Select the desired *Data Action*, **SETS** *Data Format*, **Sequence** *Data Type*, then the desired *File Type*.



Process - Continue the load or extract operation for the selected Sequence File Type.
Cancel - Close the *Load and Extract Data* dialog.

11.4 Recovering the Data Base

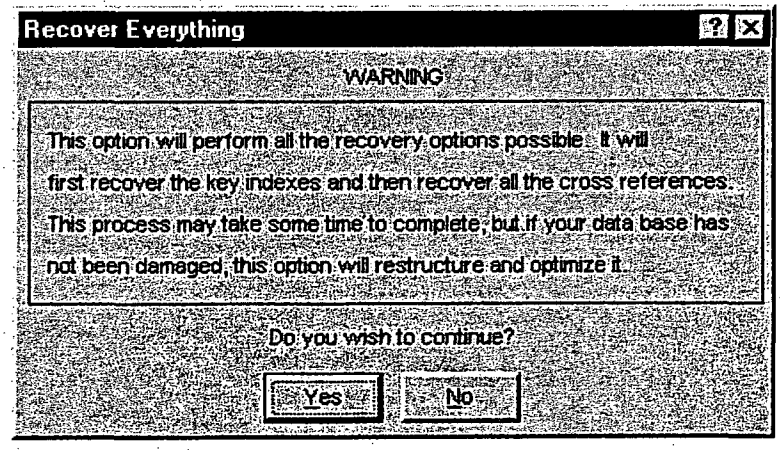
PURPOSE

This option allows you to restructure the database and re index the data.
NOTE

Before recovering the database, it is suggested that you backup the *.DFL files and corresponding data files (*.IDX , *.DAT, and *.BLK) in the directory of the project to be rebuilt.

STEPS

1. From the menu select **Utility**.
2. Choose **Recover Data Base** from the menu. The *Recover Everything* warning dialog will be displayed.



Yes - Continue with the recovery. Status messages will appear in the message bar of the SAPHIRE window as the recover process proceeds. "Successful Completion," will be the last message displayed in the message bar.

No - Close the *Recover Everything* dialog without recovering the database.

Some indications that a database rebuild is necessary include:

- Data elements such as events/fault trees have been deleted and seem to reappear
- During cut set generation or update, the min cut upper bound seems surprisingly high
- Cross Reference reports show/don't show events being used properly
- Events/fault trees that don't appear to be referenced cannot be deleted, and
- After a software version update
- SAPHIRE displays a message stating a database recovery is required.

You may rebuild the database anytime because the rebuild process compacts the data and generally helps the software run faster. This option will recover all key indexes and then recover the cross-references. This process may take several minutes to complete. If your database has not been damaged, this option will just restructure and optimize your database.

11.5 Update and Align

11.5.1 Align Primary and Alternate Descriptions

PURPOSE

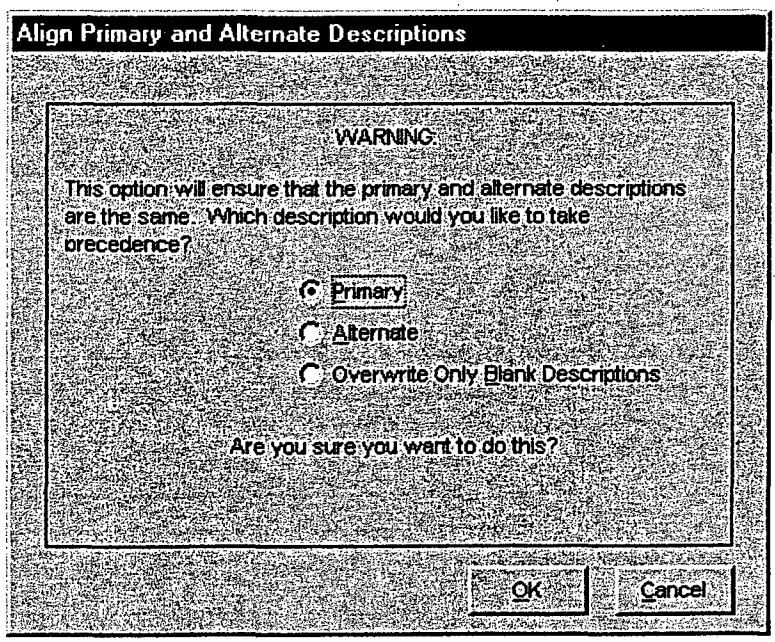
This option allows you to synchronize the primary and alternate descriptions for all data items (fault trees, gates, events, etc) in the current project. When an alternate naming scheme is not desired, this utility can be used to ensure that all primary and alternate descriptions match.

SAPHIRE supports a dual naming scheme. Each record in a project has two names and two descriptions, referred to as primary and alternate. The purpose of the alternate name and description is to allow an alternate view of the project. The alternate name and description could be expressed in another language (see International SAPHIRE), or in layman's terms, or whatever the project developer wishes. The primary name and descriptions are displayed by default through SAPHIRE, in lists, graphics, and reports. The default display can be changed to show alternate names and descriptions (see Defining Constants).

Not all projects need to use this feature, particularly in the model development phase of the project. However SAPHIRE still requires that the alternate names be specified. As new items are added to the project, SAPHIRE defaults the alternate name and description to match the primary name and description, unless otherwise specified. When these items are later modified by the user, SAPHIRE does not automatically update the alternate name and description information; it assumes the name and description are different by design. This can cause confusion, since the item can now be referred to in the project by either name. When an alternate naming scheme is not desired, this utility can be used to ensure that all primary and alternate descriptions match.

STEPS

1. From the menu select **Utility**.
2. Choose **Update Descriptions** from the menu. The *Align Primary and Alternate Descriptions* warning dialog will be displayed.
3. Select *Primary, Alternate, or* as the name field to retain.
4. Choose **OK** to perform the alignment.



Primary - Overwrite the alternate description with the primary description.

Alternate - Overwrite the primary description with the alternate description.

Overwrite Only Blank Descriptions - Overwrite the alternate description with the primary description only when the alternate description is blank.

OK - Align all project item descriptions to match the selected option.

Cancel - Close the dialog without performing the alignment.

11.5.2 Align Primary and Alternate Names

PURPOSE

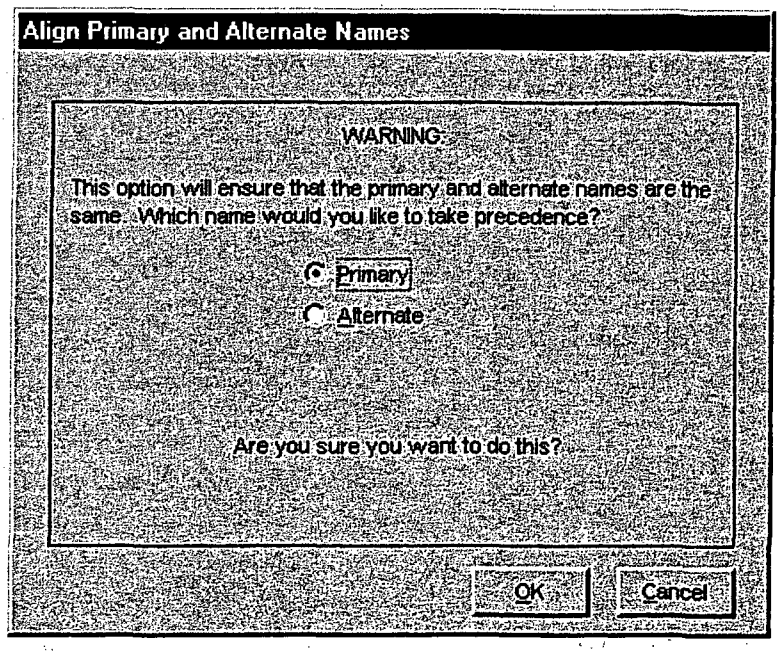
This option allows you to synchronize the primary and alternate names for all data items (fault trees, gates, events, etc) in the current project. When an alternate naming scheme is not desired, this utility can be used to ensure that all primary and alternate names match each other.

SAPHIRE supports a dual naming scheme. Each record in a project has two names and two descriptions, referred to as primary and alternate. The purpose of the alternate name and description is to allow an alternate view of the project. The alternate name and description could be expressed in another language (see International SAPHIRE), or in layman's terms, or whatever the project developer wishes. The primary name and descriptions are displayed by default through SAPHIRE, in lists, graphics, and reports. The default display can be changed to show alternate names and descriptions (see Defining Constants).

Not all projects need to use this feature, particularly in the model development phase of the project. However SAPHIRE still requires that the alternate names be specified. As new items are added to the project, SAPHIRE defaults the alternate name and description to match the primary name and description, unless otherwise specified. When these items are later modified by the user, SAPHIRE does not automatically update the alternate name and description information; it assumes the name and description are different by design. This can cause confusion, since the item can now be referred to in the project by either name. When an alternate naming scheme is not desired, this utility can be used to ensure that all primary and alternate names match.

STEPS

1. From the menu select **Utility**.
2. Choose **Align Names** from the menu. The *Align Primary and Alternate Names* warning dialog will be displayed.
3. Select *Primary* or *Alternate* as the name field to retain.
4. Choose **OK** to perform the alignment.



Primary - Overwrite the alternate name with the primary name.
Alternate - Overwrite the primary name with the alternate name.

OK - Align all project item names to match the selected option.
Cancel - Close the dialog without performing the alignment.

SEE ALSO

Align Primary and Alternate Descriptions

11.6 Fault Tree Utility Options

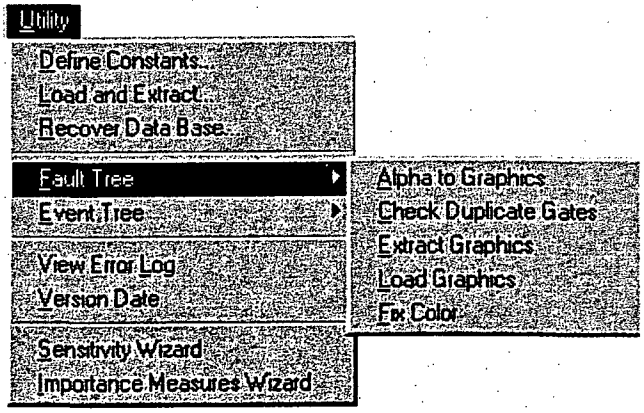
11.6.1 Fault Tree Utility Options

PURPOSE

This option provides you with the ability to manage fault tree graphics files. You can convert fault tree logic from alphanumeric format to graphical format, and load or extract fault tree graphics.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Fault Tree** from the menu. Select a sub-menu option.



Alpha to Graphics -	Convert fault tree or sub-tree alphanumeric logic to a graphical format.
Check Duplicate Gates –	Finds and reports duplicate gates.
Extract Graphics -	Extract fault trees from the database.
Load Graphics -	Load fault tree graphic files into the database.
Fix Color -	Quickly map unique fault tree diagram display attributes (including color and font) to individual or groups of gates and events. The selected attributes will be automatically applied to all fault tree diagrams in the project.

11.6.2 Check Duplicate Gates

PURPOSE

This option searches the logic of all project fault trees for duplicate gates. A duplicate gate is a gate that appears in more than one fault tree. Although it is acceptable practice to have a gate in more than one fault tree, the gate with the same name in different fault trees should be of the same type (e.g., AND, OR, N/M, etc.) and should have the same inputs.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Fault Tree** from the menu.
3. Select the **Check Duplicate Gates** sub-menu option.

The software will search for duplicate gates. If no duplicate gates are found, a message on the status bar such as, "No Duplicate Gates were found" will be displayed. If duplicate gates were found the message, "Duplicate Gates were found. Check error.log file."

Sample error.log file:

```
Duplicate Gate Error->LPI-RCS-LP1, Fault Tree->LPR
Duplicate Gate Error->LPI-RCS-LP1, Fault Tree->RHR
Duplicate Gate Error->LPI-RCS-LP1, Fault Tree->LPI
*DUPLICATE GATE ERROR->LPI-RCS-LP2, Fault Tree->LPR
*DUPLICATE GATE ERROR->LPI-RCS-LP2, Fault Tree->RHR
*DUPLICATE GATE ERROR->LPI-RCS-LP2, Fault Tree->LPI
```

The example above shows part of the error log file that is generated after checking for duplicate gates. The first three lines indicate that the gate, LPI-RCS-LP1 is duplicated in three fault trees: LPR, RHR, and LPI. The next three lines indicate that the gate, LPI-RCS-LP2 is also duplicated in three fault trees, but because those lines begin with an asterisk, there is an error in the fault tree logic. There could be an error because the gate, LPI-RCS-LP2, is defined as a different type in one of the trees or because it has different inputs in one of the trees. Duplicate gates with an asterisk indicate conflicts that will need to be resolved before sequence cut set generation is performed.

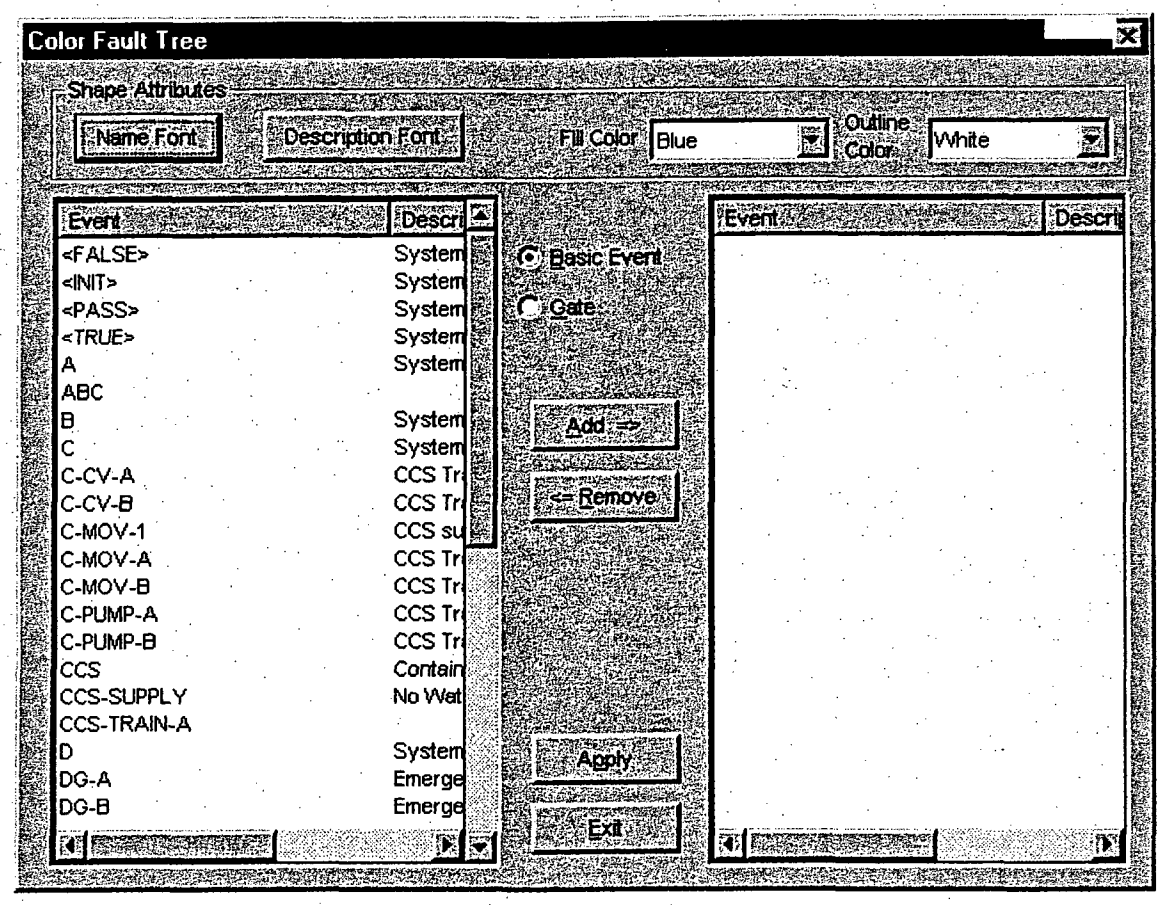
11.6.3 Color Fault Trees

PURPOSE

This option allows you to quickly assign distinctive display attributes to one or more gates or events across all fault tree diagrams. You can change the name and description fonts, and the shape and outline colors.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Fault Tree** from the menu. Select the **Fix Colors** sub-menu option.
3. The *Color Fault Tree* dialog will be displayed listing either all of the events or all of the gates in the current project.



Set the desired shape display attributes. Use the **Add** and **Remove** buttons to create a list of selected gates or events from the current project. When the list and the attributes are satisfactory, press the **Apply** button to update the fault tree graphics.

Shape Attributes

Name Font - Select a font name, size, and color to apply to the selected event(s)/gate(s) names.

Description Font - Select a font name, size, and color to apply to the selected event(s)/gate(s) descriptions.

Fill Color - Select a color to apply to the selected event(s)/gate(s) shapes.

Outline Color - Select a color to apply to the selected event(s)/gate(s) shape outline.

List Type

Basic Event - When this option is selected, all of the basic events in the current project will be listed on the left.

Gate - When this option is selected, all of the gates in the current project will be listed on the left.

Add - Add the selected event(s)/gate(s) from the list on the left, and add them t o the list on the right.

Remove - Remove the selected event(s)/gate(s) from the list on the right.

Apply - For each fault tree in the database, apply the currently selected shape attributes to each occurrence of the events/gates listed in the list on the right. If any fault trees have logic but no graphics, you will be prompted to optionally create the graphics.

Exit - Close the *Color Fault Tree* dialog.

Note: Choose your colors with care. For example, a white description font color will not show up on a white shape color, nor will blue name color show up on a blue background color.

11.6.4 Fault Tree Page Numbering

11.6.4.1 Fault Tree Page Numbering

PURPOSE

This option allows you to select a page numbering scheme to use for all fault trees in the current project. Three options are currently available: **By Hierarchy, By Name,** and **By Level**. The starting page number can also be set.

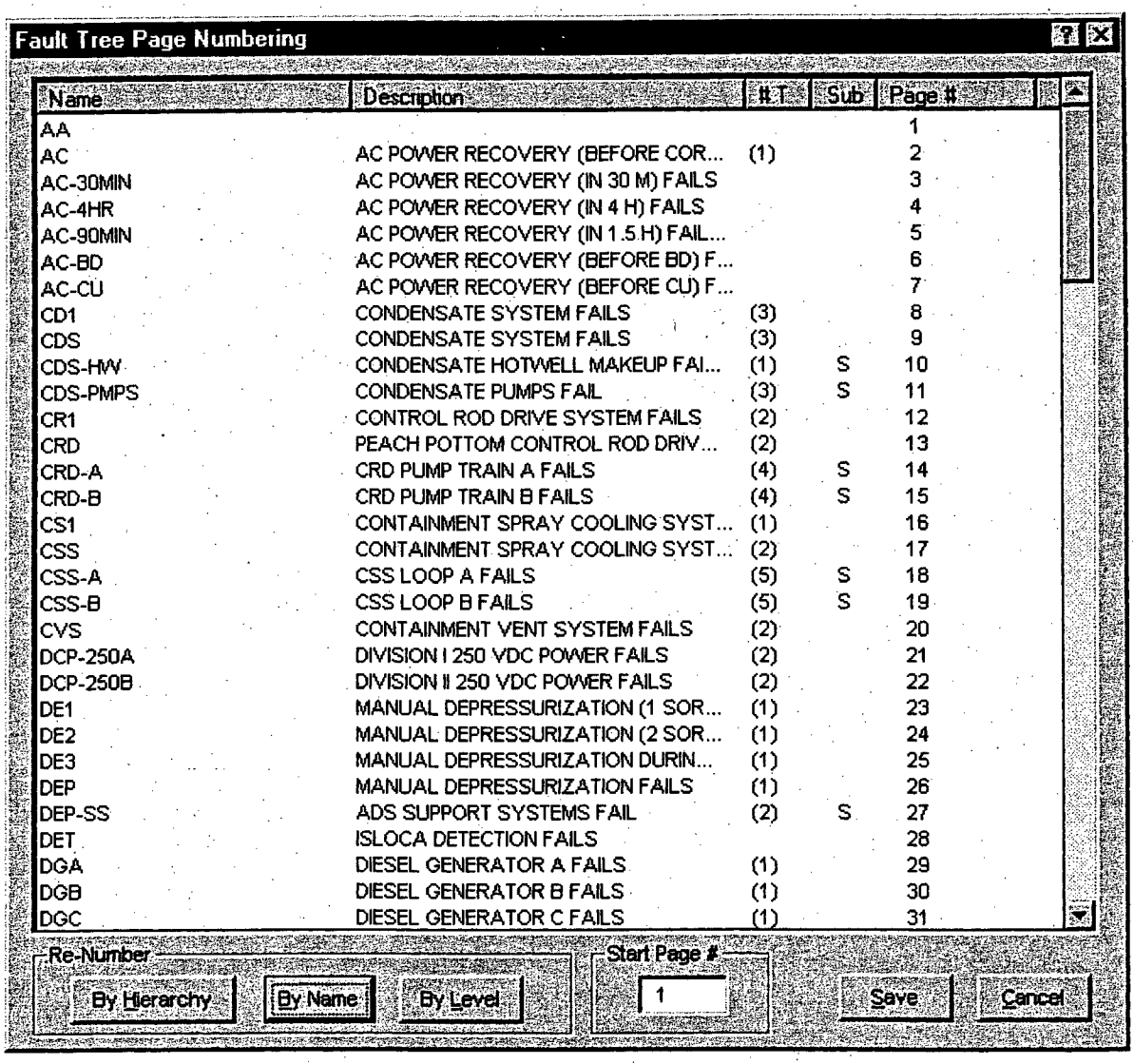
The **By Name** option orders fault trees strictly alphabetically. The **By Level** option orders top level fault trees first, followed by sub-trees. Within that order, fault trees are numbered alphabetically. The **By Hierarchy** option groups fault trees with their sub-trees. Top level trees appear in relative alphabetic order.

The dialog lists the name, description, number of transfers contained in the fault tree, top/sub tree status, and proposed page number assignment. Click any of the columns (except description) to view the list in order of that column. Click the column again to reverse the sort order.

Note: When a new fault tree is added to the project, an unused page number will be assigned to it. The page numbering utility must be run before the new fault tree's page number will fit into the selected page numbering scheme. Likewise, when a fault tree is deleted, there will be a hole in the page numbering scheme until the fault trees are renumbered using this utility.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Fault Tree** from the menu. Select the **Number Pages** sub-menu option.
3. The *Fault Tree Page Numbering* dialog will be displayed listing all of the fault trees in the current project.
4. Enter the desired *Start Page* number.
5. Choose the desired *Re-Number* button. The page numbers will be updated.
6. (Optional) Click on the Page Number column heading to view the list in the updated page number order.
7. Choose **Save** to update the page numbers, or Cancel to revert to the original numbers.



Click on the column headings to view the list in that column's order.

Re-Number

By Hierarchy - When this option is selected, fault trees and their referenced sub-trees will be grouped together. Top level fault trees appear in relative alphabetical order.

By Name - When this option is selected, fault trees will be numbered in alphabetical order.

By Level - When this option is selected, all top level trees will be numbered first, followed by all sub-trees. Within each level, the fault trees will be ordered alphabetically.

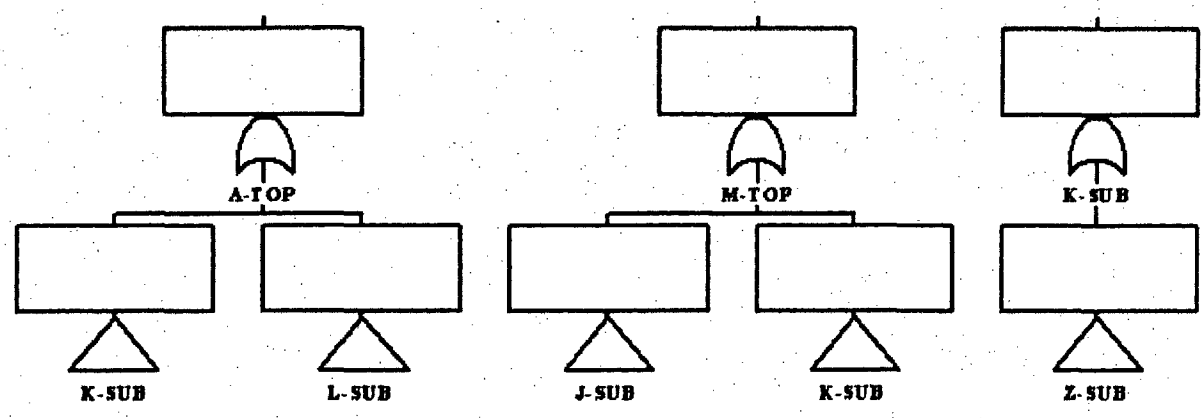
Start Page # - Enter the desired page number of the first fault tree.

Save - Permanently assign the fault tree page numbers as currently shown in the fault tree list, and close the *Page Numbering* dialog.

Cancel - Close the *Page Numbering* dialog without changing the previously assigned page numbers.

11.6.4.2 Fault Tree Page Numbering Example

Given a project containing the following fault trees,



the table below shows the page numbers that would result from each ordering scheme.

Page Number	Order By Hierarchy	Order By Name	Order By Level
1	A-TOP	A-TOP	A-TOP
2	K-SUB	J-SUB	M-TOP
3	Z-SUB	K-SUB	J-SUB
4	L-SUB	L-SUB	K-TOP
5	M-TOP	M-TOP	L-SUB
6	J-TOP	Z-SUB	Z-SUB

11.6.5 Alpha to Graphics Conversion

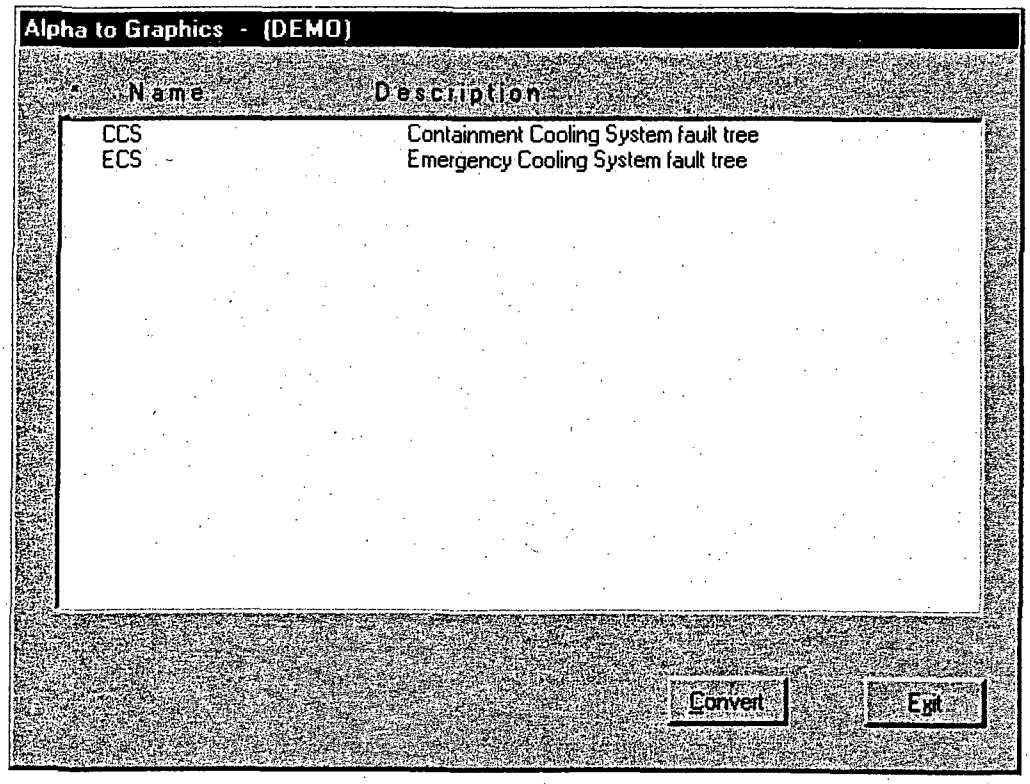
11.6.5.1 Alpha to Graphics

PURPOSE

This option allows you to convert the alphanumeric logic for a fault tree or sub-tree to a graphical format. Use this option when you define the fault tree logic with an ASCII file or use the logic editor to change the logic for a fault tree. The graphics and logic information for a fault tree are two separate entities, therefore, a conversion must be made if any changes are made to the alphanumeric logic. However, if changes are made the to the graphical representation, the changes are automatically reflected in the alphanumeric logic.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Fault Tree** from the menu. Select the **Alpha to Graphics** sub-menu option.
3. The *Alpha to Graphics* dialog will be displayed listing all of the fault trees in the current project.



Convert - Continue with the conversion process.
Exit - Close the *Alpha to Graphics* dialog without performing conversions.

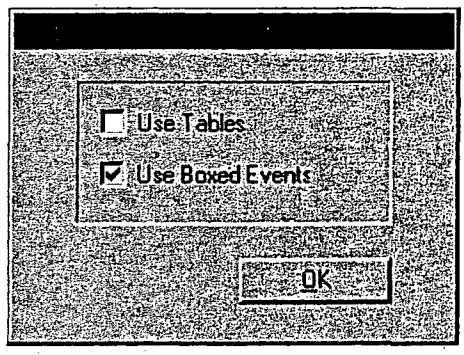
11.6.5.2 Convert

PURPOSE

Convert the selected alphanumeric format file(s) to a graphical format.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Fault Tree** from the menu.
3. Select the **Alpha to Graphics** sub-menu option. The *Alpha to Graphics* dialog will be displayed.
4. Highlighted the desired fault tree(s) and choose the **Convert** button.
5. A dialog, as shown below, will be displayed. Make the selections as desired and choose the **OK** button.



If neither check box is selected, normal basic events are used in the model.

Use Tables - For each logic gate with many basic events as inputs, create a table of basic events, using the Primary name field. SAPHIRE processes the table of basic events as if they were separate symbols.

Use Boxed Events - Create each basic events with a box for descriptive text. The text from the description field of the associated event will be placed in the box. This does not influence the logic of the fault tree, but adds clarity to the model for those using and reviewing it.

OK - Complete the conversion process.

11.6.6 Extract Fault Tree Graphics

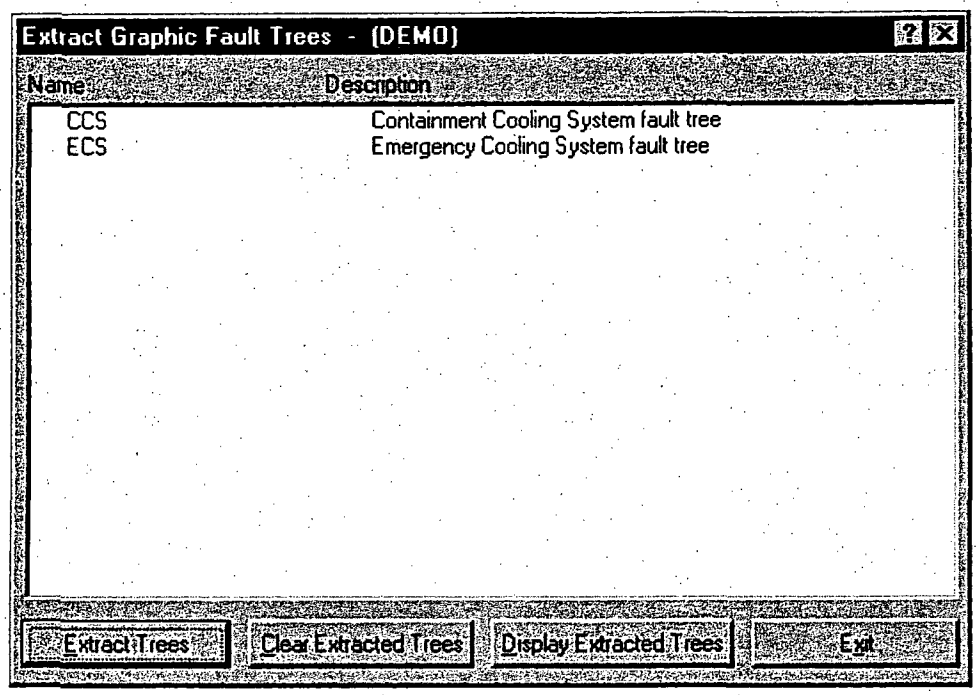
11.6.6.1 Extract Fault Tree Graphics

PURPOSE

This option allows you to extract fault trees from the database. When you create a fault tree diagram and save it, the .DLS file is saved in the database as well as in a temporary .DLS file in the current project directory. When you delete the .DLS file, the file still exists in the database. With this option, you can extract the stored .DLS file from the database.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Fault Tree** from the menu.
3. Select the **Extract Graphics** sub-menu option. The *Extract Graphic Fault Trees* dialog will be displayed listing all of the fault trees in the current project.



Extract Trees - Copy the specified trees out of the database.
Clear Extracted Trees - Delete all extracted trees from the project directory.
Display Extracted Trees - Display all available extracted trees.
Exit - Close the *Extract Graphic Fault Trees* dialog.

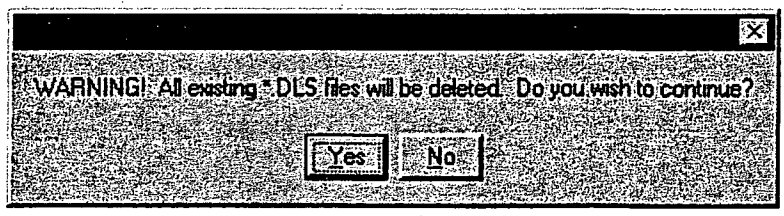
11.6.6.2 Extract Trees

This option allows you to copy the highlighted trees from the database to a file in the directory of your choice. When complete, a message indicating that the tree was successfully extracted will be displayed in the message bar of the SAPHIRE window. The .DLS or .ETG file name of the extracted tree will be included in the message.

The extraction process does not delete the tree from the database.

11.6.6.3 Clear Extracted Trees

This option allows you to clear all extracted tree files from the current project directory. When you choose this button, a warning dialog, as shown below, will be displayed.



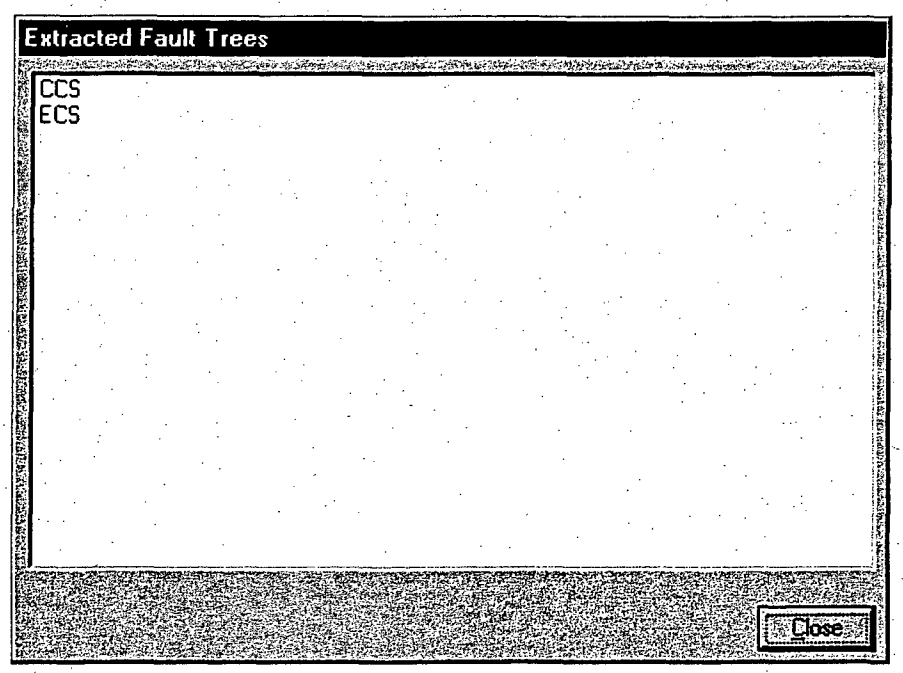
Yes - Delete all existing extracted tree files (.DLS or .ETG) from the current project directory. When complete the message, "All extracted files cleared..." will be displayed in the message bar of the SAPHIRE window.

No - Terminate the delete process.

11.6.6.4 Display Extracted Trees

This option lists all the extracted trees, by file name, residing in the current project directory. When you choose this button, the *Extracted Fault Trees* or *Extracted Event Trees* dialog will be displayed.

The example here shows the dialog for fault trees.



Close - Close the *Extracted Fault Trees* or *Extracted Event Trees* dialog.

11.6.7 Load Fault Tree Graphics

11.6.7.1 Load Fault Tree Graphics

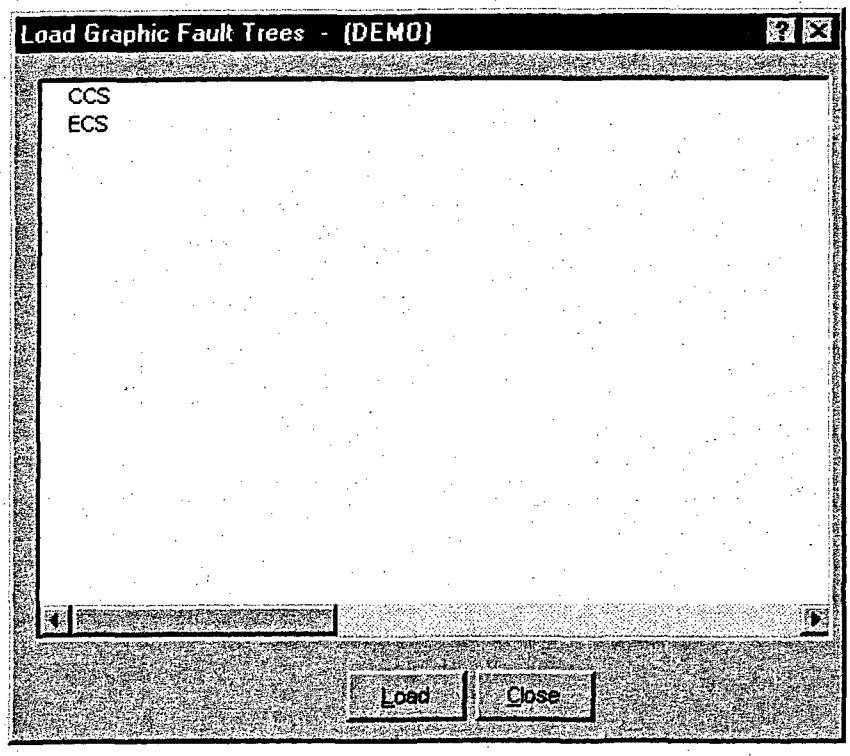
PURPOSE

This option allows you to load graphic files into the database. When you perform this operation, the selected file will be converted and loaded into the existing fault tree record whose name matches that of the file. If no record exists, SAPHIRE will create a new fault tree record using the file name for the name of the fault tree in the database.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Fault Tree** from the menu.

3. Select the **Load Graphics** sub-menu option. The *Load Graphic Fault Trees* dialog will be displayed listing all of the .DLS files in the current project directory.



Load - Perform the load operation.
Close - Close the *Load Graphic Fault Trees* dialog.

11.7 Event Tree Utility Options

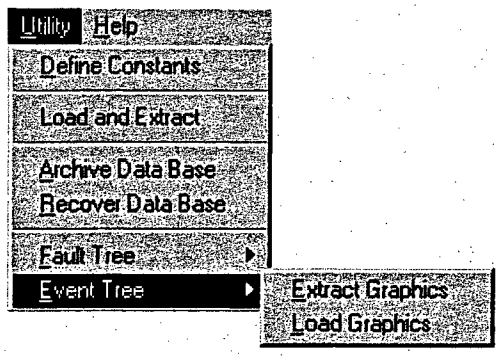
11.7.1 Event Tree Utility Options

PURPOSE

This option allows you to load or extract event tree graphics files.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Event Tree** from the menu. Select a sub-menu option.



Extract Graphics - Extract event trees from the database.
Load Graphics - Load event tree graphic files into the database.

11.7.2 Event Tree Page Numbering

PURPOSE

This option allows you to select a page numbering scheme to use for all event trees in the current project. Three options are currently available: **By Hierarchy, By Name,** and **By Level**. The starting page number can also be set.

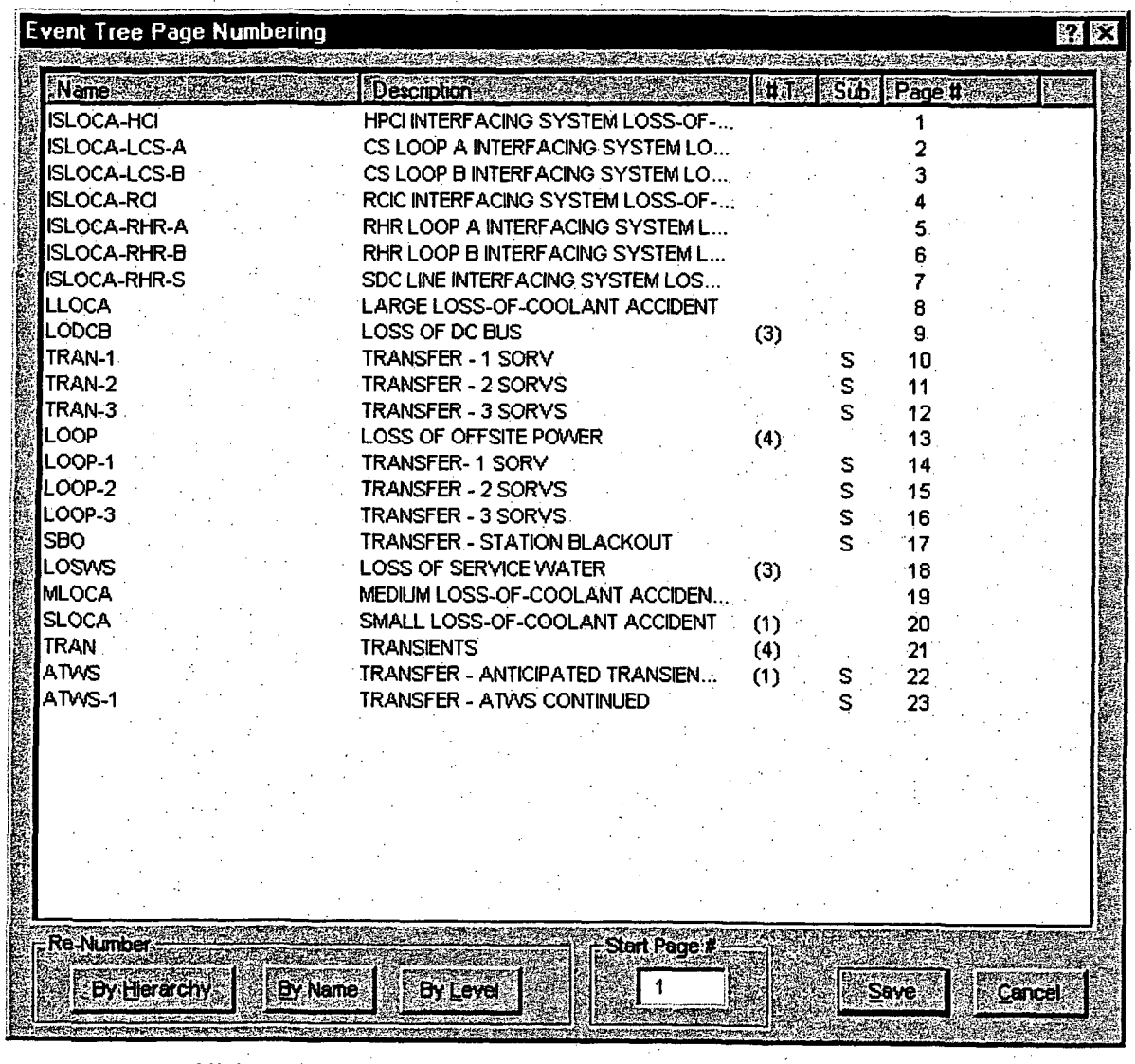
The **By Name** option orders event trees strictly alphabetically. The **By Level** option orders top level event trees first, followed by sub-trees (or transfer trees). Within that order, event trees are numbered alphabetically. The **By Hierarchy** option groups event trees with their sub-trees. Top level trees appear in relative alphabetic order.

The dialog lists the name, description, number of transfers contained in the event tree, top/sub tree status, and proposed page number assignment. Click any of the columns (except description) to view the list in order of that column. Click the column again to reverse the sort order.

Note: When a new event tree is added to the project, an unused page number will be assigned to it. The page numbering utility must be run before the new event tree's page number will fit into the selected page numbering scheme. Likewise, when an event tree is deleted, there will be a hole in the page numbering scheme until the event trees are renumbered using this utility.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Event Tree** from the menu. Select the **Number Pages** sub-menu option.
3. The *Event Tree Page Numbering* dialog will be displayed listing all of the fault trees in the current project.
4. Enter the desired *Start Page* number.
5. Choose the desired *Re-Number* button. The page numbers will be updated.
6. (Optional) Click on the Page Number column heading to view the list in the updated page number order.
7. Choose **Save** to update the page numbers, or Cancel to revert to the original numbers.



Click on the column headings to view the list in that column's order.

Re-Number

By Hierarchy - When this option is selected, event trees and their referenced transfer trees will be grouped together. Top level event trees appear in relative alphabetical order.

By Name - When this option is selected, event trees will be numbered in alphabetical order.

By Level - When this option is selected, all top level trees will be numbered first, followed by all transfer trees. Within each level, the event trees will be ordered alphabetically.

Start Page # - Enter the desired page number of the first event tree.

Save - Permanently assign the event tree page numbers as currently shown in the event tree list, and close the *Page Numbering* dialog.

Cancel - Close the *Page Numbering* dialog without changing the previously assigned page numbers.

11.7.3 Extract Event Tree Graphics

11.7.3.1 Extract Event Tree Graphics

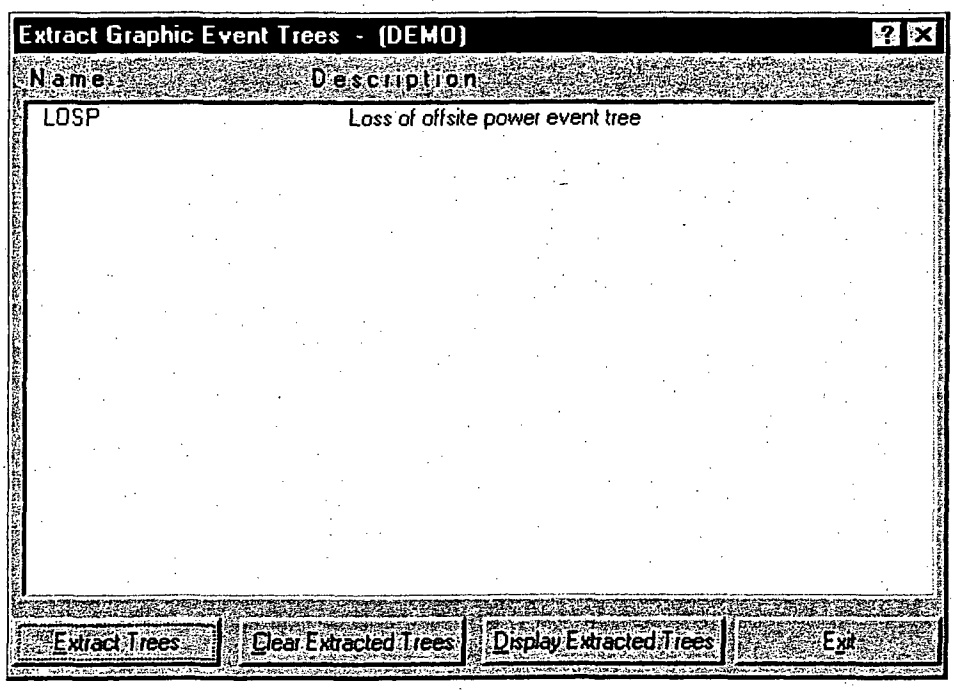
PURPOSE

This option allows you to extract event trees from the database. When you create an event tree diagram and save it, the .ETG file is saved in the database as well as in a temporary .ETG file in the current project directory. When you delete the .ETG file, the file still exists in the database. With this option, you can extract the stored .ETG file from the database.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Event Tree** from the menu.

Select the **Extract Graphics** sub-menu option. The *Extract Graphic Event Trees* dialog will be displayed listing all of the event trees in the current project.



Extract Trees - Remove the specified trees from the database.
Clear Extracted Trees - Delete all extracted trees from the project directory.
Display Extracted Trees-Display all extracted trees to date.
Exit - Close the *Extract Graphic Event Trees* dialog.

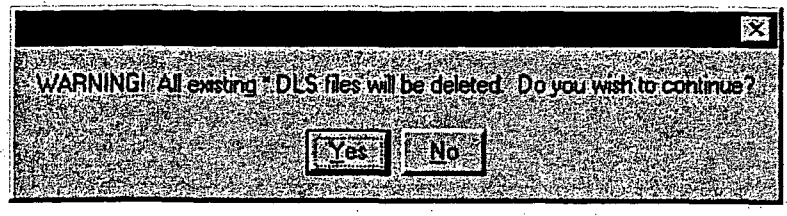
11.7.3.2 Extract Trees

This option allows you to copy the highlighted trees from the database to a file in the directory of your choice. When complete, a message indicating that the tree was successfully extracted will be displayed in the message bar of the SAPHIRE window. The .DLS or .ETG file name of the extracted tree will be included in the message.

The extraction process does not delete the tree from the database.

11.7.3.3 Clear Extracted Trees

This option allows you to clear all extracted tree files from the current project directory. When you choose this button, a warning dialog, as shown below, will be displayed.



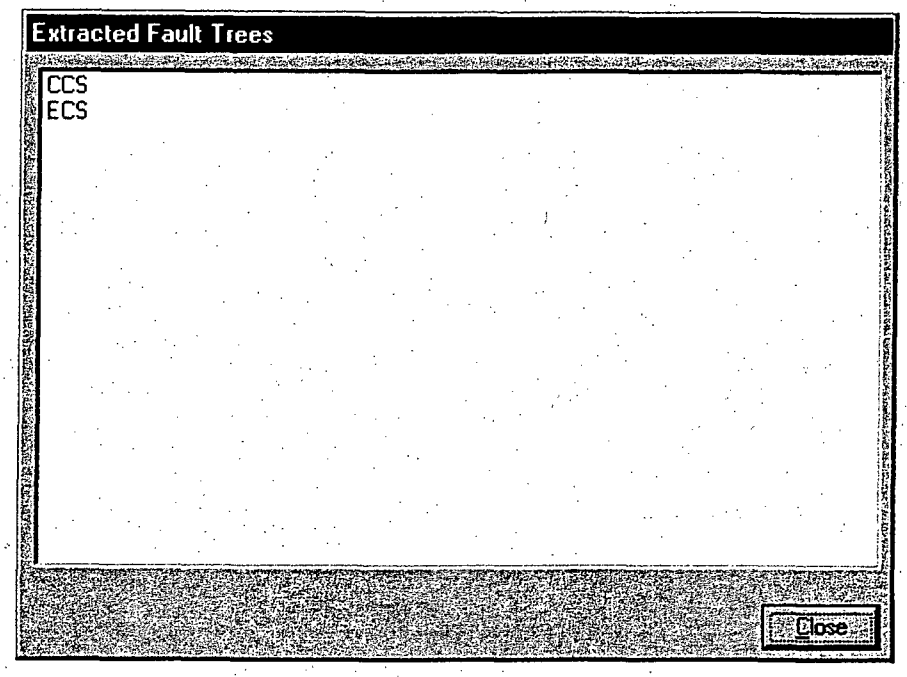
Yes - Delete all existing extracted tree files (.DLS or .ETG) from the current project directory. When complete the message, "All extracted files cleared..." will be displayed in the message bar of the SAPHIRE window.

No - Terminate the delete process.

11.7.3.4 Display Extracted Trees

This option lists all the extracted trees, by file name, residing in the current project directory. When you choose this button, the *Extracted Fault Trees* or *Extracted Event Trees* dialog will be displayed.

The example here shows the dialog for fault trees.



Close - Close the *Extracted Fault Trees* or *Extracted Event Trees* dialog.

11.7.4 Load Event Tree Graphics

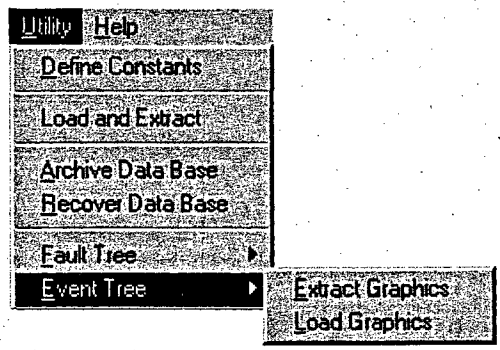
11.7.4.1 Load Event Tree Graphics

PURPOSE

This option allows you to load event tree graphic files into the database. When you perform this operation, the selected file will be converted and loaded into the existing event tree record whose name matches that of the file. If no record exists, SAPHIRE will create a new event tree record using the file name for the name of the event tree in the database.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Event Tree** from the menu.
3. Select the **Load Graphics** sub-menu option. The *Load Graphic Event Trees* dialog will be displayed listing all of the .ETG files in the current project directory.



Load - Perform the load operation.
Close - Close the *Load Graphic Event Trees* dialog.

11.8 View Error Log

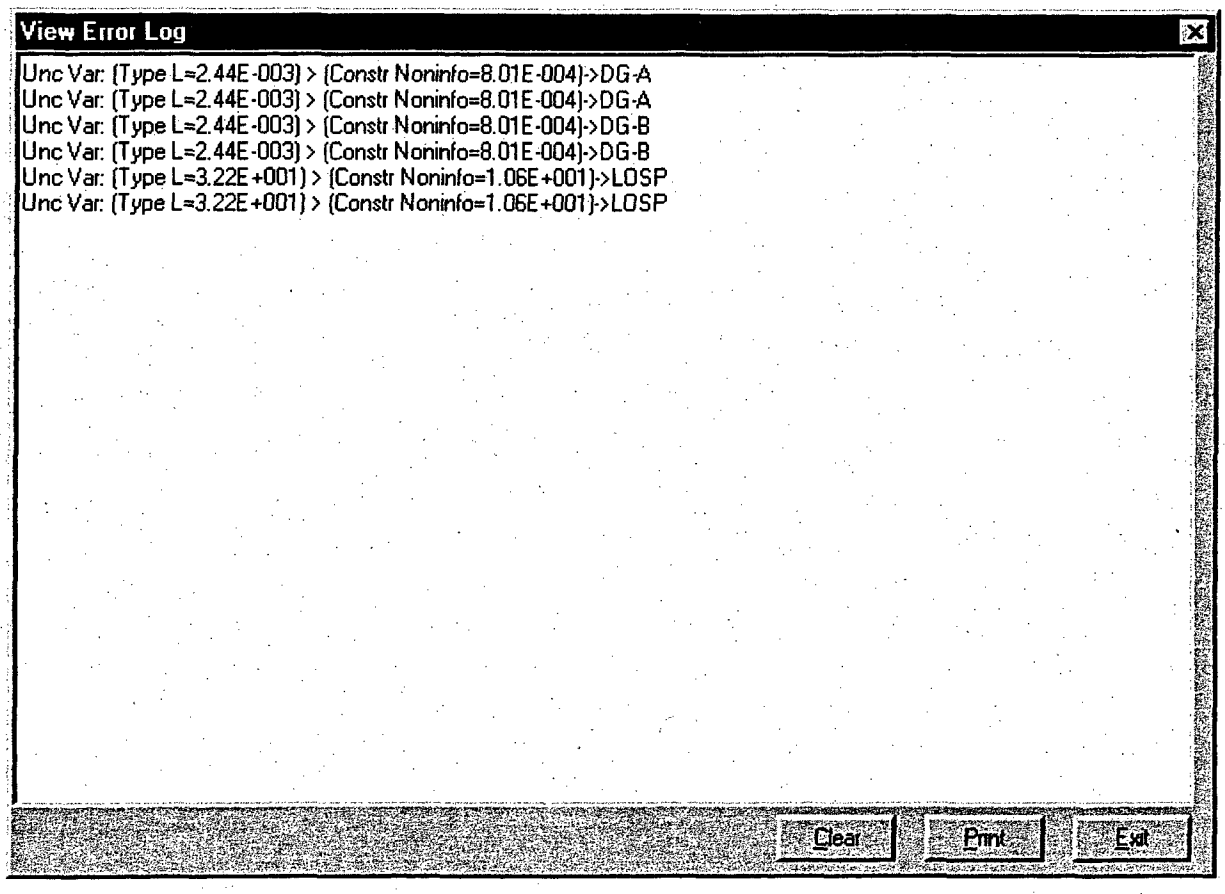
11.8.1 View Error Log

PURPOSE

This option allows you to view the results of the SAPHIRE error log. An error log is produced each time basic event data is generated. The log reports errors, warnings, and inconsistencies detected in the current event data. The error log can remain open while other SAPHIRE dialogs are accessed, allowing you to see the error list while making any desired corrections. The error log is automatically cleared upon exiting SAPHIRE.

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **View Error Log** from the menu. The *View Error Log* dialog will be displayed.



Print – Print the error log.
Clear – Clear the error log.
Exit – Close the *View Error Log* dialog.

11.9 Version Date

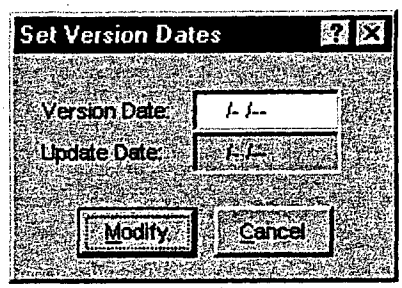
11.9.1 Version Date

PURPOSE

This option allows you to set the version date of the current project (i.e., model).

STEPS

1. From the SAPHIRE menu select **Utility**.
2. Choose **Version Date** from the menu. The *Set Version Dates* dialog will be displayed.

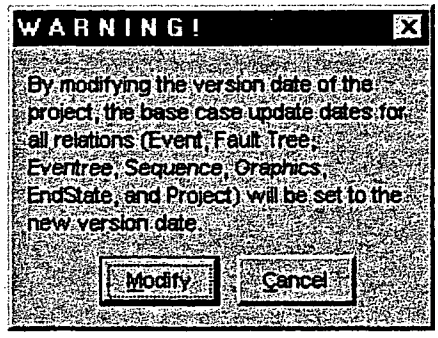


Version Date – Enter the new version date in YYYY/MM/DD format. The version date indicates the date the model development was complete.

Update Date – Displays the last date that the model was updated in YYYY/MM/DD format. The update date is changed by SAPHIRE whenever any of the following are changed: **fault tree logic, fault tree graphics, event tree logic**, or **event tree graphics**. The update date will also be changed if a **basecase update** is performed.

Modify – Close the *Set Version Dates* dialog and change the version date for the project. When you choose this button, the *Warning* dialog, below, will be displayed.

Cancel – Close the *Set Version Dates* without changing the project's version date.



Modify – Close the *Warning* dialog and change the version date for the project.

Cancel – Close the *Warning* dialog without changing the project's version date.

11.10 Importance Measures Wizard

11.10.1 Importance Measures Wizard

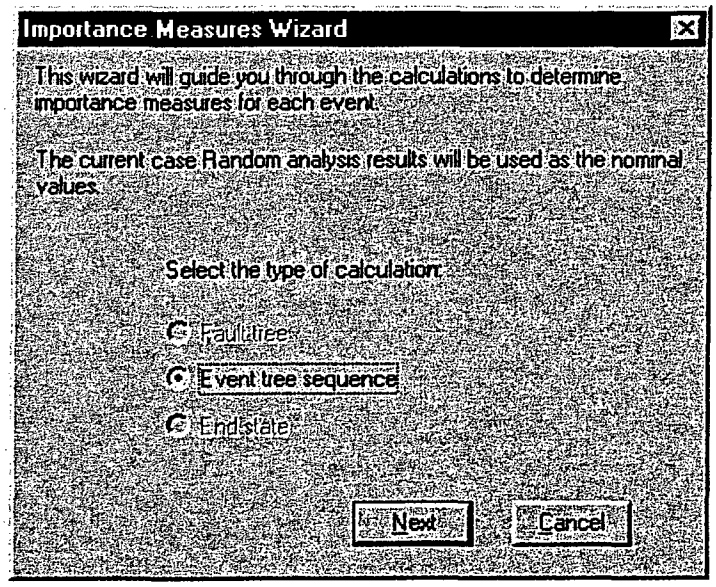
PURPOSE

The *Importance Measures Wizard* is an extension of the basic Importance Measurescalculations originally provided in earlier versions of SAPHIRE. It provides a convenient interface for extending importance concepts for single events into groups of events (components). It also provides the option to readily recalculate rather than just requantify cut sets. This may result in more accurate answers since event interactions will be more fully accounted for.

STEPS

1. From the SAPHIRE menu select **Utilities | Importance Measures Wizard.**

2. The *Importance Measure Wizard* dialog will be displayed.



From this dialog, you can select the type of cut sets to analyze: Fault tree, Event tree sequence, or End state.

Navigation Options

Next - Proceed to the next step in the *Importance Measures Wizard* process.
Cancel - Close the *Importance Measures Wizard* dialog.

11.10.2 Component Group Definition

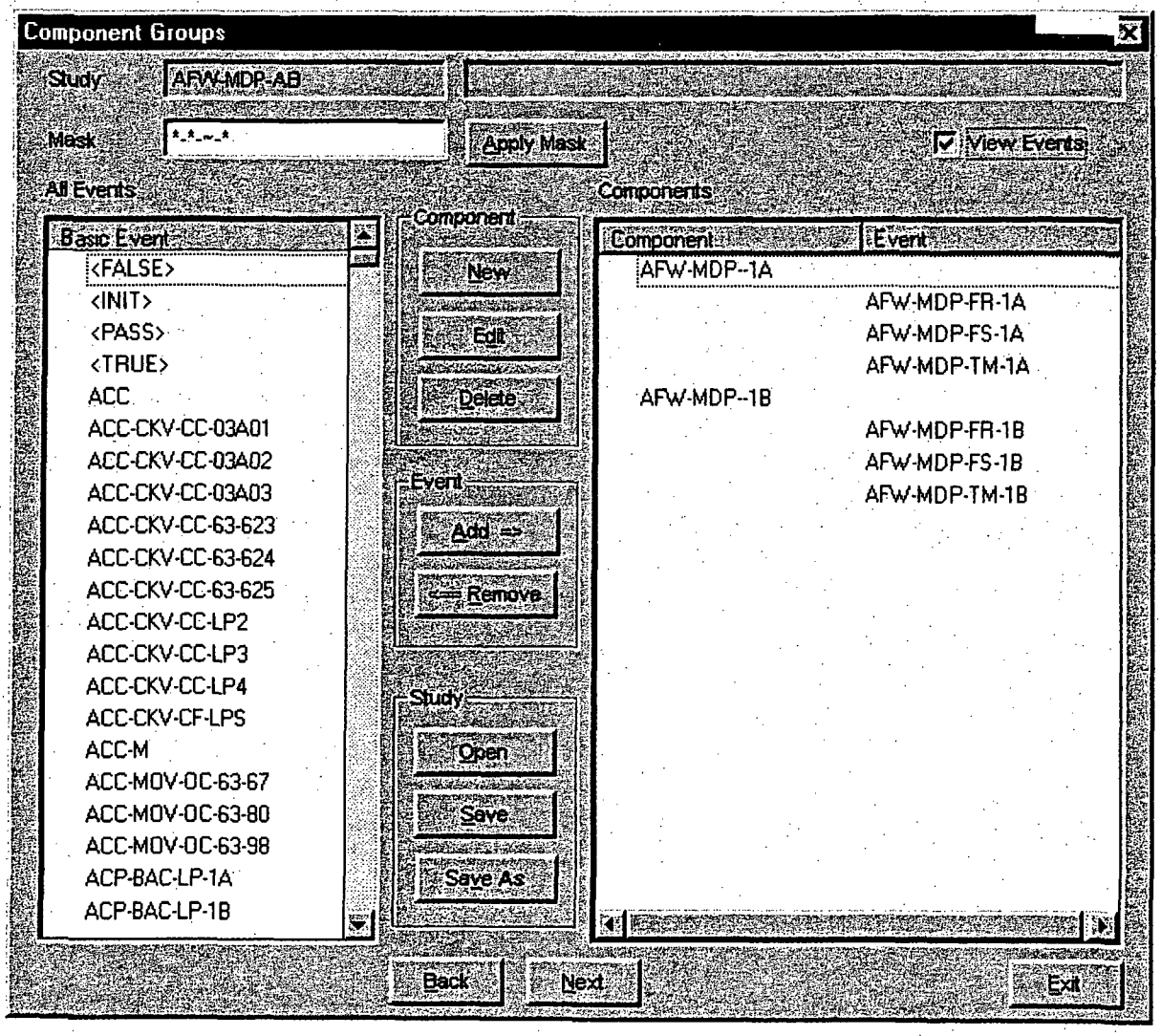
PURPOSE

The *Component Groups* dialog allows you to select and/or group basic events into component groups. These groups will be analyzed as single entities later in the Importance Measures wizard.

The *Mask* feature allows you to quickly group basic events together based on the schema of their names. Then, the *Component* and *Event* features can be used to refine the groupings as needed. Finally, the *Study* options allow you to save the component group definitions for later access.

STEPS

1. From the SAPHIRE menu select **Utilities | Importance Measures Wizard.**
2. The *Importance Measures Wizard* dialog will be displayed. Select the type of cut sets to analyze, and choose the **Next** button. The *Component Groups* dialog will appear.



Study - The name and description of the currently selected study. If there are no studies defined in the project, the study name will read "New", until a study definition is saved.

Mask Options

Mask - Define a wild card mask to collectively assign basic events from the project to component groups based on the naming scheme of the basic events.

Apply Mask - Apply the *Mask* to create component groups. Warning: all existing component groups will be replaced when this option is selected.

View Events - When this option is selected, basic events appear in the *Component* list beneath the component group to which they are assigned. When this option is not selected, the *Component* list contains only component names.

Component Options

New - Create a component group name to include in the *Component* list.

Edit - Rename the selected component group.

Delete - Remove the component group and its events from the study.

Event Options

Add - Add the selected event(s) from the *All Events* list to the selected component group in the *Component* list. A check mark will appear beside each selected event in the *All Events* list, indicating it belongs to a component group.

Remove - Remove the selected event from the component group in the *Component* list.

Study Options

Open - Load a previously saved "study", or component groups definition.

Save - Save the current *study*.

Save As - Save the currently defined component groups definition as a *study*.

Navigation Options

Back - Return to the previous step in the *Importance Measures Wizard* process.

Next - Proceed to the next step in the *Importance Measures Wizard* process.

Exit - Close the *Importance Measures Wizard* dialog.

11.10.3 Component Group Masks

*** (asterisk)** - Designates a group of one or more characters to be used to define a group. Each unique combination of characters indicates a distinct component.

~ (twiddle) - Designates a group of one or more characters *not* to be used to define a group. That is, this part of a basic event name does not contribute to the distinctness of a component.

separator character(s) - Designates boundary marker(s) within the naming scheme.

Example 1

PRA models typically use a naming scheme for basic events that includes the system, component, failure mode, and possibly train, divided by a separator character, such as "-" or "_".

To combine events into groups of distinct train, system, and components, a mask will use a * (**asterisk**) in the character positions for train, system, and component, and a ~ (twiddle) in the failure mode and train positions. The * (**asterisk**) and ~ (twiddle) characters will be split up by a separator character(s), which indicate the boundaries. So, if the naming scheme follows the format SYS_COMP_FM_T, the desired mask would be *_*_~_*

Example 2

Consider the following events:

M-CHILD-JOHN M-CHILD-JAKE
M-CHILD-BOB M-ADULT-JOHN
M-ADULT-JACK F-CHILD-SUSAN
F-CHILD-JANE F-ADULT-SHIRLEY
GEORGE

The mask *-*-~ would produce the following groups:

M-CHILD-
M-CHILD-JOHN M-CHILD-JAKE M-CHILD-BOB

M-ADULT-
M-ADULT-JOHN M-ADULT-JACK

F-CHILD-
F-CHILD-SUSAN F-CHILD-JANE

F-ADULT-
F-ADULT-SHIRLEY

GEORGE
GEORGE
Note: George starts out as a possible match to the mask specifications, but does not conform to the entire mask, so it is placed into a single member group, "as is".

Alternatively, the mask ~-*~ would produce the following groups:

-CHILD-
M-CHILD-JOHN M-CHILD-JAKE M-CHILD-BOB
F-CHILD-SUSAN F-CHILD-JANE

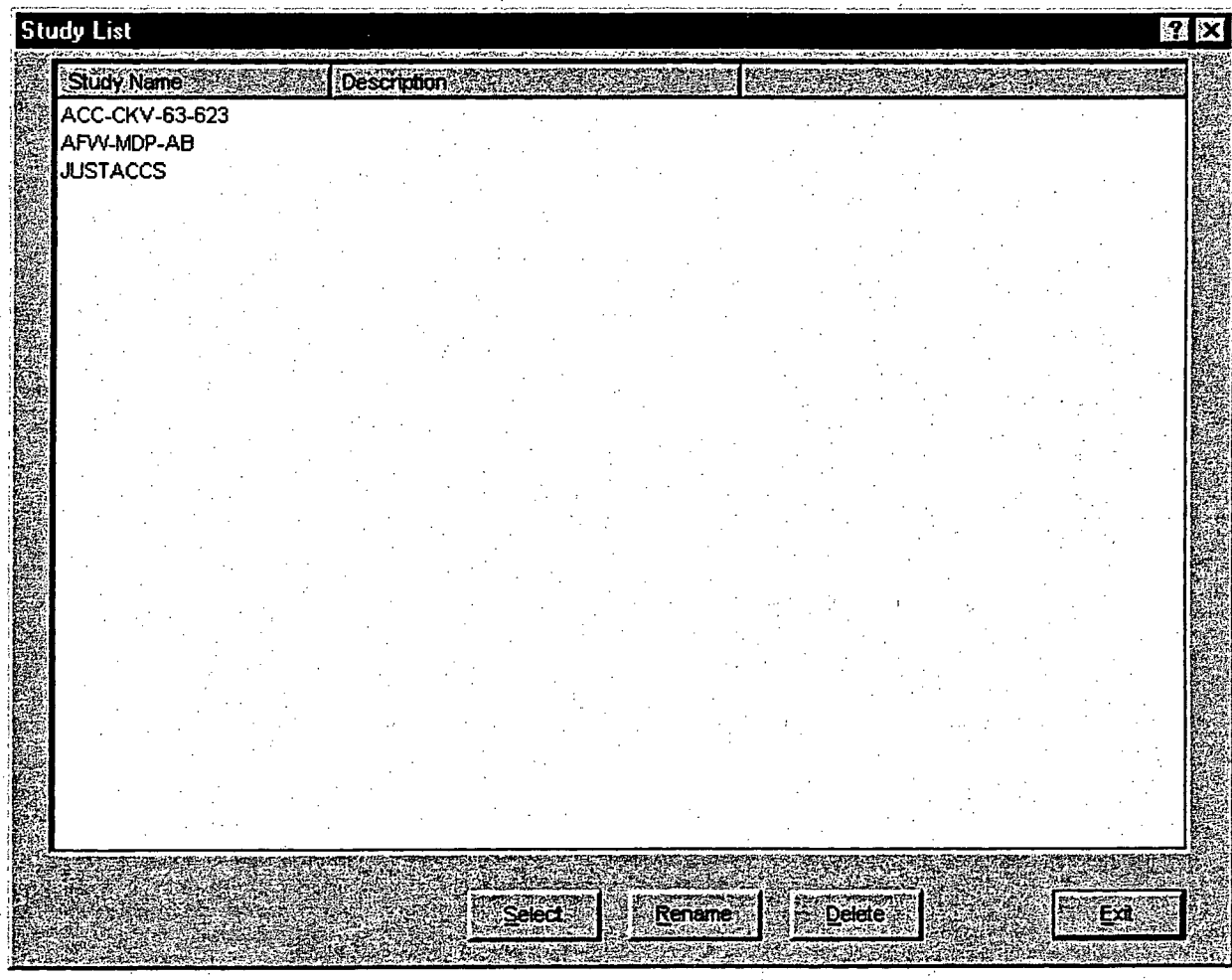
-ADULT-
M-ADULT-JOHN M-ADULT-JACK F-ADULT-SHIRLEY

Note: GEORGE does not appear in any groups, because it never even begins to match the mask specification.

11.10.4 Study List

PURPOSE

The *Study List* dialog allows you select and manage studies created through the Importance Measures Wizard. Each study consists of a collection of events grouped into components.

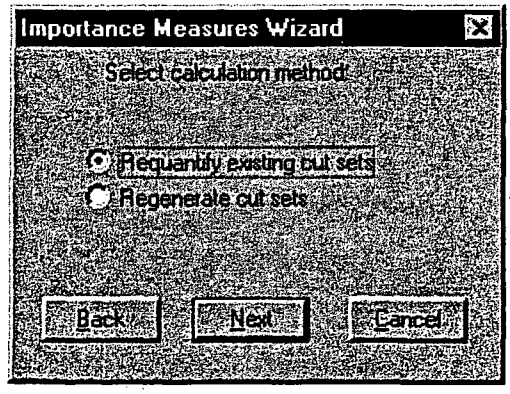


Select - Load the selected study into the *Component Group* dialog.
Rename - Rename the selected study.
Delete - Delete the selected study.
Exit - Return to the *Component Group* dialog without selecting a study.

11.10.5 Select Calculation Method

PURPOSE

From this dialog, you can choose to either requantify the existing cut sets, or generate new cut sets for each component group. The requantify option is generally much faster, but the regenerate option may be more accurate, since any interactions between events will be more fully taken into account.



Navigation Options

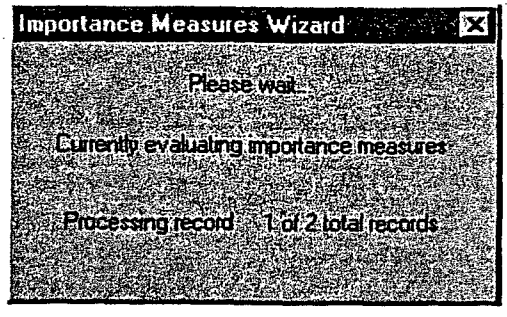
Back - Return to the previous step in *Importance Measures Wizard* process.
Next - Proceed to the next step in the *Importance Measures Wizard.*
Cancel - Cancel the *Importance Measures Wizard* calculations.

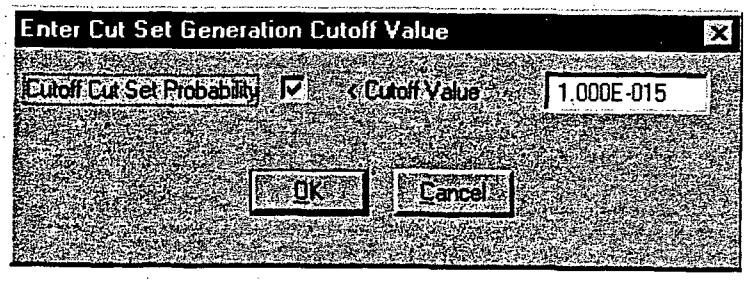
11.10.6 Evaluate Importance

PURPOSE

The *Importance Measures Wizard* will perform the importance evaluations according to the inputs given in earlier steps of the wizard. If the option to regenerate the cut sets was selected from an earlier step in the *Importance Measures Wizard*, you will be prompted to enter the cut set probability. This value should match the one used to generate the nominal cut sets.

Depending upon the size of the cut sets and the number of component groups, this process could take some time to complete.





Navigation Options

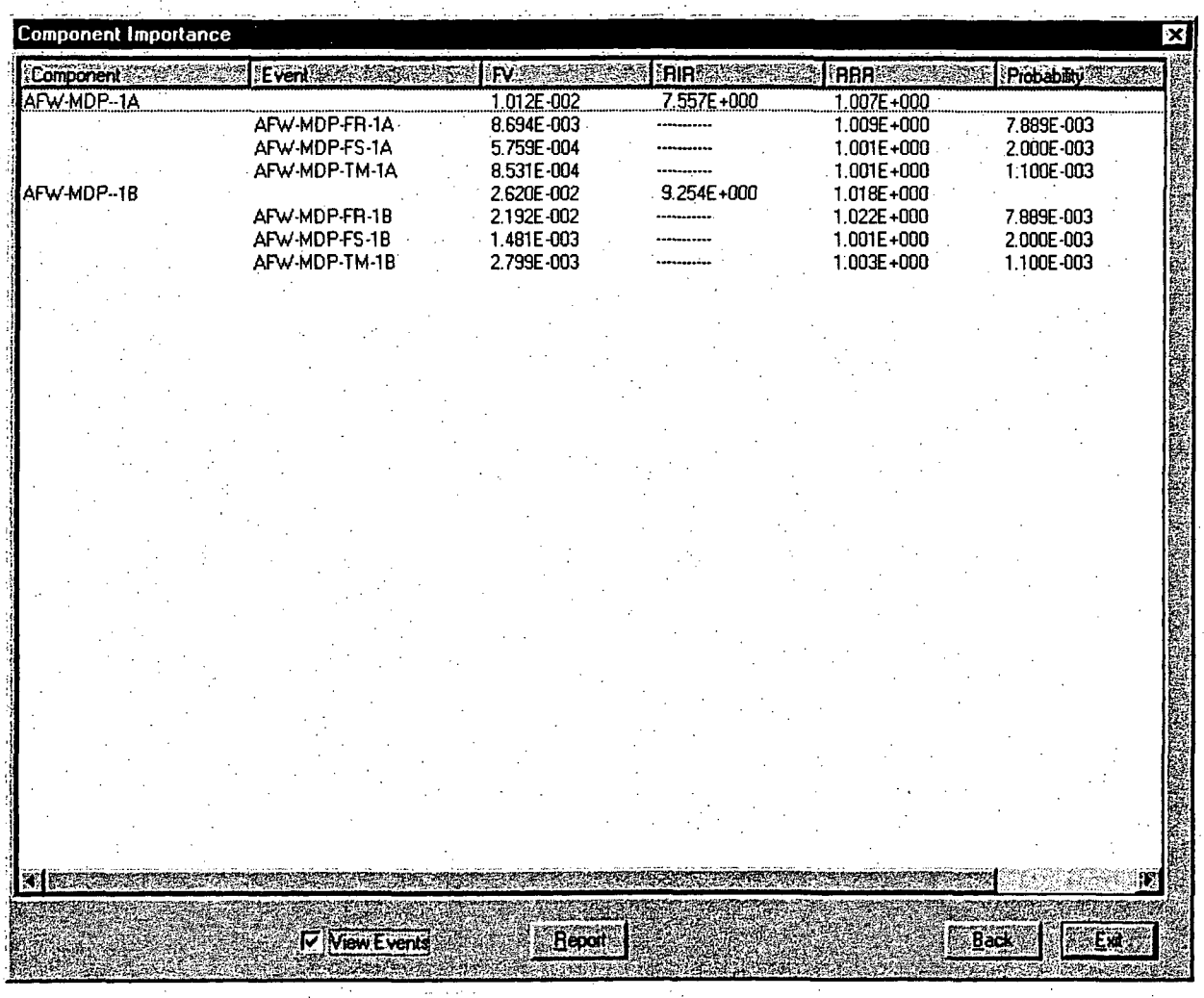
OK - Perform the importance evaluation .

Cancel - Cancel the *Importance Measures Wizard* calculation.

11.10.7 Component Importance Results

PURPOSE

The *Component Importance* dialog displays the results of the importance measures evaluation.



View Events - When this option is selected, basic events appear in the list beneath the component group to which they are assigned. When this option is not selected, the *Component* list contains only component names

Report - Report the results to a comma delimited file. The *Choose file* dialog (similar to the Save As dialog) will be displayed.

Navigation Options

Back - Return to the previous step in the *Importance Measures Wizard* process.

Exit - Close the *Importance Measures Wizard*. The wizard will restore the database to the state it was in when the wizard was invoked, which may take a few moments.

11.11 Sensitivity Wizard

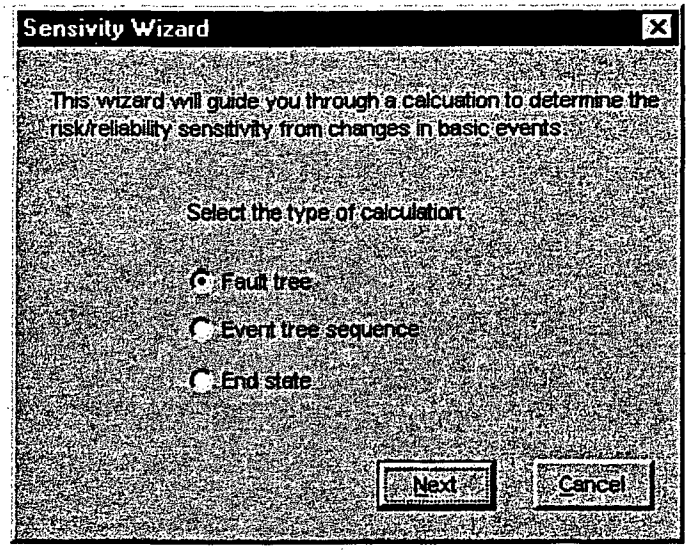
11.11.1 Sensitivity Wizard

PURPOSE

The *Sensitivity Wizard* provides a convenient interface for evaluating changes in overall risk and reliability resulting from changes in basic event probabilities. The wizard gives you the option of recalculating cut sets, rather than just requantifying them. This may result in more accurate answers since any event interactions will be accounted for.

STEPS

1. From the SAPHIRE menu select **Utilities | Sensitivity Wizard.**
2. The *Sensitivity Wizard* dialog will be displayed.



From this dialog, you can select the type of cut sets to analyze: Fault tree, Event tree sequence, or End state.

Navigation Options

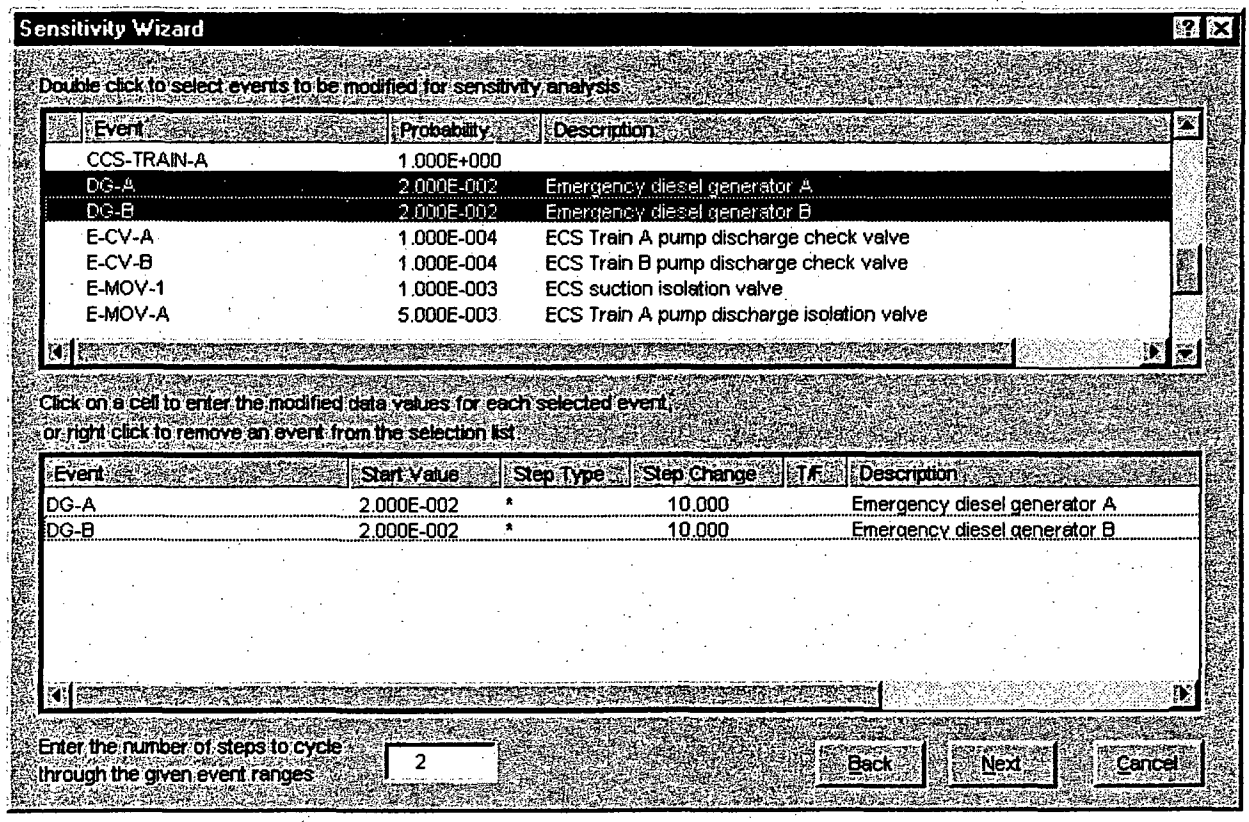
Next - Continue to the next step in the *Sensitivity Wizard* process.

Cancel - Close the *Sensitivity Wizard* dialog.

11.11.2 Select Sensitivity Events

PURPOSE

This option allows you to choose the basic events to study.



The top of this dialog lists all the basic events in the project. From this list you choose the events to study by selecting one or more events, invoking the pop-up menu and choosing the **Add** option. The chosen events will appear in the bottom list, where you indicate the changes to make to those events.

Popup Menu Options

Add - Copy the selected event(s) in the top list to the bottom list.

Delete - Remove the selected events from the bottom list.

Event Change Columns

Event - The name of the event chosen from the top list.

Start Value - Type in the desired starting probability/frequency for the event. The successive columns indicate how to modify this value for sensitivity analysis.

Step Type - Click the cell to toggle the increment type between +(add) and *(multiply). A + will add the specified *Step Change* value to the *Start Value*. A * will multiply the *Step Change* to the *Start Value*.

Step Change - Type in the desired increment value. This value will be initially added or multiplied to the *Start Value*, depending on the chosen *Step Type*.

T/F - Click the cell to toggle between TRUE, FALSE, and BLANK values. A blank setting will not make the event a house event - it will use the *Start* and *Step* values from the other columns. A TRUE or FALSE setting will ignore the *Start* and *Step* values, and will perform the analysis as if the basic event were a house event.

Description - The description of the basic event (for information purposes only).

Number of Steps - Select the number of times to increment the selected events' probabilities, and recalculate the min cuts. The chosen *Step Value* will be used at step 1. Each additional step will add or multiply the *Step Value* to the previous step value, according to the chosen *Step Type*. Events designated as house events will remain house events for each step.

Navigation Options

Back - Return to the previous step in the *Sensitivity Wizard* process.

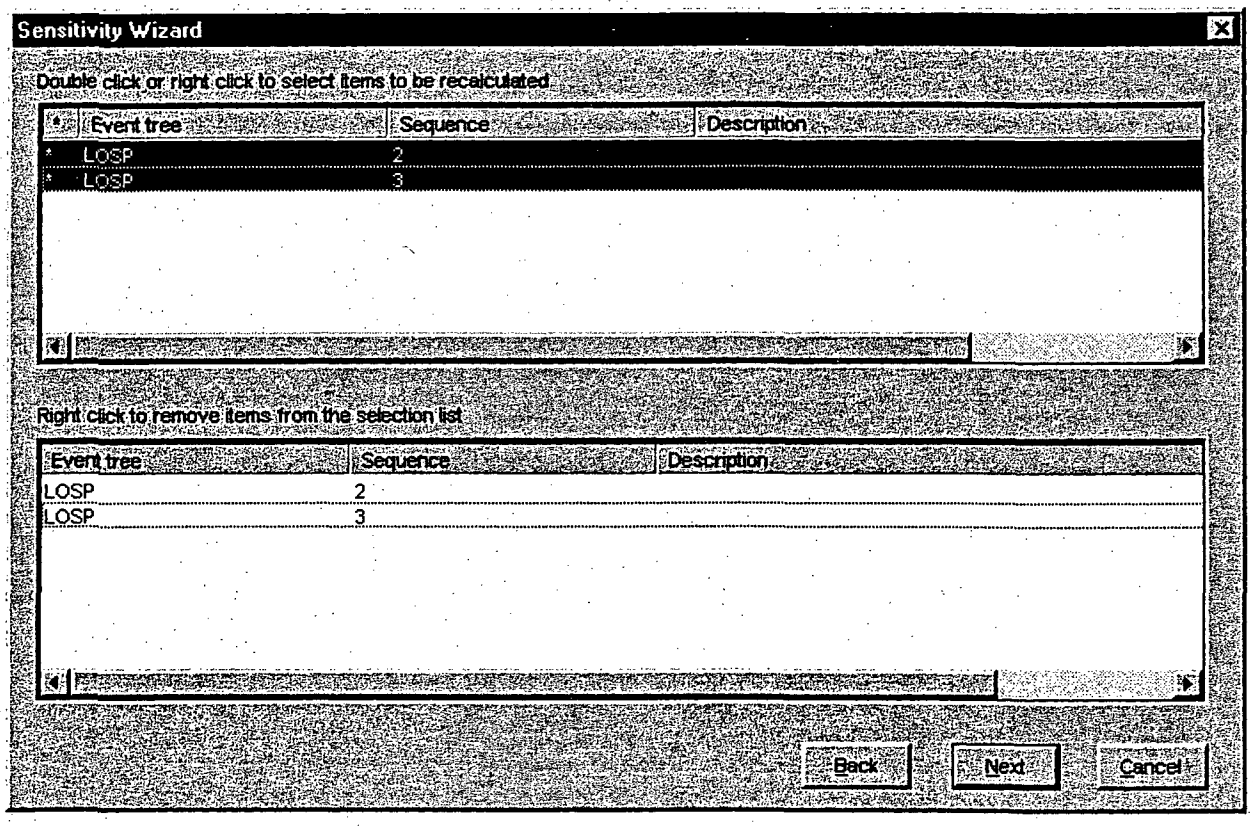
Next - Proceed to the next step in the *Sensitivity Wizard.*

Cancel - Cancel *Sensitivity Wizard* calculation.

11.11.3 Select Items to Recalculate

PURPOSE

This option allows you to choose the sequences, fault trees, sequences, or end states (depending on the type of analysis selected in an earlier step) to recalculate.



The top of this dialog lists all the fault trees/sequences/end states in the project. Each item that has a * (asterisk) in the first column has been detected as possibly affected by the events selected in the previous step. From this list you choose the fault trees/sequences/end states to study by selecting one or more, invoking the pop-up menu, and choosing the **Add** option. The chosen items will appear in the bottom list.

Popup Menu Options

Add - Copy the selected fault tree(s)/sequence(s)/end state(s) from the top list to the bottom list.

Delete - Remove the selected fault tree(s)/sequence(s)/end state(s) from the bottom list.

Navigation Options

Back - Return to the previous step in the *Sensitivity Wizard* process.

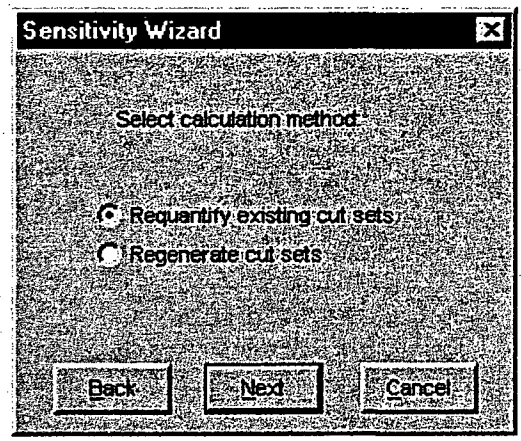
Next - Proceed to the next step in the *Sensitivity Wizard.*

Cancel - Cancel *Sensitivity Wizard* calculation.

11.11.4 Select Sensitivity Calculation Method

PURPOSE

This option allows you to choose whether to recalculate new cut sets or requantify existing cut sets. Requantification is generally faster, but recalculation is generally more accurate.



Navigation Options

Back - Return to the previous step in the *Sensitivity Wizard* process.

Next - Proceed to the next step in the *Sensitivity Wizard.*

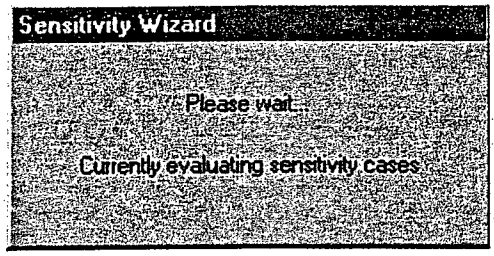
Cancel - Cancel *Sensitivity Wizard* calculation.

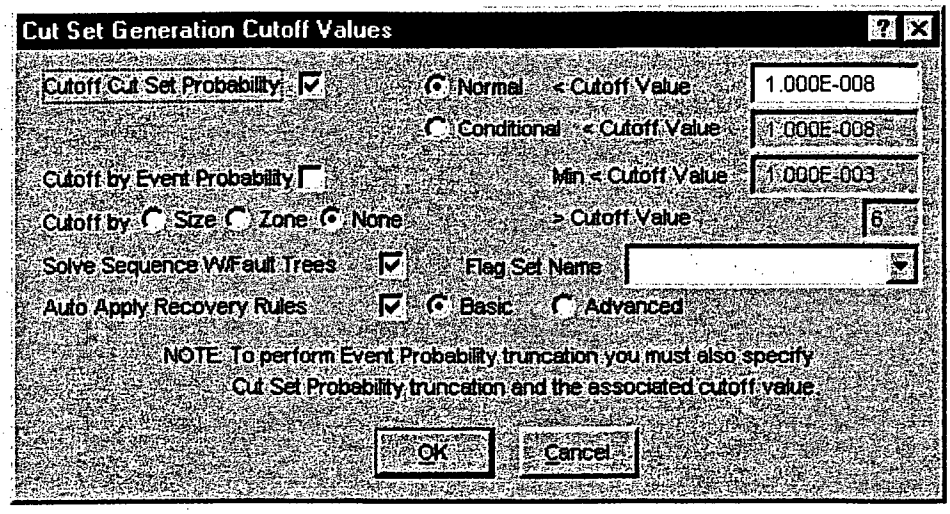
11.11.5 Evaluate Sensitivity

PURPOSE

The *Sensitivity Wizard* will perform the sensitivity evaluations according to the inputs given in earlier steps of the wizard. If the option to regenerate the cut sets was selected in the *Sensitivity Wizard*, you will be prompted to enter the cut set generation options.

Depending upon the size of the cut sets and the number of chosen steps, this process could take some time to complete.





Navigation Options

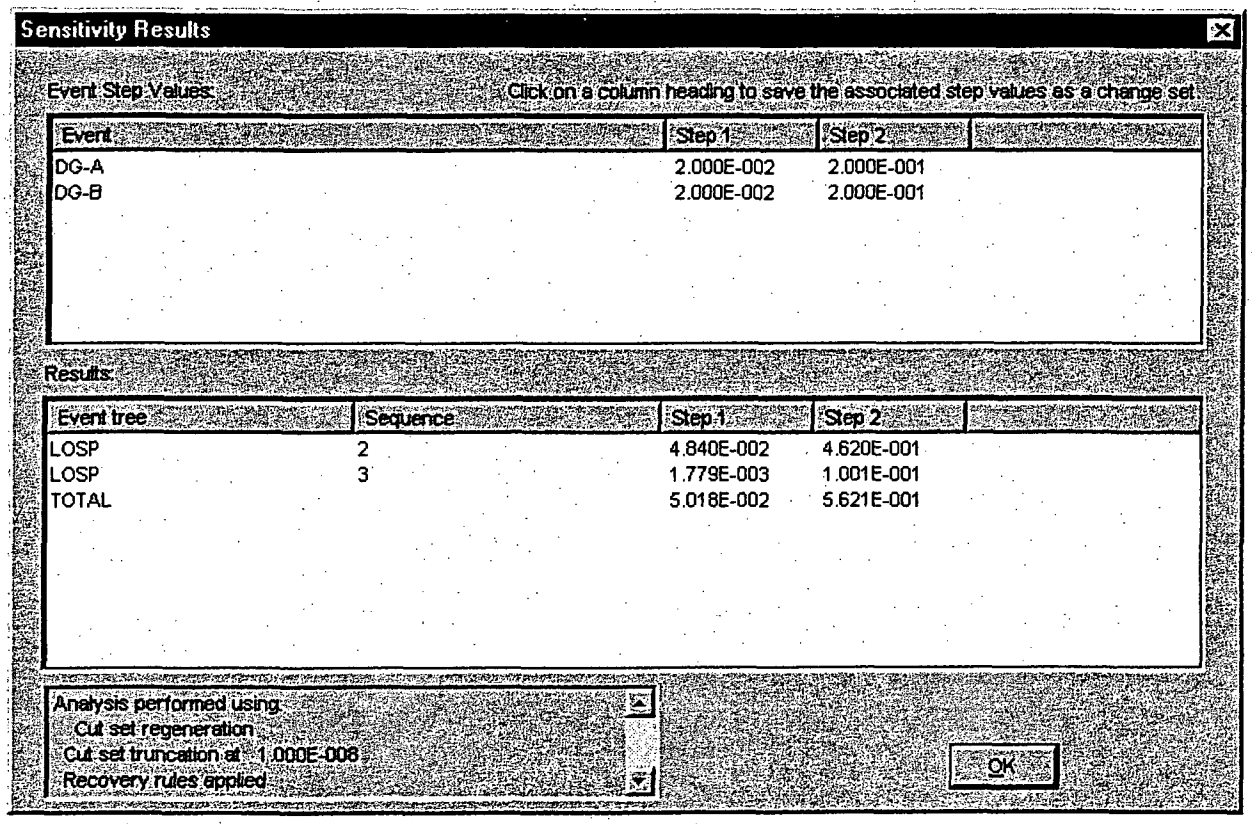
OK - Perform the sensitivity analysis .

Cancel - Cancel *Sensitivity Wizard* calculation.

11.11.6 Sensitivity Results

PURPOSE

The *Sensitivity Wizard Results* dialog presents the calculated results.



The top of this dialog lists the basic events selected for sensitivity analysis. The bottom list contains the selected fault trees/sequences/end states. Additional columns correspond to the number of steps selected in an earlier step. The step values for the basic events represent the probability/frequency used in each step. The step values for the fault trees/sequences/end states represents the min cut upper bound for each step calculated using the basic event values listed for the corresponding step.

Step Column Click - Create a change set containing the event probabilities used in that step. (Each calculation type will be set to 1). The Add Change Set dialog will appear.

Step Value Double Click - Double click on a min cut upper bound result to view the cut sets associated with it. A *View Step Detail* dialog will appear, asking you to choose which step results to view.

Navigation Options

OK - Close the *Sensitivity Wizard.* You may be prompted to regenerate the basic event data. This is recommended in order to return the database to the state it was in prior to invoking the *Sensitivity Wizard.*

12. ACCESSING HELP

12.1 Accessing Help

Included with the SAPHIRE installation is an on-line hypertext help system containing a complete reference to SAPHIRE.

The SAPHIRE help system was designed using the Windows 95 and NT 4.0 help features, based on "Help tabs". The **Contents Tab** is arranged like a table of contents containing a list of topics available in the help system. It is organized with books and pages. The pages are the link to the specified topic. The **Index Tab** provides access to the help system index of topic keywords. The **Find Tab** provides the functionality of a full-text search.

In general, the topic pages in the SAPHIRE system contain a "PURPOSE" section which provides a description of the selected topic and its primary function. This is followed by a "STEPS" section which provides step-by-step instructions for performing the function. Often, a picture of a dialog is included on the topic page. Brief explanations of the options on the dialog are provided below the graphic image and when necessary, links for additional details are presented. On occasion, the graphic image on the topic page will contain hot-spots which provide links to other topic pages or invoke a pop-up window containing a definition. When applicable, a "SEE ALSO" section is included for links to related topics.

Some of the topic pages are general and text-oriented. They provide background information or in-depth discussion about a specific concept and step-by-step directions are not applicable.

The help system is accessed via the Help option on the SAPHIRE menu or through the context-sensitive help on each dialog. The Help menu offers access to the on-line hypertext help system through the **Contents Tab** or **Index Tab**. Context-sensitive help offers a brief description of the selected feature on the current dialog.

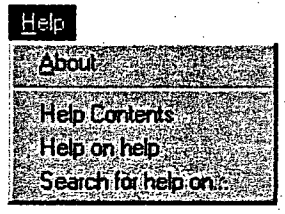
12.2 Help Menu

PURPOSE

This option provides access to the SAPHIRE hypertext help system through a variety of options.

STEPS

1. From the SAPHIRE menu select **Help**.
2. The drop-down menu with available items will be displayed.



About -	Provides product information about the version of SAPHIRE currently running.
Help Contents -	Invokes the SAPHIRE help table of contents. From the table of contents, any "book" or "chapter" can be opened and the topic "page" of interest can be selected.
Search for help on... -	Invokes the SAPHIRE detailed help index. Any keyword in this index can be selected to access information about the associated topic.
Help on help -	Invokes the Windows help system table of contents. Information on how to use a hypertext help system is provided.

12.3 Context-Sensitive Help

PURPOSE

Context-sensitive help usually provides definition-style information about the selected control on a dialog box in a pop-up window. An entire topic page from the SAPHIRE help system may be invoked when the function of the selected control requires a detailed explanation.

STEPS

1. On the active dialog box, choose the **Help** button (), located in the upper right-hand corner of the dialog. The cursor will now be the Help cursor (arrow with a question mark).
2. Click on the desired control (i.e., field, button, etc.) on the dialog.

When you are done viewing help topics:

If a pop-up window was invoked, simply click on the dialog to close the pop-up window and proceed with SAPHIRE operations.

If the SAPHIRE help system was invoked, you can close the help system or leave it open. In either case, because the help system runs independently of the SAPHIRE program, you can click on the dialog and proceed with SAPHIRE operations.

13. CAPABILITIES AND LIMITATIONS

13.1 Capabilities and Limitations

SAPHIRE Database Limits:

(limits are per project unless otherwise indicated)

Number of projects limited by user's available disk space.
Number of cut sets limited by user's available disk space.
Number of basic events 64,000
Number of event trees 64,000
Number of sequences 2,000,000
Number of end states 64,000
Number of fault trees 64,000 - # of basic events
Number of gates 64,000
Number of change sets 10,000
Number of analysis types 16
Size of block text fields limited by user's available disk space.
Lines of rules limited by user's available disk space.

Inputs per OR gate 5,000
Inputs per AND gate 256
Inputs per N/M gate N/ 99, where N = 2 to 98.
Gates/events per fault tree 10,000 (Limit when loading a fault tree through MAR-D).
Number of events per cut set 256
Number of uncertainty samples 99,999
Alpha to Graphics conversion 50 levels deep

Logic display 255 characters wide (dialog display and reporting only)
Path search display 255 characters wide (dialog display and reporting only)

General Field Limits:

Name references - Unless specified otherwise, limited to 24 uppercase characters. If lowercase characters are entered, they will be converted by SAPHIRE to uppercase. Embedded spaces (or blanks) are not allowed.
Descriptions - 120-character upper- or lowercase alphanumeric characters. Embedded spaces (or blanks) are allowed.
File names - Long file names are supported.
Floating point fields - Use scientific notation (e.g., 1.0E-15).

Information on specific fields can be found in the **Utilities | Load and Extract** section.

14. RECOVERY RULES

14.1 Recovery Rules Usage And Examples

PURPOSE
The Recovery Rule editor allows you to create rules that affect existing cut sets in a "post-processing" fashion. The rule-based editor is available for both fault tree and sequence cut sets.

The recovery rules are "free-form" logic rules that allow for the alteration or deletion of fault tree or sequence cut sets. They follow a format similar to the structure that is found in traditional programming languages (e.g., BASIC or PASCAL). As such, the ability exists to define "macros" and "if...then" types of structures. The rules are entered in a free-form text editor within SAPHIRE. They can also be exported and imported through the SAPHIRE **Utility | Load and Extract** option.

The rules can be used for probabilistic risk assessment techniques such as the automated inclusion of sequence recovery events or common-cause failure cut sets, or elimination of mutually exclusive events (e.g., impossible combinations of events). They may be developed for a particular fault tree, all fault trees (project fault tree rules), a particular sequences, a single event tree, or all sequences (project sequence rules). Initiating events, basic events, and recovery events can all be added into the relational database directly from within the rule editor.

When the rules are applied, existing fault tree or sequence cut sets are scanned for cut sets matching the search criteria defined in the rule. The rule is used to modify the cut sets matching the search criteria. This searching process is a multi-step procedure:

> If the first cut set in the list matches the search criteria in the first rule, the applicable modifications defined in the first rule are made to the cut set. Once a cut set qualifies for a search criterion it is not affected by any of the remaining rules.
>
> If the first cut set in the list does not match the search criteria in the first rule, the second rule (SAPHIRE processes the rules from top to bottom) is evaluated.
>
> After all the rules are evaluated for the first cut set, the second cut set in the list is evaluated, starting again with the first rule. This process is repeated until all cut sets are evaluated.

Since the recovery rules have the capability to remove basic events from individual cut sets, application of the rules could result in non-minimal cut sets. Consequently, it is recommended practice to re-minimize the cut sets after applying recovery rules. Thus, the typical steps you would take in performing an analysis using recovery rules are:

> Finalize logic models and data changes and then Generate changes
>
> Generate fault tree or sequence cut sets
>
> Apply recovery rules to applicable fault trees or sequences
>
> Perform a cut set update to fault tree or sequence cut sets
>
> Perform uncertainty analysis
>
> Display or report results.

14.2 Search Criteria Examples - Basic Recovery

These examples are used based on basic events X, Y, and Z.

Search Criteria	Meaning of Search Criteria
X	Basic event X appears in the cut set
/X	Success of event X appears in the cut set
~X	Basic event X never occurs in the cut set
X + Y	Either event X or Y appear in the cut set
X * Y	Both events X and Y appear in the cut set
~X * Y	Y does appear and X does not appear in the cut set
X * (Y + Z)	Either X and Y or X and Z appear in the cut set
always	This pre-defined macro name means the criteria is always met

14.3 General Recovery Rule Structure

Shown below are two examples of the actual basic rule structure and syntax.

The following example shows how the "if...then" rule structure can be used to check if a failure of an electric power bus event appears in a cut set. If the event does occur, the rule will muliply the cut set containing the event by an operator non-recovery probability named "BUSREC" via the "recovery" keyword.

```
| The "if-then" Rule Structure:
| This rule adds a recovery action BUSREC
| when electric bus B or C is failed
if EL-BUS-B + EL-BUS-C then
recovery = BUSREC;

endif
```

The following example demonstrates the use of the "elsif" keyword. For this second example, the rule first checks to see if two diesel generators (DG) maintenance events appear in the same cut set. If the do, the cut set is deleted since plant technical specifications do not permit two diesel generators to be out of service simultaneously for routine maintenance. Or, if the cut set does not qualify under the first search criteria, the rule checks to see if the random failure events, DG-1-RAND and DG-2-RAND appear in the same cut set. If they do, the rule will first copy the original cut set, then remove them and add the common-cause event, "DG-CCF-1AND2."

```
| The "if-then-elsif" Structure:
| This rule deletes the cut set if both diesel
| generators are out for maintenance.
| If the two DGs fail randomly, add a
| common-cause event.
if (DG-MAINT * DG-2-MAINT) then
DeleteRoot;

elsif (DG-1-RAND * DG-2-RAND) then
| Copy the original cut set, remove the
| two failure events, then add CC
```

CopyRoot;

DeleteEvent = DG-1-RAND;

DeleteEvent = DG-2-RAND;

AddEvent = DG-CCF-1AND2;

Endif

14.4 Recovery Rules - Add Recovery Actions

In a PRA, cut sets for accident sequences are generated using fault tree and event tree logic. Since most PRAs are analyzed in a failure space, each cut sets represents the minimal set of components or fault trees that have to fail (given a particular initiating event) in order to result in an undesired condition (e.g., core damage). As such, operator actions that could prevent the accident sequence from progressing to the point of an accident may not be specifically included in the logic models. to model the PRA accident sequences as accurately as practical, the analyst will typically want to apply recovery events (such as appropriate) to the accident sequence cut sets.

The recovery events represent the probability that the operator or operators *fail* to successfully prevent the accident by restoring one or more of the failed components in the sequence cut sets. Consequently, the recovery events are frequently called the non-recovery probability events.

To demonstrate the potential uses of the recovery rules, the example here shows how the rules could be used to include recovery actions on specific cut sets in a particular sequence. As shown, the rule will add the NRAC-12HR operator recovery event (recovery of DGs within 12 hours) to those cut sets that contain LOSP as the sequence initiating event and failure of either DG A or DG B. This rule would probably be typed into the event tree sequence rule editor for the sequence of interest.

(Basic Rule syntax)

| Search on LOSP initiator and failure of
| diesel generators A or B.
if init(LOSP) * (DG-A + DG-B) then
recovery = NRAC-12HR;

endif

Once this rule is saved in the project database, SAPHIRE will evaluate all the sequence cut sets and then automatically include the recovery actions as specified in the recovery rule. The "if...then" line can be as complex as needed.

14.5 Recovery Rules - Mutually Exclusive Event Removal

To demonstrate the potential uses of the recovery rules, the example here shows how the rules could be used to remove a particular cut set from the cut set list.

NOTE: There may be instances where a cut set should be removed because of mutually exclusive events (i.e., the combination of basic events should not occur).

(Basic Rule syntax)

```
| This rule could be placed in either (or both) the
| fault tree project rules or the event tree project rules.
| Define a macro to get those cut sets that have
| combinations of two motor driven pumps out
| for maintenance.
PUPMS-IN-MAINT = MDP-A-MAINT * MDP-B-MAINT;

| Search for the maintenance vents and then
| delete the cut set.
if PUMPS-IN-MAINT then
| Delete the cut set
DeleteRoot;

endif
```

14.6 Recovery Rules - Common-Cause Failure Modeling

To demonstrate the potential uses of the recovery rules, the example here shows how the rules could be used to incorporate common cause modeling into the PRA.

Common cause failure modeling, one part of dependent failure analysis, is a standard PRA modeling technique. It attempts to model simultaneous failures of multiple components due to a single cause (i.e., common mode failure). Recovery rules can be created to search for cut sets containing specific groups of independent, random failure events and replace them with a single common-cause basic event in the cut sets.

Note these two important issues when using recovery rules for common-cause failure modeling:

The usefulness of the recovery rules for common-cause failure modeling is limited by the fact that groups of independent failures must be present in the cut sets in order for a rule to replace the independent failures with a single common cause failure event.

If a probability or size truncation is specified when generating fault tree or sequence cut sets, the independent failure cut sets may be omitted that would be above the truncation limit after being modified by the rule.

(Basic Rule syntax)

```
| The search criteria identifies the failure
| combination of two auxiliary feedwater pumps.
| If these two basic events are found in a cut set
| then a new cut set will be created that replaces
| the independent failures of the two pumps with
| a single common-cause basic event. This rule
| could be placed in either (or both) the fault tree
| project rules or the event tree project rules.

| Define a macro to only pick up those cut sets
```

```
| that have combinations of AFW-PUMP-A
| and AFW-PUMP-B.
CCF-AFW-PUMPS = AFW-PUMP-A * AFW-PUMP-B;

| Search for the AFW pump basic events and
| make a new cut set with the CCF event.
if CCF-AFW-PUMPS then
| First make a copy of the original cut set
CopyRoot;

| Now remove the two independent failure events
DeleteEvent = AFW-PUMP-A;
DeleteEvent = AFW-PUMP-B;

| Now add the CCF event
AddEvent = AFW-PUMP-CCF;
endif
```

14.7 Keywords and Symbols

14.7.1 Keywords and Symbols - Basic Recovery Rules

Below is a list of keywords and symbols used in the SAPHIRE slice rules. Note the spelling of each keyword.

if-then -	Indicates a search criteria is being specified.
endif -	Indicates the end of a particular rule.
else -	Specifies some action to be taken if all the search criteria are not met. The "else" should be the last condition in the recovery rule.
elsif -	Specifies an alternative search criteria.
always -	Indicates that every cut set that is being evaluated satisfies the search criteria.
init()-	Used in the search criteria to indicate that a sequence cut set has a particular initiating event.
~ (never) -	Used in the search criteria to indicate that a particular event ***will not*** be in the cut set that is being evaluated.
/ (complement) -	Represents a complemented event (i.e., the success of a failure basic event).
| (comment) -	Represents a comment contained in the rules.
; (end macro) -	Indicates the end of a macro line or a line that modifies the cut set being evaluated.
*** (AND)** -	Indicates the logical AND command.
+ (OR) -	Indicates the logical OR command.
( ) -	Indicate a specific grouping of items.
recovery = -	Indicates that a recovery event is going to be added to the cut set being evaluated.
AddEvent = -	Indicates that an event will be added to the cut set being evaluated
DeleteEvent = -	Indicates that an event will be deleted from the cut set being evaluated.
NewCutset; -	Indicates that a new, empty cut set will be added to the list of cut sets.
DeleteRoot; -	Indicates that the "found" cut will be deleted.

CopyCutset; - Indicates that the cut set being evaluated will be copied and added to the list of cut sets.

CopyRoot; - Indicates that the original cut set will be copied

MACRO - A user-definable keyword that specifies a search criteria.

14.7.2 if-then

(Basic Rule syntax)

This keyword indicates that a search criteria is being specified.

if "search criteria" **then**
perform some action on each cut set...;

endif

14.7.3 endif

(Basic Rule syntax)

This keyword indicates that the end of a particular rule.

if "search criteria" then
perform some action on each cut set...;

endif

14.7.4 else

(Basic Rule syntax)

This keyword specifies that some action to be taken if all the search criteria are not met. The "else" should be the last condition in the recovery rule.

if "search criteria" then
perform some action on each cut set...;

else
perform some other action on each cut set if "search criteria" not met...;

endif

14.7.5 elsif

(Basic Rule syntax)

This keyword specifies an alternative search criteria. Any number of "elsifs" can be used within a recovery rule.

if "search criteria" then
perform some action on each cut set...;

elsif "2nd search criteria" then
perform some other action on each cut set …;

elsif "3rd search criteria" then
perform some other action on each cut set …;

endif

14.7.6 always

(Basic Rule syntax)

This keyword indicates that every cut set that is being evaluated satisfies the search criteria.

if **always** then
perform some action on each cut set…;

endif

14.7.7 init()

(Basic Rule syntax)

This keyword is used in the search criteria to indicate that a sequence cut set has a particular initiating event.

if **init**(INITIATOR-NAME) "and optional other search criteria" then
perform some action on each cut set…;

endif

14.7.8 ~ (never)

(Basic Rule syntax)

This symbol is used in the search criteria to indicate that a particular event ***will not*** be in the cut set that is being evaluated.

if (~SEARCH-CRITERIA) "and optional other search criteria" then
…

The search criteria will be satisfied for cut sets that do not contain SEARCH-CRITERIA (and also contains the optional ("other search criteria"). SEARCH-CRITERIA may be either an initiating event, basic event, macro, or logic expression.

14.7.9 / (complement)

(Basic Rule syntax)

This symbol is used to represent a complemented event (i.e., the success of a failure basic event).

if (/BASIC-EVENT) "and optional other search criteria" then

The search criteria will be satisfied for all cut sets that contain the complement of BASIC-EVENT (and also contains the optional "other search criteria").

14.7.10 | (comment)

(Basic Rule syntax)

This symbol is used to represent a comment contained in the rules. Everything on a line to the right of this symbol will be ignored by the rule compiler. Blank lines are encouraged in the editor to improve readability and comments can be entered following the "|" character.

```
| Place your comments here!

| Note that blank lines are also permissible.
```

14.7.11 ;(end macro)

(Basic Rule syntax)

This symbol is used to indicate the end of a macro line or a line that modifies the cut set being evaluated.

```
| usage for a macro command
MACRO-NAME = "search criteria";

| usage for a cut set modification line
recovery = RECOVERY-EVENT;
```

14.7.12 * (AND)

(Basic Rule syntax)

This symbol is used to indicate the logical AND command.

if SEARCH-CRITERIA1 * SEARCH-CRITERIA2 then

The search criteria will be satisfied for all cut sets that match SEARCH-CRITERIA1 and SEARCH-CRITERIA2. The SEARCH-CRITERIA# may be either an initiating event, macro, or logic expression.

14.7.13 + (OR)

(Basic Rule syntax)

This symbol is used to indicate the logical OR command.

if SEARCH-CRITERIA1 + SEARCH-CRITERIA2 then

The search criteria will be satisfied for all cut sets that match either SEARCH-CRITERIA1 or SEARCH-CRITERIA2. The SEARCH-CRITERIA# may be either an initiating event, macro, or logic expression.

14.7.14 ( ) (group)

(Basic Rule syntax)

These symbols together indicate a specific grouping of items.

if (A + B) * (C + D) then

The search criteria above would return all cut sets that contain:
[A * C], [A * D], [B * C], or [B * D].

14.7.15 NewCutset;

(Basic Rule syntax)

This keyword indicates that a new, empty cut set will be added to the list of cut sets. The new cut set then becomes the cut set that is being evaluated. Note the particular capitalization of the keyword.

if "search criteria" then
NewCutset;
now make additions to the empty cut set...
endif

14.7.16 DeleteRoot;

(Basic Rule syntax)

This keyword indicates that the original cut set (i.e., that cut set which satisfies the search criteria) will be deleted. Note the particular capitalization of the keyword.

if "search criteria" then
DeleteRoot;
endif

14.7.17 CopyCutset;

(Basic Rule syntax)

This keyword indicates that the cut set being evaluated will be copied and added to the list of cut sets. This copied cut set then becomes the cut set that is being evaluated.

if "search criteria" then
CopyCutset;
now make modifications to a copy of the cut set...
endif

14.7.18 CopyRoot;

(Basic Rule syntax)

This keyword indicates that the original cut set (i.e., that cut set which satisfied the search criteria) will be copied. The copied cut set will then become the cut set that is being evaluated.

if "search criteria" then
CopyRoot;
now make modifications to a copy of the original cut set...
endif

14.7.19 Recovery Rules - Keyword Examples

The example here illustrates how cut sets can be changed by the use of some of the recovery rule keywords occurring in the basic rule syntax.

For this example, assume that only a single cut set matches the search criteria. This cut set starts with a single basic event, A, and is called the Root cut set.

Step	Keyword that is applied	Resulting cut set(s)	Comment
1	AddEvent = B;	(1) A * B	Event B is attached to the "currently evaluated" cut set.
2	NewCutset;	(1) A * B (2) *blank*	A new blank cut set is included in the list of cut sets. This new cut set now becomes the "currently evaluated" cut set.
3	AddEvent = C;	(1) A * B (2) C	Event C is attached to the "currently evaluated" cut set.
4	DeleteRoot;	(1) C	The Root cut set is removed.
5	CopyCutset;	(1) C (2) C	A new cut set is included in the list of cut sets that is a duplicate of the old "currently evaluated" cut set.
6	AddEvent = D;	(1) C (2) C * D	Event D is attached to the "currently evaluated" cut set

15. PARTITION RULES

15.1 Partition Rules Usage and Examples

PURPOSE

The partition rules test the existing sequence cut sets for the presence or absence of specific combinations of basic events or initiating events, and assigns characters in the end state name when the criteria are met. This allows end state names to be built as the rules are applied.

TOPICS

Symbols
Search Criteria Examples
Keywords and Rule Structures
Partitioning Rule Example

15.2 Rule Editor Symbols

| (comment)	Denotes a comment line
* (AND)	Logical AND operator
+ (OR)	Logical OR operator
~ (never)	Logical operator for "never" or "not present"
/ (complement)	Complement
; (end macro)	Indicates the end of a macro line or a line that modifies the criteria being evaluated.
( )	Parentheses for grouping terms

15.3 Search Criteria Examples

(Basic Rule syntax)

Search Criteria	Meaning of the Search Criteria
DG-A	Basic event DG-A (failure)
/DG-A	Complemented basic event DG-A (success)
~DG-A	Failure of DG-A is not in the cut set
init(LOSP)	Initiating event with the name LOSP
SYSTEM(ECS)	Fault tree with the name ECS

15.4 Keywords and Rule Structures

15.4.1 Keywords and Rule Structures

Below is a list of keywords used in the SAPHIRE partition rules (Basic Rule syntax). Note the spelling of each keyword. The General Rule Structures describe how the keywords can be used.

if-then - Indicates that a search criteria is being specified.
always - Indicates that the search criteria is always satisfied.
elsif - Specifies an alternative search criteria.
else - Specifies that some action to be taken if all the search criteria are not met.
Cast Operators - Define substitutions for macros, fault trees, end states, events, and initiating events.
CurrentPart() - Creates a different end state.
GlobalPartition - Partitions all cut sets in a sequence to an end state.
Macro - A user-definable keyword that specifies a search criteria.

15.4.2 General Rule Structure

(Basic Rule syntax)

The information between the "if-then " and the "endif" clause defines the end state for the cut set. The user can either specify an entire end state name as in the example or substitute only certain character positions in the name.

```
if always then
partition = "NO-FLOW-";
endif
```

The following rules demonstrate the substitution process.

```
if NO-FLOW then
partition = "????????END";
elsif ~EVENT7 then
partition = "????????START";
else
partition = "????????NONE";
endif
```

This rule will result in one of three end states being assigned to a cut set. Either, "NO-FLOW-END", "NO-FLOW-START", or "NO-FLOW-NONE". A rule can contain some additional clauses as indicated above.

15.4.3 if-then

(Basic Rule syntax)

This keyword indicates that a search criteria is being specified. The information between the "if" and the "endif" clause defines the tops to be replaced. There is no limit on the number of replacements that can be defined between these clauses.

This is an example of the if-then rule structure along with the use of wildcard characters.

```
| This rule adds -SBO as characters 4 through 7
| of the end state name when both DG-A and
| DG-B are present in the cut sets. The ???
| are wildcards (or placeholders) in the end state
| name. (The end state name is initially blank.)
if DG-A * DG-B then
partition = "???-SBO";

endif
```

The partition statement must end with a semicolon. The end state name must be less than or equal to 24 characters in total length. The end state characters are enclosed in quotation marks.

15.4.4 always

(Basic Rule syntax)

This keyword indicates that the search criteria is always satisfied. The substitutions following this condition are always made.

```
| This rule adds END as the first 3 characters
| in every cut set.
if always then
partition = "END";

endif
```

The partition statement must end with a semicolon. The end state name must be less than or equal to 24 characters in total length. The end state characters are enclosed in quotation marks.

15.4.5 elsif

(Basic Rule syntax)

This keyword specifies an alternative search criteria. Any number of "elsifs" can be used within a rule. The first condition that is met is used and the others are ignored.

```
| This rule adds characters 4 through 7 to
| the end state name. When both DG-A
| and DG-B are failed, -SBO is added.
| When DG-A is failed (but not DG-B),
| characters -DGA are added. When DG-B
| is failed (but not DG-A), characters -DBG
| are added.
if DG-A * DG-B then
partition = "???-SBO";

elsif DG-A then
partition = "???-DGA";

elsif DG-B then
partition = "???-DGB";

endif
```

Note that the cut sets that do not contain DG-A or DG-B are not assigned to any end state in the rule above. See the "else" statement to resolve this.

15.4.6 else

(Basic Rule syntax)

This keyword specifies that some action to be taken if all the search criteria are not met. The "else" should be the last condition in the rule.

```
| This rule adds characters 4 through 7 to
| the end state name. When both DG-A
| and DG-B are failed, -SBO is added.
| When DG-A is failed (but not DG-B),
| characters -DGA are added. When DG-B
| is failed (but not DG-A), characters -DBG
| are added. For all others, characters
| -FLW are added.
if DG-A * DG-B then
partition = "???-SBO";

elsif DG-A then
partition = "???-DGA";

elsif DG-B then
partition = "???-DGB";

else
partition = "???-FLW";

endif
```

Note that the cut sets that do not contain DG-A or DG-B are assigned to the ???-FLW end state by the "else" statement.

15.4.7 Cast Operators

(Basic Rule syntax)

SAPHIRE provides type cast functions for macros, fault trees, end states, events, and initiating events. If you want to change the default assumptions, then you can "cast" the names to something else by enclosing the name in parenthesis with a cast name preceding it.

For instance, if the name "EVENT5" was really the name of a fault tree then you could change its default type with the following syntax; "SYSTEM(EVENT5)". The editor will then treat this name as a fault tree.

15.4.8 CurrentPart()

(Basic Rule syntax)

The "current partition" rule structure uses the end state created by a partition to append on other information and create a different end state. This rule can use wildcards as part of its search criteria.

```
| This rule creates an end state containing all
| cut sets with the initiating event LOSP. The
| rule then creates a new end state using the,
| current end state and cut sets containing both
| DG-A and DG-B.
|
if init(LOSP) then
partition = "LOSP-1";
endif

if CurrentPart(L???-?) * DG-A * DG-B then
partition = "SBO-SEQ";
endif
```

15.4.9 GlobalPartition Keyword Usage

15.4.9.1 GlobalPartition - Transfer to Level 2 Tree

(Basic Rule syntax)

Global partition rules are geared more for Level 2 studies since the end state that is created is also an event tree with the same name. The event tree that is created uses the end state frequency as its initiating event frequency and then transfers to a Level 2 event tree. The end state event tree can be looked at as an event tree which transfers Level 1 information to Level 2 trees.

This rule creates an end state event tree to be used by a Level 2 event tree already created.

```
| LEVEL2TREENAME can be viewed as
| the name for a Level 2 event tree that
| will use the end state frequency gathered
| in the end state CD-SEQ3 for its analysis.
|
if init(LOSP) * SYSTEM(ECS) * SYSTEM(CCS) then
GlobalPartition = "CD-SEQ3";
transfer = LEVEL2TREENAME;
endif
```

NOTE: Global partitioning is designed for partitioning cut sets into end states based upon sequence logic.

15.4.9.2 GlobalPartition - Wildcards

(Basic Rule syntax)

The use of wildcards can also be used by the global partition rules. This adds the ability of partition rules to test on the initiating event name.

```
| This rule will search on the LOSP
| initiating event and gather cut sets
| into an end state.
|
if "L???" then
GlobalPartition = "CD";
endif
```

NOTE: Global partitioning is designed for partitioning cut sets into end states based upon sequence logic.

15.4.10 Macro Structures

15.4.10.1 Macro Structures

(Basic Rule syntax)

Macros can be used to streamline complex rules. A macro is simply a statement to define a search criteria (or variable) and assign a name to the search criteria. The macro name must be all uppercase, must be 24 characters or less, and must not include any of the restricted characters (e.g., a space, *, ?, \, /). The macro line can wrap around to more than one line, but must end with a semicolon.

The macro defined by the following syntax,

```
NO-FLOW = (EVENT1 + EVENT2) * (EVENT3 + /EVENT4);
```

indicates that the macro "NO-FLOW" is to be true if the following conditions are true. (1) EVENT1 or EVENT2 appear in the cut set and EVENT3 appears or EVENT4 appears as a complemented event. Once a macro is defined, it can be used in an "if" test. The following rule is an example of an "if" test using a macro.

```
if NO-FLOW then
partition = "NO-FLOW-END";
endif
```

The following is an example using the "?" wildcard character for string substitution. This character is a place holder character and has no effect on the resulting string substitution.

```
| Define a macro named ALL-DGS
ALL-DGS = DG-A * DG-B;
|
| Use the macro in a rule
if ALL-DGS then
partition = "???-SBO";
endif
```

You can also use the ~ (never) character to indicate that the item following the character is not in the current cut set.

15.4.10.2 Using ~ (never) With a Macro

(Basic Rule syntax)

When creating a rule that applies when the events in the macro do not occur, use the ~ (never) symbol. Do not "complement" a macro.

```
| The rule applies when failure of
| both DG-A and DG-B is not in
| the cut set.
if ~(ALL-DGS) then
partition = "???-TRS";
endif
```

15.5 Partition Rule Example

15.5.1 Partition Rules - Example

PURPOSE

Provide an example, along with the steps for using partitioning rules.

STEPS

1. Create project rules
2. Create individual sequence rules
3. Apply the rules
4. Gather the end states

End state results can be displayed and/or reported from either the **End State | View** or **Report | End State** options.

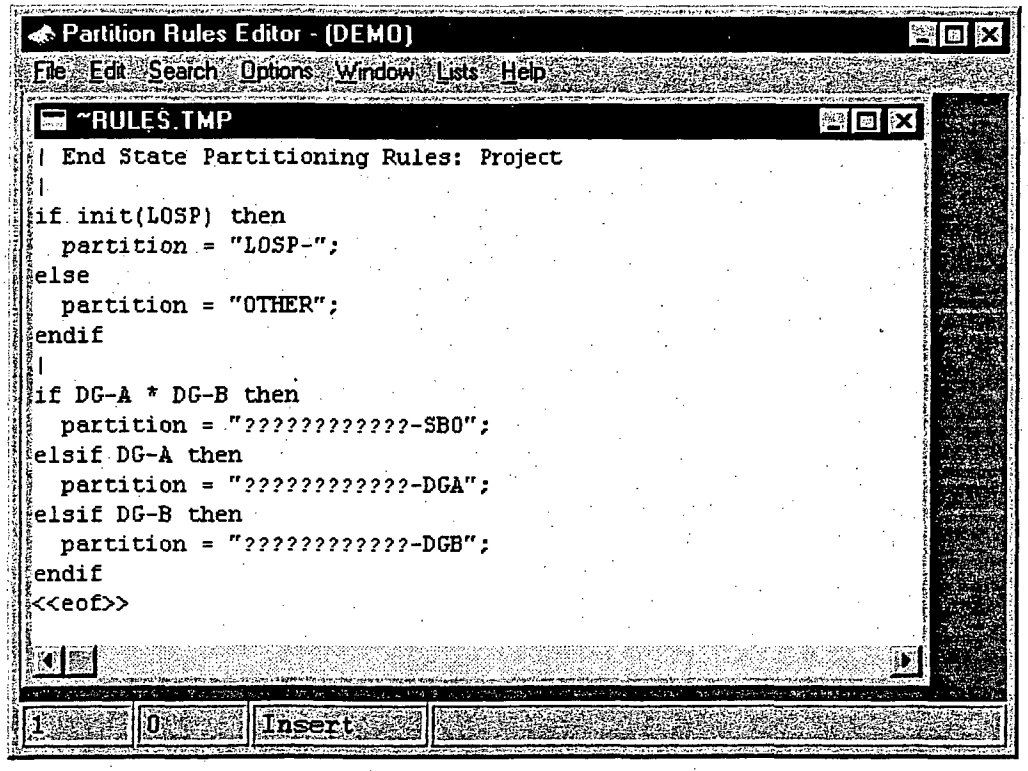
15.5.2 Project Rules

PURPOSE

Create partition rules for the DEMO project.

STEPS

1. Select **Sequence** from the SAPHIRE menu. The *Sequences* dialog will be displayed.
2. Right-click to invoke the pop-up menu and select **Cut Sets | Partition | Edit Rules.**
3. The *Edit Partition Rules* dialog will be displayed. Select *Rule Level* **Project**, *Rule Type* **Basic**, and choose the **OK** button.
4. The *Partition Rules Editor* window will be displayed. Because these are project rules, only the project name appears in the title bar.



5. Enter the partition rules as displayed. Select **File | Exit** when complete.

The first project rule shown adds characters 1 through 5 as "LOSP-" when LOSP is the initiating event, and "OTHER" if LOSP is not the initiating event.

The next rule adds characters 13 through 16 as "-SBO" when both DG-A and DG-B are failed. When DG-A is failed, "DGA" is added, and when DG-B is failed, "DGB" is added (as characters in the end state name).

15.5.3 Sequence Rules

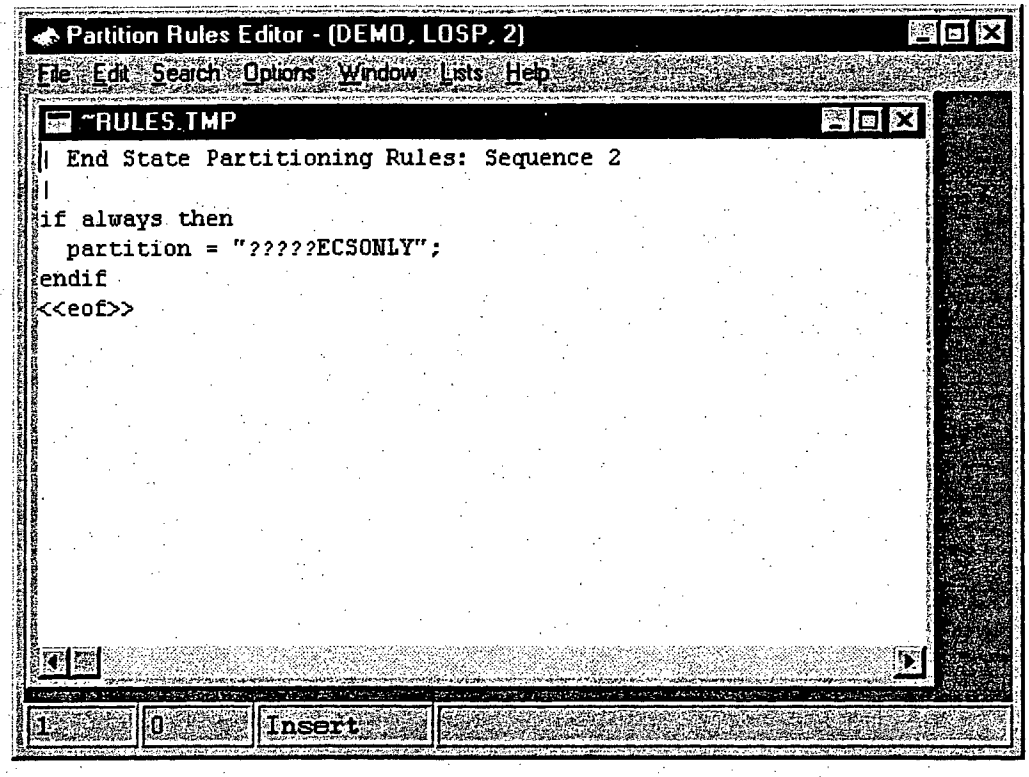
PURPOSE

Create sequence partition rules for each of the sequences in the DEMO project LOSP event tree.

STEPS

1. Select **Sequence** from the SAPHIRE menu. The *Sequences* dialog will be displayed.

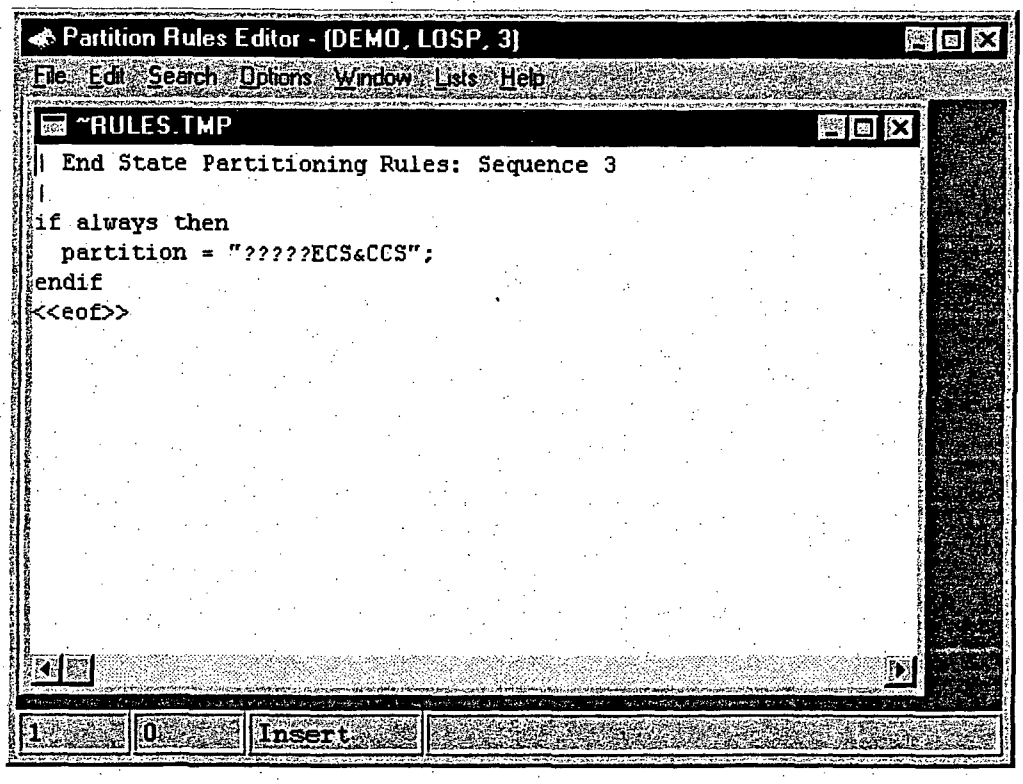
2. Highlight the LOSP 2 sequence.
3. Right-click to invoke the pop-up menu and select **Cut Sets | Partition | Edit Rules**.
4. The *Edit Partition Rules* dialog will be displayed. Select *Rule Level* **Sequence**, *Rule Type* **Basic**, and choose the **OK** button.
5. The Edit Partition Rules dialog will be displayed.
6. The *Partition Rules Editor* window will be displayed. The event tree and sequence name, along with the current project name will appear in the title bar when an individual sequence is selected.



7. Enter the partition rules as displayed. Select **File | Exit** when complete.

The rule in sequence 2 adds characters 2 through 12 as "ECSONLY".

Repeat the process above for the LOSP 3 sequence, beginning with step 2.



The rule in sequence 3 adds characters 6 through 12 as "ECS&CCS".

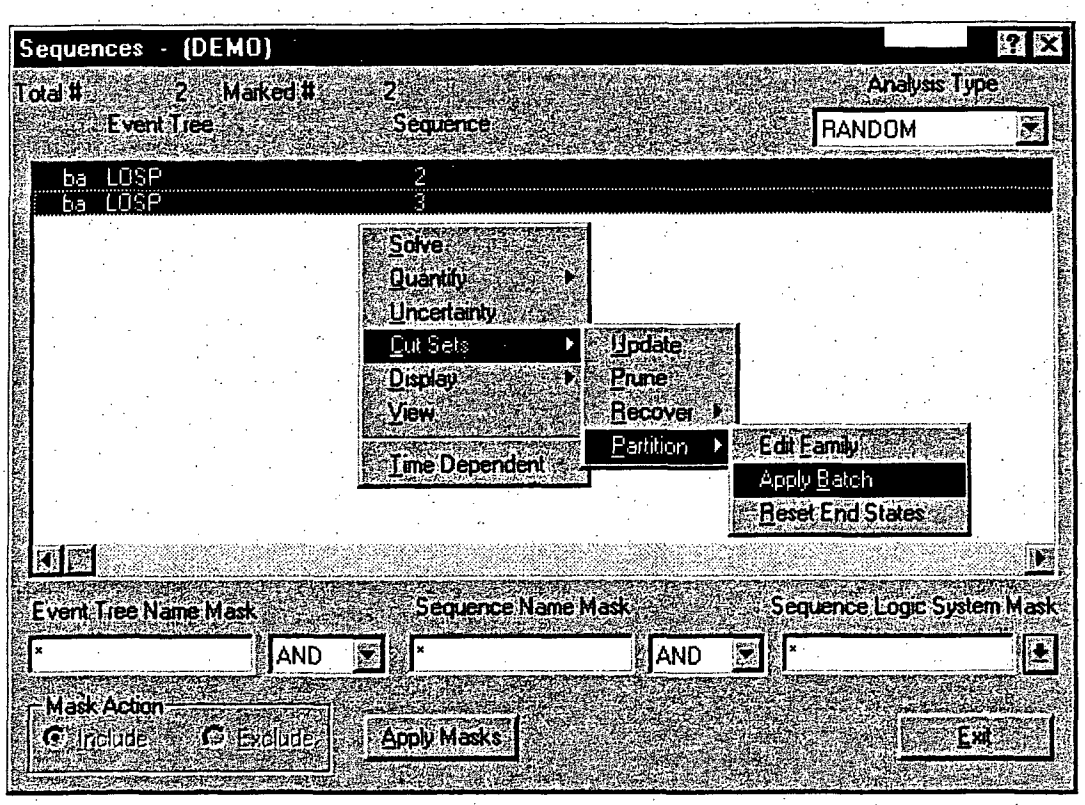
15.5.4 Applying Partitioning Rules

PURPOSE

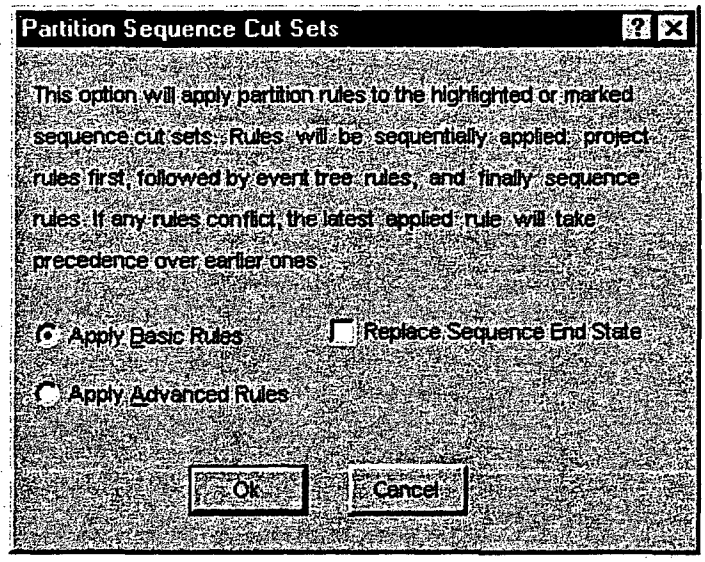
Apply the partition rules after entering them via the *Partition Rules Editor*. After creating the partition rules, they must now be applied.

STEPS

1. From the *Sequences* dialog, highlight the LOSP sequences.
2. Right-click to invoke the pop-up menu and select **Cut Sets | Partition | Apply Batch**.



3. A warning dialog will be displayed. Choose **Ok** to continue.



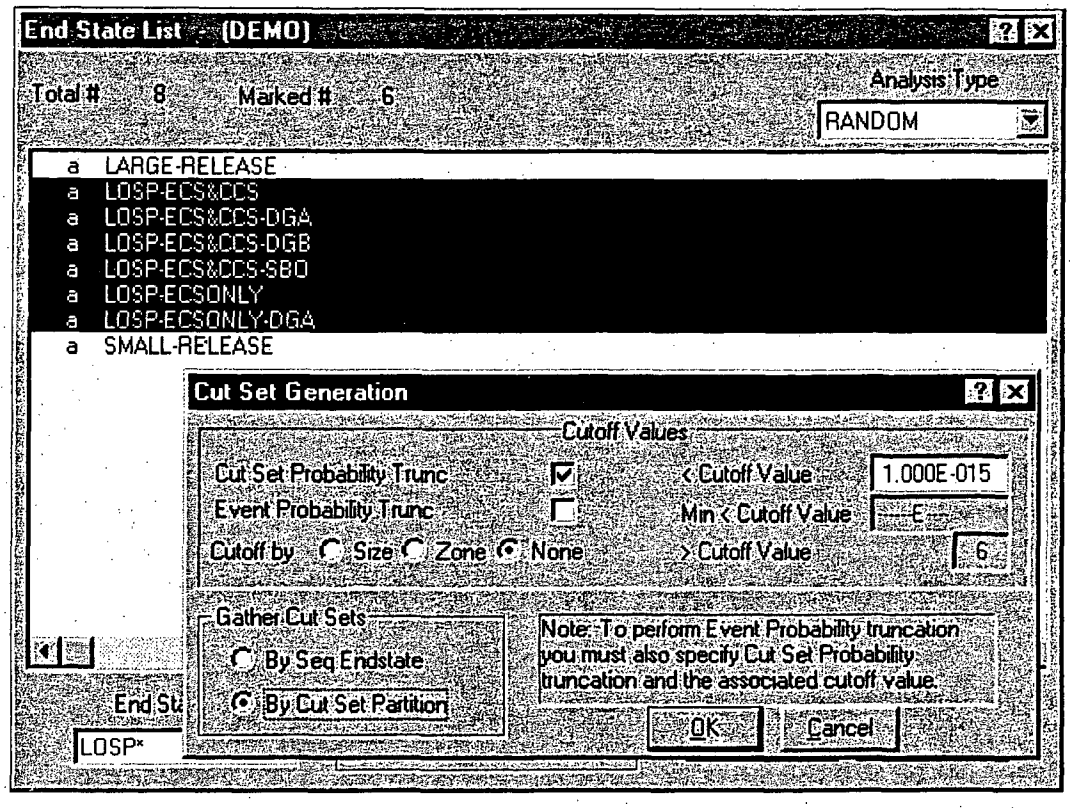
15.5.5 Gathering the End States

PURPOSE

Gather the cut sets for the end states created by the partition rules. After applying the partition rules the end states must now be "gathered".

STEPS

1. From the SAPHIRE menu select **End State**. The *End State List* dialog will be displayed.
2. Highlight the end states created by the partition rules.
3. Right-click to invoke the pop-up menu and choose **Gather**. The *Cut Set Generation* dialog will be displayed.



You may change any of the data on this dialog. For more information about the fields in the graphic above, click on the desired field.

In order to gather the partitioned cut sets, from the *Gather Cut Sets* section you must select the **By Cut Set Partition** radio button.

16. LARGE EVENT TREE METHODOLOGY

16.1 Application of the Large Event Tree Methodology

PURPOSE

Describe the "large event tree" methodology and how SAPHIRE can be used to evaluate sequences using this approach. The options that allow truncation of sequences during the process of linking event tree sequences and other options related to analyzing large event trees are presented.

TOPICS

Large Event Tree Methodology
Connecting Support System to Plant Response Trees
Example Event Trees
Using Link Event Tree Rules to Assign Split-fractions
Using the "S" Calculation Type
Truncating Sequences During Linking
Using Process Flags
Obtaining Cut Sets in Terms of Basic Events

16.2 Large Event Tree Methodology

There are two basic approaches for accident sequence quantification:

1. Fault tree linking
2. Large event tree methodology (also called "event trees with boundary conditions")

Characteristics of the large event tree methodology include:

1. Important support systems are modeled as top events in the event trees rather than being contained in the "front line system" or "plant response system" fault trees.

   This type of modeling removes shared dependencies from the plant response system fault trees.

2. The paths through the event tree (i.e., sequences) can be quantified by multiplying the split-franctions along the path because the top events are independent.

   This multiplication is in contrast to the fault tree linking approach, where simply multiplying the branch probabilities together would probably yield incorrect results because of the potential for double-counting component failures unless Boolean processes are applied.

3. The split-fraction for each branch point in the model is derived from a fault tree that applies to the branch point. The successes and failures on the path leading to that branch point (which define the

"boundary conditions" for the fault tree) must be recognized when the fault tree is developed and solved. The resulting "split-fraction" is conditional upon the path through the event tree.

This is in contrast to the fault tree linking approach which usually has only one fault tree that corresponds to a particular top event.
The split-fractions underneath the top event are assigned by using the "Link Event Tree" rules to specify the particular fault tree that corresponds to the branch point.

4. Each path through the event tree (i.e., sequence) is characterized by the initiating event and by the combination of failed and successful fault trees in the path. Success branch probabilities are retained along with the failed branch probabilities for the sequence.

   In SAPHIRE, the sequence is stored as a "cut set" even though the term "cut set" implies retaining only the failed branch probabilities.

There are important differences that arise between the fault tree linking approach and the large event tree approach. How the user must construct the model and choose analysis options for the large event tree approach in SAPHIRE are summarized below:

- The sharing of support system event trees with different plant response event trees, depending on the initiating event.
- The use of the Link Event Tree rules to assign split-fractions
- The use of multiple-split branching in the event tree, such as 3-split or 4-split branching.
- The need to use truncation when generating sequences because of the large number of sequences that could be generated. (Currently, a SAPHIRE database cannot retain more than 100,000 sequences.)
- The analysis options and use of Process Flags used to retain success events in the cut sets.

16.3 Connecting Support System Event Trees to Plant Response Event Trees

A "support system event tree" (that represents power systems, instrument air, etc.) may be used by several initiating events.

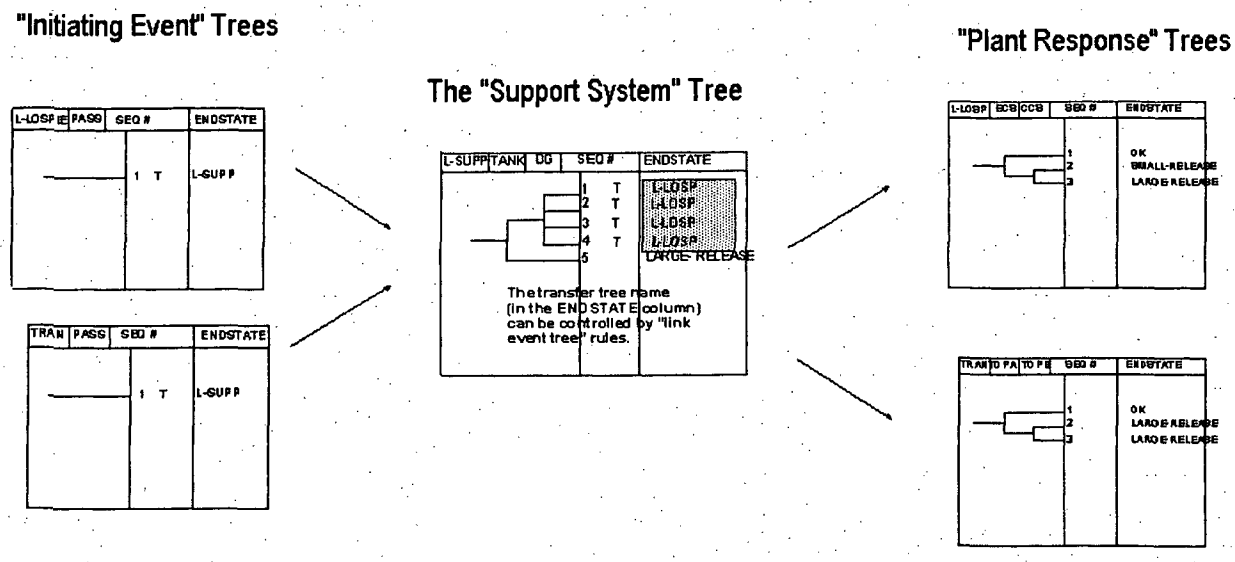
The "support system event tree" needs to transfer to the appropriate plant response event tree, depending on the initiating event.

To connect the event trees (and avoid having to duplicate event trees), the following approach is preferred:

1. Create an event tree that will contain the initiating event and will transfer to the appropriate support system event tree. (Note: SAPHIRE requires that at least two tops be present in each event tree; however, there does not need to be any branching.)

2. The path through the support system event trees (which may contain many transfers to include all of the support system event trees) will ultimately result in the need to transfer to the appropriate plant response event tree.
3. Rules can be written in the "Link Event Tree" rule editor to enact this transfer.

Notice that more than one initiating event could call the same support system tree. And the support system tree could transfer to different plant response trees.



16.4 Using "Link Event Tree" Rules to Assign Split-Fractions

In the large event tree approach, the particular path through the event tree determines the status of support systems and therefore, determines the appropriate fault tree for plant response systems that have dependencies on the support systems. There may be several variations of the basic fault tree used underneath each event tree top event.

In contrast, with the fault tree linking approach, each top event usually corresponds to a single fault tree or single top event probability for the failed branch.

Plant Response Tree Rules

To assign the proper split-fractions, the rules shown here must be applied during sequence generation. (Additionally, the sample support system rules must also be applied.)

```
| Plant response system rules
|       THE ECS RULE WAS WRITTEN IN TERMS OF
|       BRANCH IDENTIFIERS
if /DG then
  /ECS = ECS-0;
   ECS = ECS-0;
elsif DG[1] then
  /ECS = ECS-B;
   ECS = ECS-B;
elsif DG[2] then
  /ECS = ECS-A;
   ECS = ECS-A;
else
  /ECS = ECS-AB;
   ECS = ECS-AB;
endif
|       THE CCS RULE WAS WRITTEN IN TERMS OF
|       SUBSTITUTED NAMES
if /DG0 then
  /CCS = CCS-0;
   CCS = CCS-0;
elsif DG1 then
  /CCS = CCS-B;
   CCS = CCS-B;
elsif DG2 then
  /CCS = CCS-A;
   CCS = CCS-A;
else
  /CCS = CCS-AB;
   CCS = CCS-AB;
endif
```

Notice that the rules can be written in terms of the *branch identifier* (e.g., DG[1]) or in terms of the *fault tree name* (e.g., DG1).

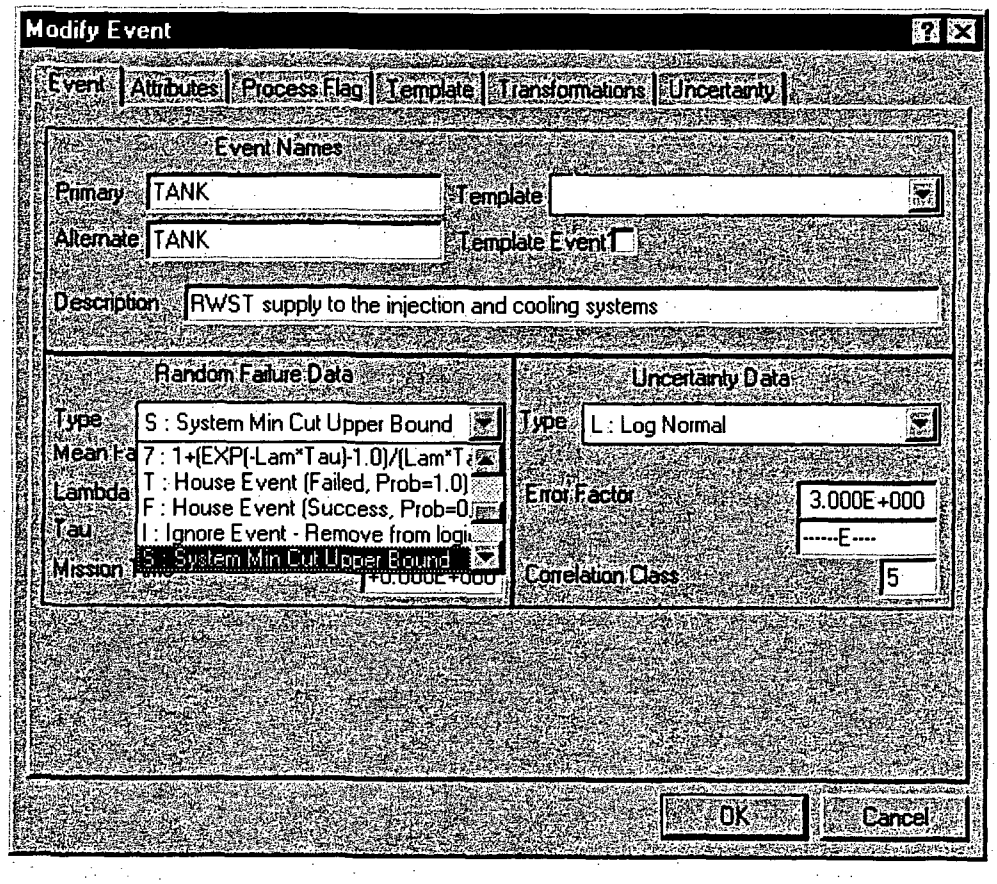
The fault tree name must have been assigned by earlier rules, usually, the support system rules.

SEE ALSO

Link Event Tree rule language and rule editor.

16.5 Using the "S" Calculation Type

The "S" calculation type is specified in the **Modify | Basic Events** option.



The "S" calculation type allows the split-fraction to be set to the previously calculated minimal cut set upper bound for the fault tree. Using this feature saves the user from having to manually type in the split-fraction probability.

When the "S" calculation type is used, the proper steps must be taken to load the minimal cut set upper bound value into the current database. The steps are as follows:

1. Enter the fault tree logic and basic event data associated with the fault tree.
2. Specify the S calculation type for the fault tree or "split-fraction" name which corresponds to the fault tree top gate name.
3. Perform the **Generate** function from the **Generate** menu option. This must be performed ***prior*** to generating the fault tree cut sets so that the new basic event data is in the "current" data base.
4. Analyze the fault tree using the **Fault Tree** option from the SAPHIRE menu. (For a new fault tree, you would begin with the **Solve** option.)
5. Repeat the **Generate** option, as in Step 3. This step must be performed ***prior*** to generating sequences with truncation. You can use the **Report | Basic Event** option to report current fault tree probabilities.

16.6 Truncating Sequences During Event Tree Linking

Prerequisites for Generating Sequences with Truncation

The fault trees that do not have logic should have a failure probability specified prior to generating (and truncating) sequences. This is entered via the **Modify | Basic Events** option.

The fault trees that have logic should also have a failure probability specified prior to generating (and truncating) sequences. This is also entered in **the Modify | Basic Events** option either by directly specifying a probability or by using the "S" calculation type.

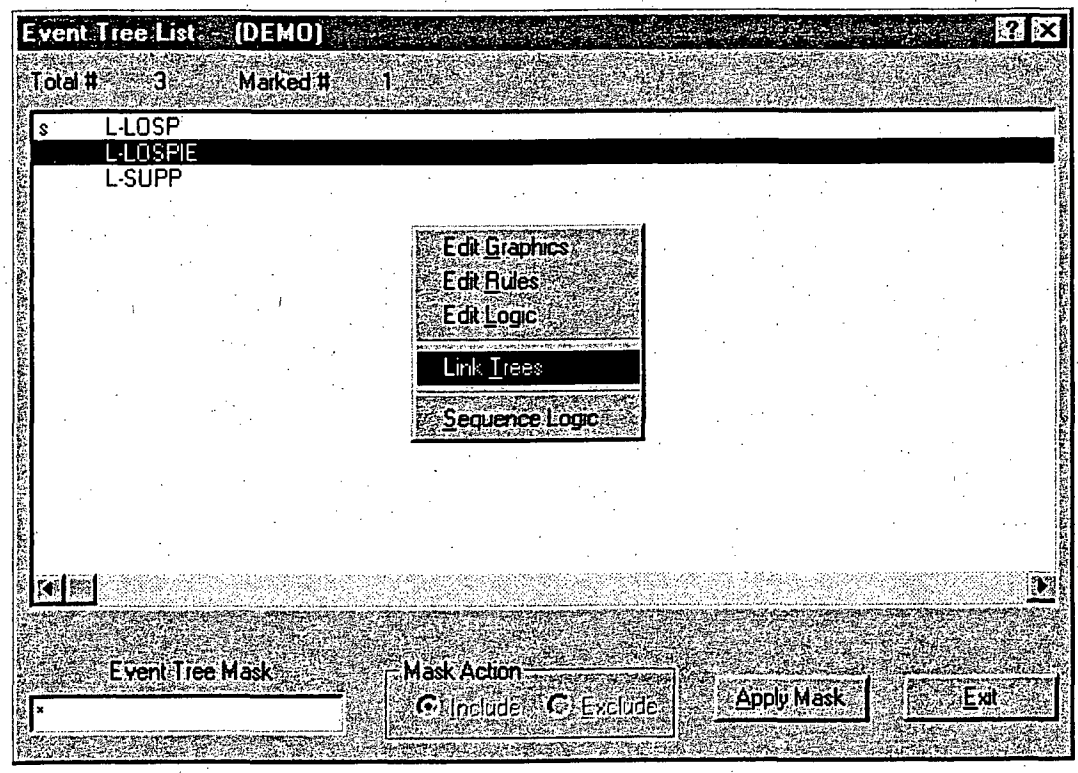
Entering Fault tree Split-Fractions

Fault trees, whether they have logic or not, are listed both as *Fault Trees* and as *Basic Events* in the SAPHIRE database. When a *Fault Tree* is entered into the database, it is automatically added to the *Basic Event* list.

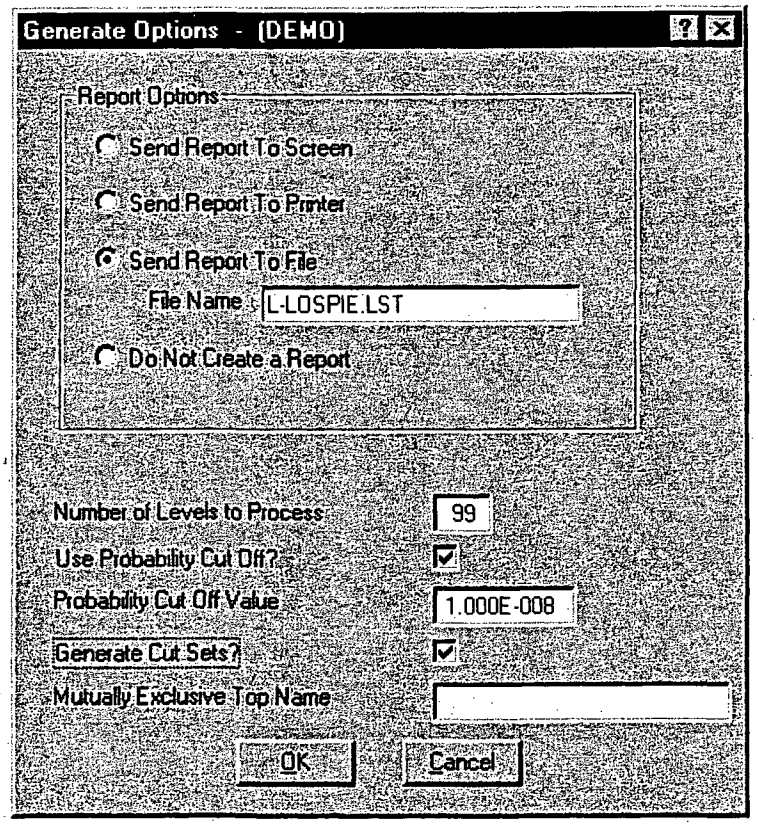
The fault tree failure probability provides the split-fraction for the failed branches and its complement provides the split-fraction for the success branches.

In this example, the L-LOSPIE event tree will transfer to the L-SUPP tree, which transfers to the L-LOSP tree. To generate all of the desired sequences:

1. Select Event Tree from the SAPHIRE menu.
2. The Event Tree List dialog will be displayed.
3. Highlight **only the L-LOSPIE tree**. Selecting the L-SUPP or L-LOSP trees would generate unwanted sequences in this example case.



4. Right-click to invoke the pop-up menu and select the **Link Trees** option.
5. The *Generate Options* dialog will be displayed.



When using the large event tree approach, perform truncation when generating sequences by selecting the following options:

1. Select the **Use Probability Cut Off?** check box, and enter the cutoff value. 1.0E-08 was used here.

2. Select the **Generate Cut Set?** check box.

Using these options will eliminate any sequences that are below the specified truncation value. The calculated sequence frequencies are based on the fault tree failure probabilities entered in the **Modify | Basic** Events option or entered via a change set.

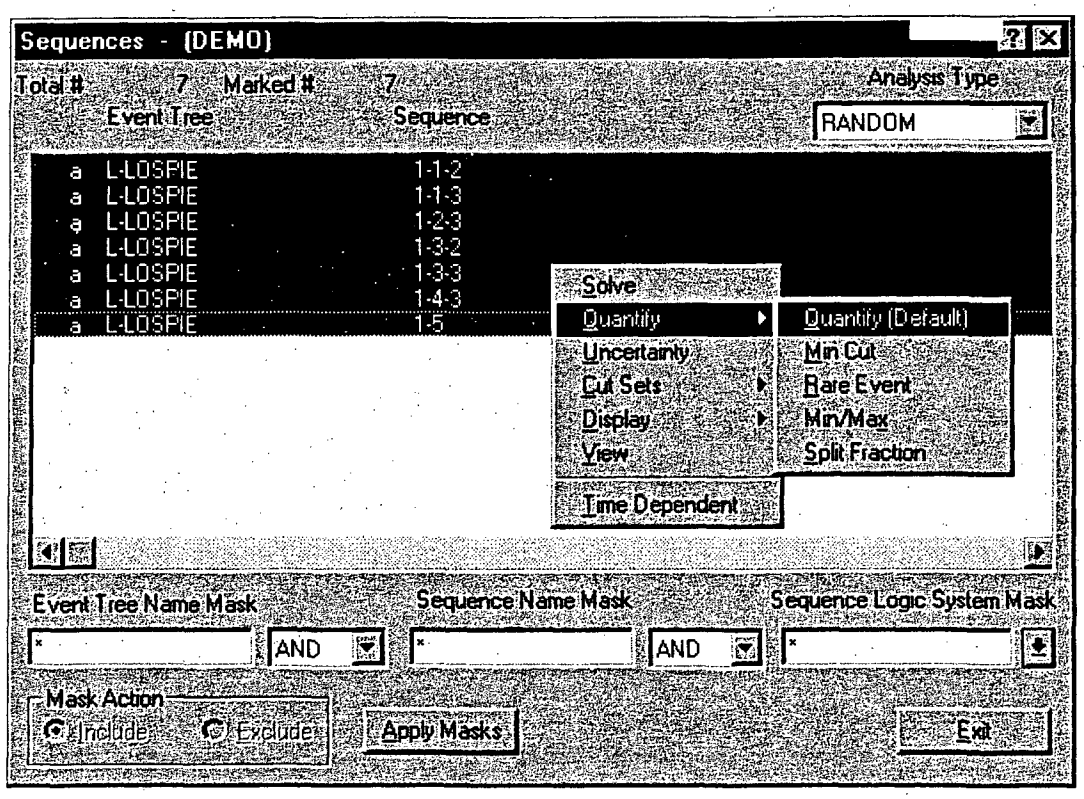
NOTE: During the sequence truncation process, fault trees are not evaluated to obtain the sequence frequency.

The sequences generated by this tree are shown in the table.

Sequence Name	System	System	System
Event Tree Name: L-LOSPIE			
Transferring to event tree : L-SUPP			
1-5	TANK		
Transferring to event tree : L-LOSP			
1-4-3	/TANK	DG[3]	ECS
substitutes		DG3	ECS-AB
	CCS		
substitutes	CCS-AB		
Transferring to event tree : L-LOSP			
1-3-3	/TANK	DG[2]	ECS
substitutes		DG2	ECS-A
	CCS		
substitutes	CCS-A		
1-3-2	/TANK	DG[2]	ECS
substitutes		DG2	ECS-A
	/CCS		
substitutes	/CCS-A		
Transferring to event tree : L-LOSP			
1-2-3	/TANK	DG	ECS
substitutes		DG1	ECS-B
	CCS		
substitutes	CCS-B		
Transferring to event tree : L-LOSP			
1-1-3	/TANK	/DG	ECS
substitutes		/DG0	ECS-0
	CCS		
substitutes	CCS-0		
1-1-2	/TANK	/DG	ECS
substitutes		/DG0	ECS-0
	/CCS		
substitutes	/CCS-0		
Saved Sequences: 7 Valid Sequences: 7 Processed: 7			

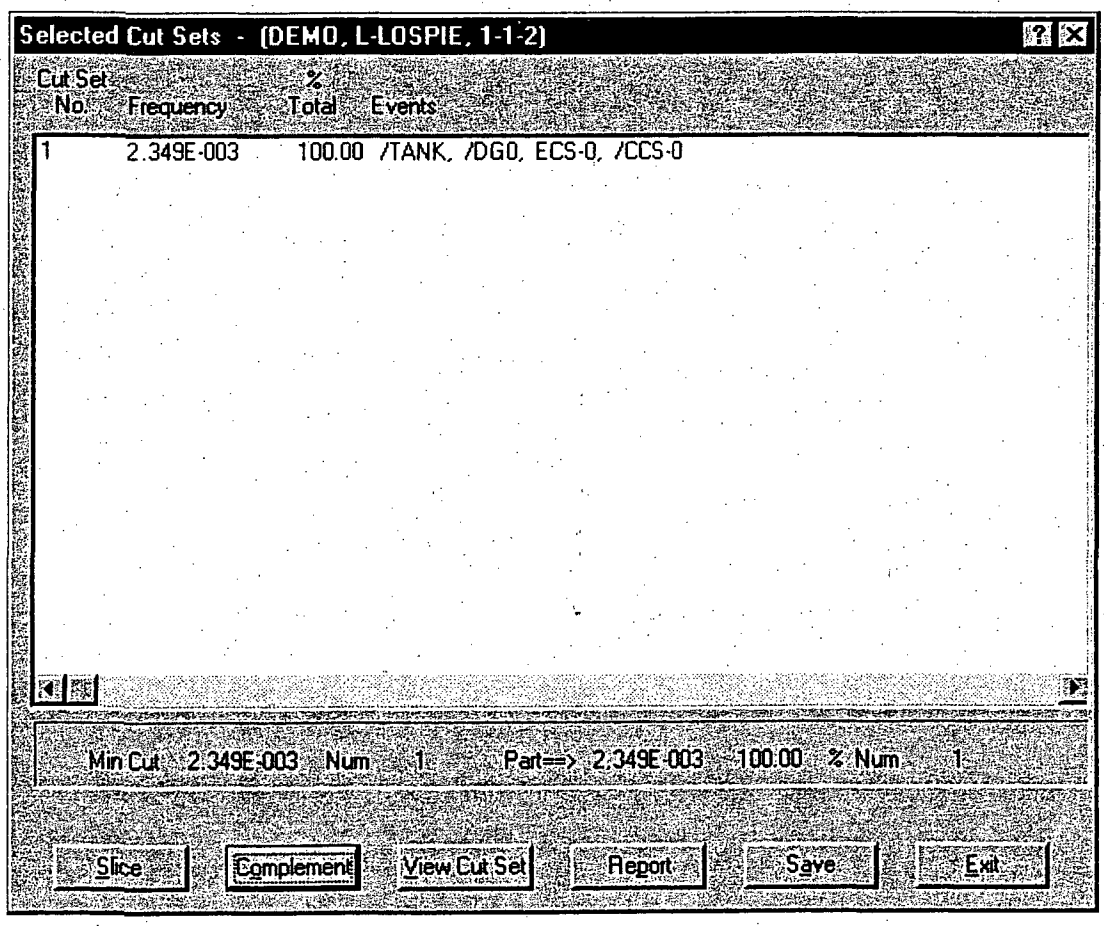
The file that the generated sequences are listed in can be named from the "File Name" field of the *Generate Options* dialog. If many sequences are generated, this file can become large – slowing down the sequence generation process and taking up disk space. It may be desirable to prevent this file from being created by selecting the **Do Not Create a Report** radio button.

3. After linking the event trees, you will need to quantify the sequences prior to reporting sequence results. Select **Sequence** from the SAPHIRE menu.
4. Highlight the L-LOSPIE sequences and right-click to invoke the pop-up menu.



5. Choose **Quantify | Default**.

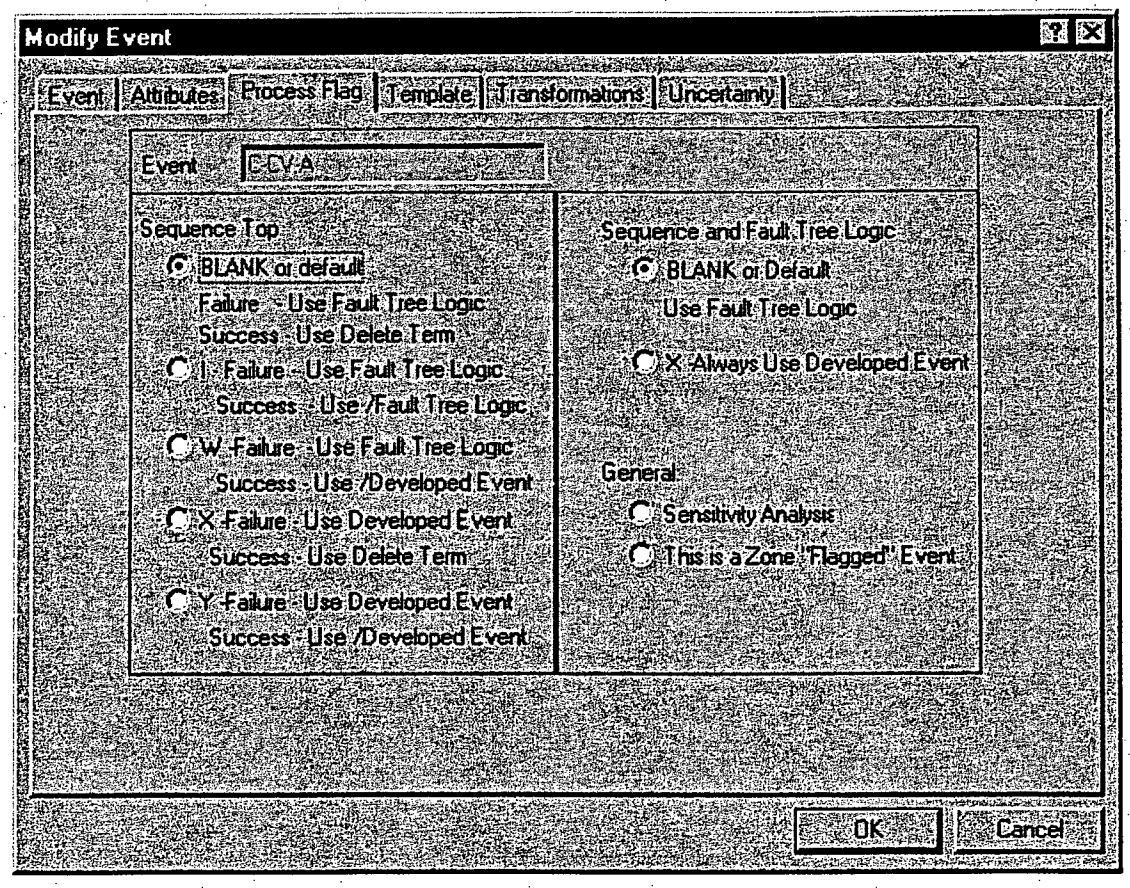
6. The sequence results can be viewed by highlighting the desired sequence(s) and the right-clicking and choosing **Display | Cut Sets** from the pop-up menu. The success and failure split-fractions are displayed in the *Selected Cut Sets* dialog.



Notice that each sequence will contain only one cut set which represents the split-fraction along the event tree path.

16.7 Using Process Flags to Retain Success Events

Process Flags are needed to specify the proper treatment of success events if cut sets are to be generated *in terms of basic events* following sequence generation. They are specified in the **Modify | Basic Events** option. Choose the down arrow adjacent to the *Process Flag* field to invoke the *Process Flag* dialog.



The Y Process Flag is used when the top event does not have fault tree logic (or you chose to ignore the fault tree logic).

The W Process Flag is used when the top event has fault tree logic that is to be used for the failed branch, but the split-fraction is to be used for the success branch.

The Process Flags and Calculation Types specified in this example are shown below:

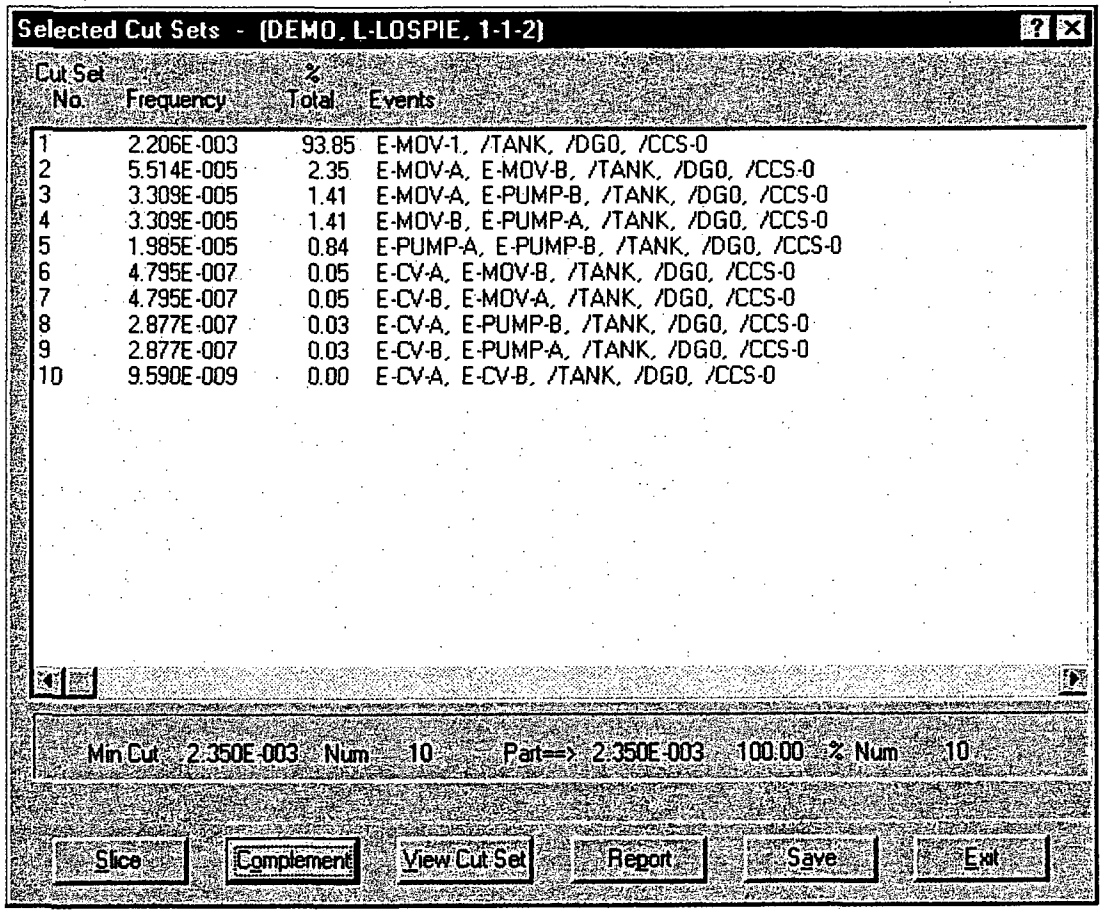
Name	Calc Type	Process Flag	Comment
CCS-0	S	W	System calls a fault tree
CCS-A	S	W	System calls a fault tree
CCS-AB	1	Y	System uses a probability
CCS-B	1	Y	System uses a probability
DG0	S	Y	System calls a fault tree, but only the success split-fraction is needed
DG1	S		System calls a fault tree that is only used for failed branches, not for success branches
DG2	S		System calls a fault tree that is only used for failed branches, not for success branches
DG3	S		System calls a fault tree that is only used for failed branches, not for success branches
ECS-0	S	W	System calls a fault tree

ECS-A	1	Y	System uses a probability
ECS-AB	1	Y	System uses a probability
ECS-B	S	W	System calls a fault tree
TANK	1	Y	Top event uses a probabliity

16.8 Obtaining Cut Sets in Terms of Basic Events

To obtain cut sets in terms of basic events from the fault tree rather than in terms of fault tree split-fractions, use the **Sequence | Solve** option. You need to have generated the sequences before you use this option.

The cut sets can be viewed from the **Sequence | Display | Cut Sets** option. In this example, the success and failures in terms of basic events are displayed.



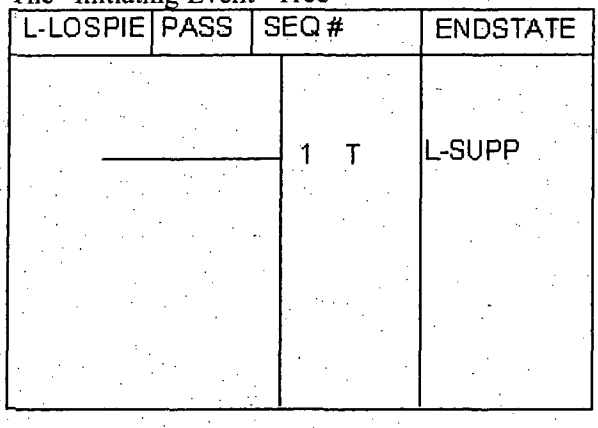
NOTE: To obtain cut sets in terms of basic events, the fault trees for every split-fraction that is affected by a basic event in a fault tree must be in the database.

16.9 Event Trees Used in Examples

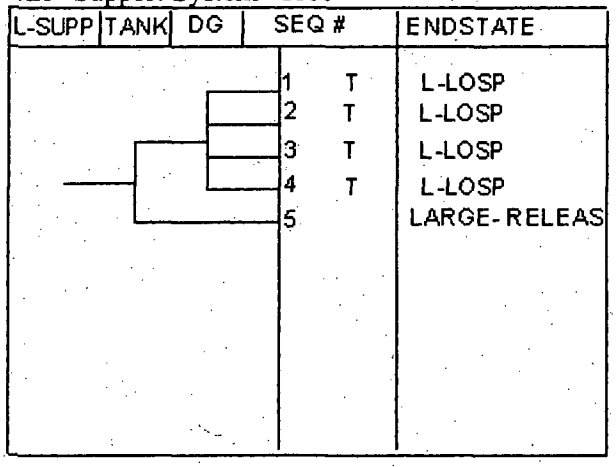
16.9.1 Example Event Trees

These three event trees will be used as examples in this section:

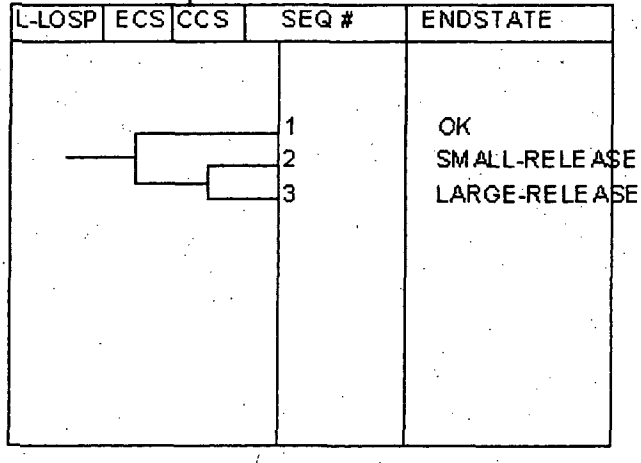
The "Initiating Event" Tree



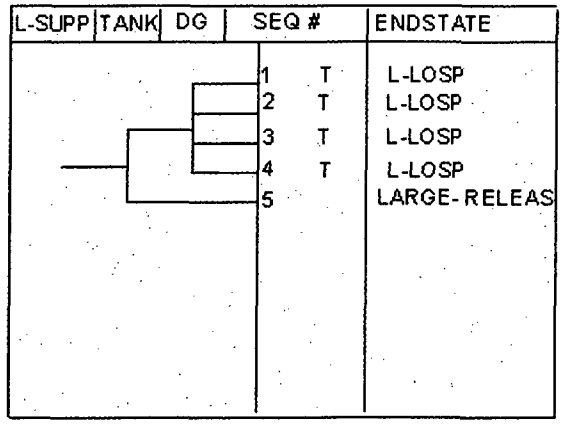
The "Support System" Tree



The "Plant Response" Tree



16.9.2 The Support System Tree



In this example, top event TANK questions the status of the tank, and if the tank is failed, both ECS and CCS fail.

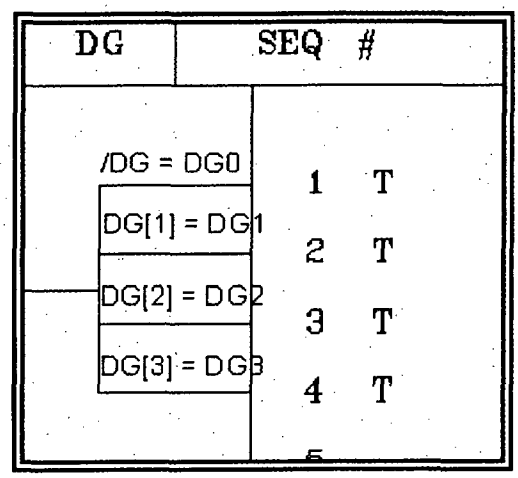
The probability of failure is specified directly, without using a fault tree. The probability of failure is entered via the Modify | Basic Events menu. SAPHIRE determines the success probability as the complement of the failure probability.

/TANK = 1- TANK

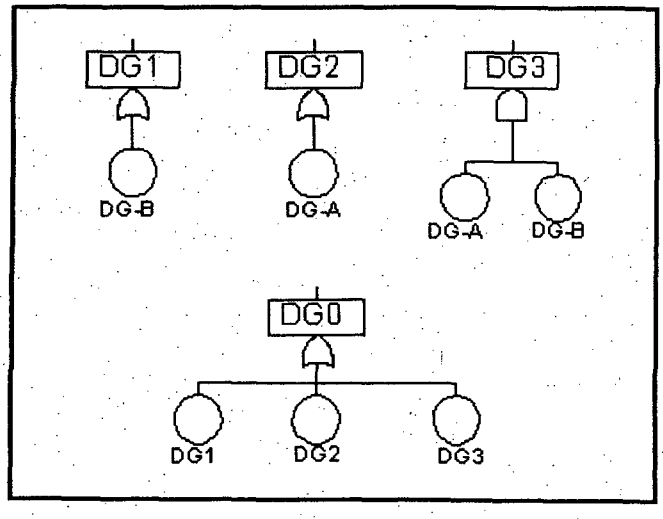
Top event DG questions the status of two diesel generators that provide support to ECS and CCS.

The fault tree for each diesel generator is assumed to consist of a single basic event. Therefore, failure of diesel generator A is modeled by a fault tree consisting of the basic event DG-A, and failure of diesel generator B is modeled by a basic event DG-B.

The top branch (/DG) represents success of both diesel generators. The next branch down represents success of diesel generator A and failure of diesel generator B. Thus, DG1 is a fault tree containing DG-B. The third branch represents success of diesel generator B and failure of diesel generator A. Thus, DG2 is a fault tree containing DG-A. The bottom branch represents both diesel generators being failed.



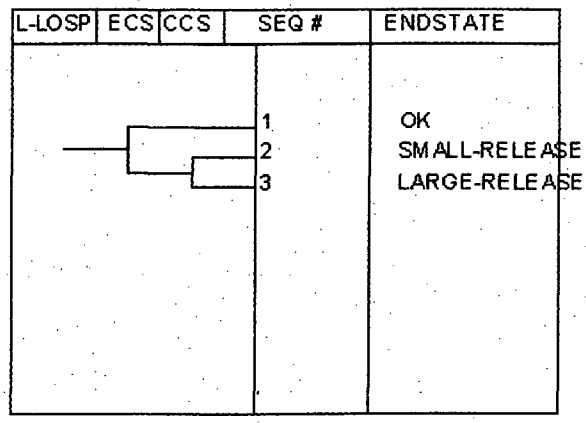
The split-fractions for each branch are determined from the corresponding fault tree. The split-fraction for /DG is taken as the complement of the DG0 fault tree.



The "link event tree" rule editor is used to assign the appropriate "system" to each branch. The rules for the support system event tree are as shown:

```
| Support system rules
|
if always then
  /DG    = DG0;
  DG[1] = DG1;
  DG[2] = DG2;
  DG[3] = DG3;
endif
```

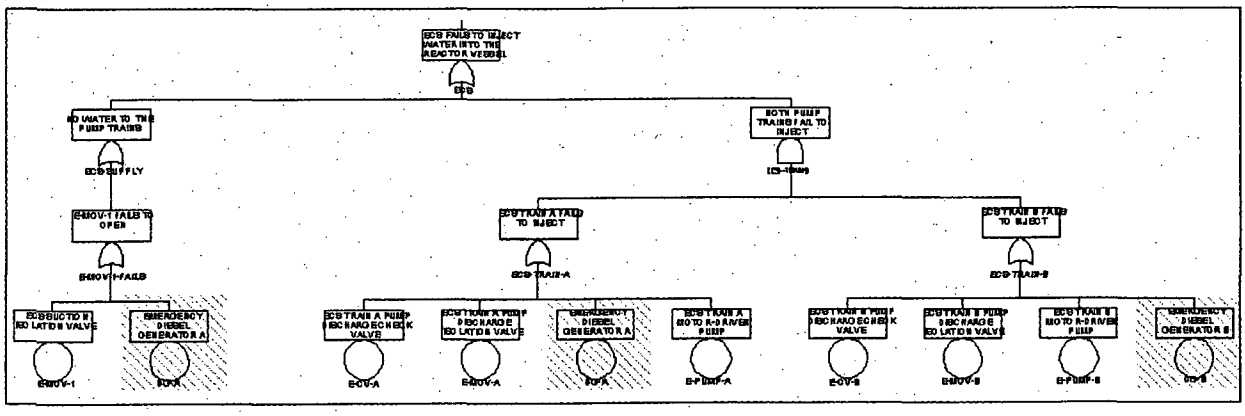
16.9.3 The Plant Response Tree



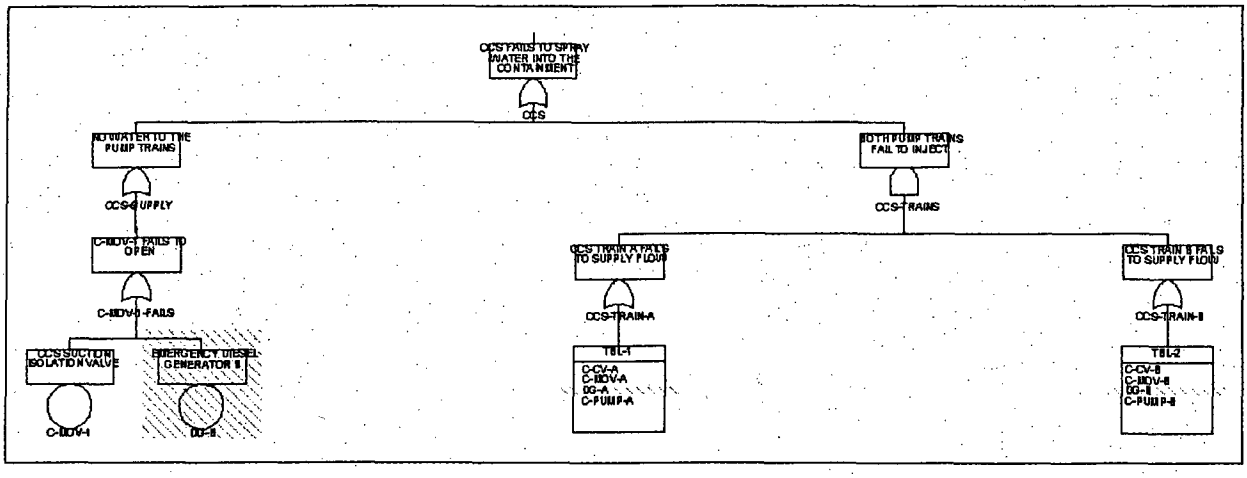
The ECS and CCS top events are similar to the fault tree linking example in the DEMO project, with the following exceptions:

- The TANK failure event is no longer contained in the fault trees for ECS and CCS.
- The status of the DGs depends on the path through the support system event tree.

When the top branch of the support system tree transfers to the plant response tree (PRT), both DGs are successful. When the second branch of the support system tree transfers to the PRT, DG A is successful but DG B is failed, and so on. As you can see, the probability of ECS (and CCS) failing depends on the path through the system support event tree. A special version of the ECS (and CCS) fault tree is created for each situation. The "Link Event Tree" rules are used to assign the fault tree name which corresponds to the proper fault tree based on the status of the DGs. The success or failure probability for each fault tree (the split-fractions) can be derived from each fault tree prior to generating sequences (via the "S" calculation type) or can be directly specified by the user as a probability.



DG Status	ECS Fault Tree Name	Split-Fraction
No DGs failed	ECS-0	1.065E-3
Only DG-A failed	ECS-A (FAILED)	1.0
Only DG-B failed	ECS-B	9.076E-3
Both DGs failed	ECS-AB (FAILED)	1.0



DG Status	CCS Fault Tree Name	Split-Fraction
No DGs failed	CCS-0	1.065E-3
Only DG-A failed	CCS-A	9.076E-3
Only DG-B failed	CCS-B (FAILED)	1.0
Both DGs failed	CCS-AB (FAILED)	1.0

17. LINKAGE RULES

17.1 Linkage Rules Usage and Examples

PURPOSE

The linkage rule editor allows you to create rules that affect sequence generation. Typically, these rules are used to replace default fault trees with substituted fault trees (or "split-fractions") based on logical conditions that are specified in the rules.

These rules are applied when sequence logic is generated. Without using rules, the sequence logic consists of the logical combinations of failed and successful top events (and the initiating event for the event tree). These rules allow the user to replace the top event (which is called a "fault tree" in SAPHIRE) with a substituted fault tree based on the logical conditions dictated by the rule.

TOPICS

Search Criteria Examples
Symbols
Keywords and Structures
Binary and Multiple-Split Branches

17.2 Search Criteria Examples

These examples are based upon an event tree with initiating event IE-1 and top events A, B, and C.

Search Criteria	Meaning of the Search Criteria	Search Criteria	Meaning of the Search Criteria
init(IE-1)	Initiating event with the name IE-1	~A	Failure of A never occurs
A	Failure of top event or fault tree A	~(/A + A)	Success of A and failure of A never occurs (can be used to test for a "pass" condition)
/A	Success of top event or fault tree A	A * B	Failure of A and of B occur
(A + B) * C	Failure of A or B occurs and failure of C occurs	always	This pre-defined macro name means the criteria is always met.

17.3 Rule Editor Symbols

| (comment)	Denotes a comment line
* (AND)	Logical AND operator
+ (OR)	Logical OR operator
~ (never)	Logical operator for "never" or "not present"
/ (complement)	Complement
; (end macro)	Indicates the end of a macro line or a line that modifies the criteria being evaluated.
( )	Parentheses for grouping terms

17.4 Keywords and Structures

17.4.1 Linkage Rules Keywords and Structure

if-then - Indicates that a search criteria is being specified.
if-then using wildcards - A search criteria can be specified using wildcards.
always - Indicates that the search criteria is always satisfied.
elsif - Specifies an alternative search criteria.
else - Specifies some action to be taken if all the search criteria are not met.
Cast Operators - Define substitutions for the end state or flag set of a sequence, or define a new transfer event tree
Skip - Indicates that the specified item should be ignored.
Macro - A user-definable keyword that specifies a search criteria.

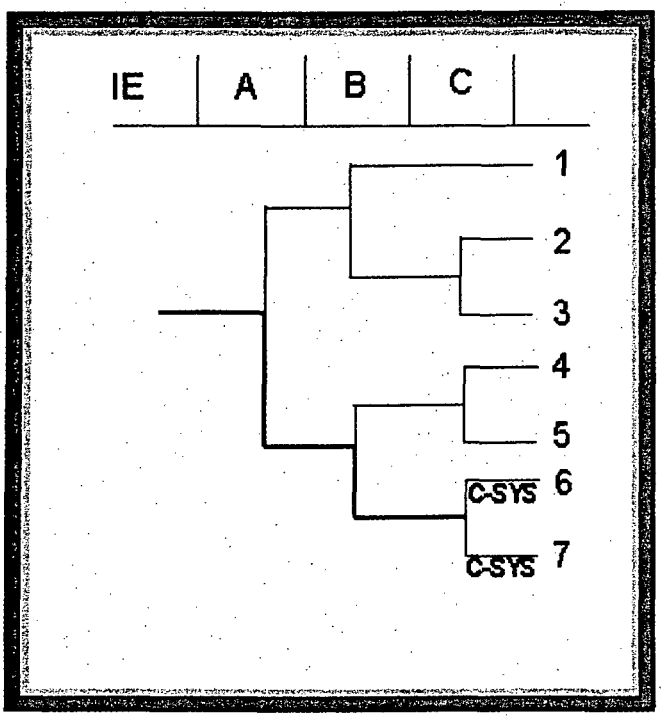
17.4.2 if-then

This keyword indicates that a search criteria is being specified. The information between the "if" and the "endif" clause defines the tops to be replaced. There is no limit on the number of replacements that can be defined between these clauses.

```
| The "if-then" rule structure:
|  This rule replaces C with C-SYS when
| A and B are both failed.
| Only sequences 6 and 7 are affected by this rule.
if A * B then
/C = C-SYS;

C = C-SYS;

endif
```



Important Reminders:

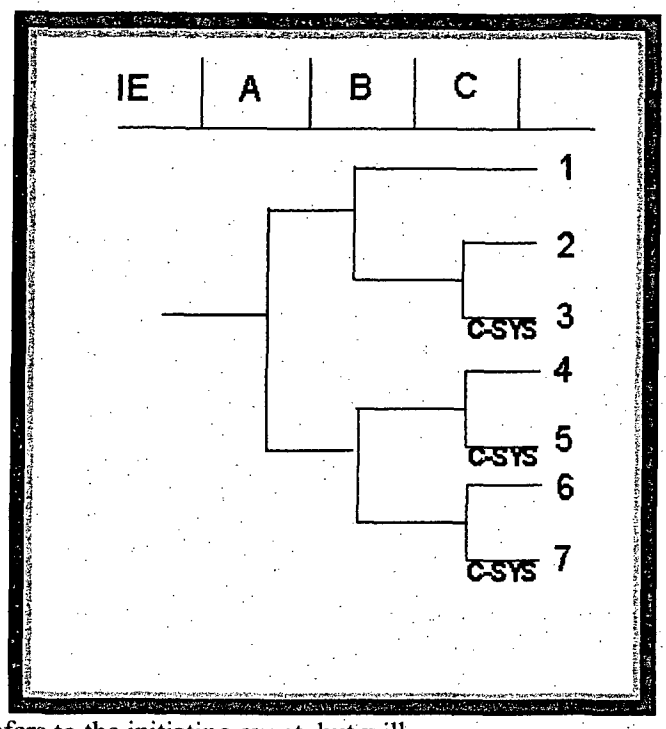
Each replacement line must end with a semicolon.
You can add as many replacement lines in the rule as you want.
The rules are sensitive to UPPER and lower case - the cOrReCtcASe mUSt bE uSeD!

17.4.3 if-then Using Wildcards

Wildcard characters can be used with the if-then structure to indicate a search for a group of criteria whose names have something in common. A wildcard is a character that can represent one or more characters in a search criteria. You can use one or more wildcards to specify groups of search criteria. The wildcard characters must be enclosed in quotation marks.

```
| The "if-then" rule structure using wildcards:
| This rule replaces C with C-SYS when
| the initiating event occurs.
|
| (Sequence 3, 5 and 7 are affected)
if "??" then
C = C-SYS;

endif
```



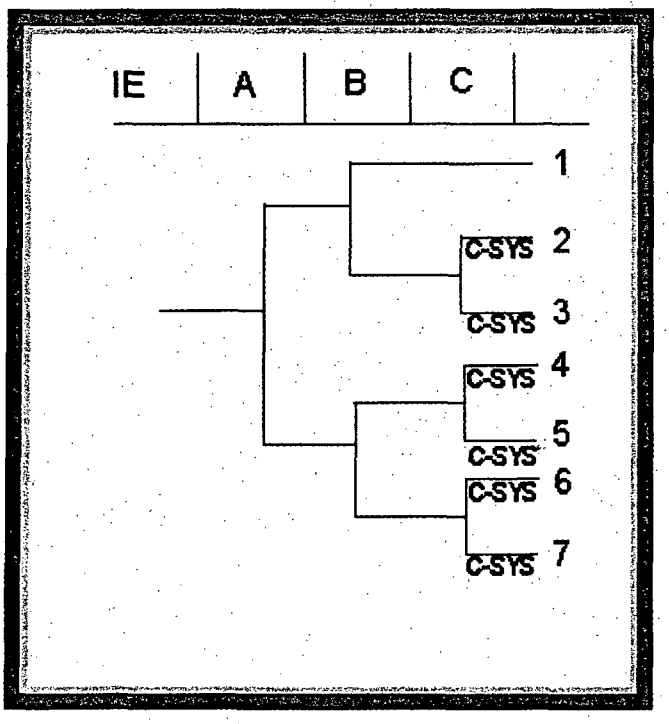
| The "??" refers to the initiating event, but will
| key on any top of only two characters.

17.4.4 always

This keyword indicates that the search criteria is always satisfied. The substitutions following this condition are always made.

| The "if-always" rule structure:
| This rule replaces every occurrence
| of C with C-SYS.
| (Sequences 2 through 7 are affected)
if always then
/C = C-SYS;

C = C-SYS;
endif



17.4.5 elsif

This keyword specifies an alternative search criteria. Any number of "elsifs" can be used within a rule. The first condition that is met is used and the others are ignored.

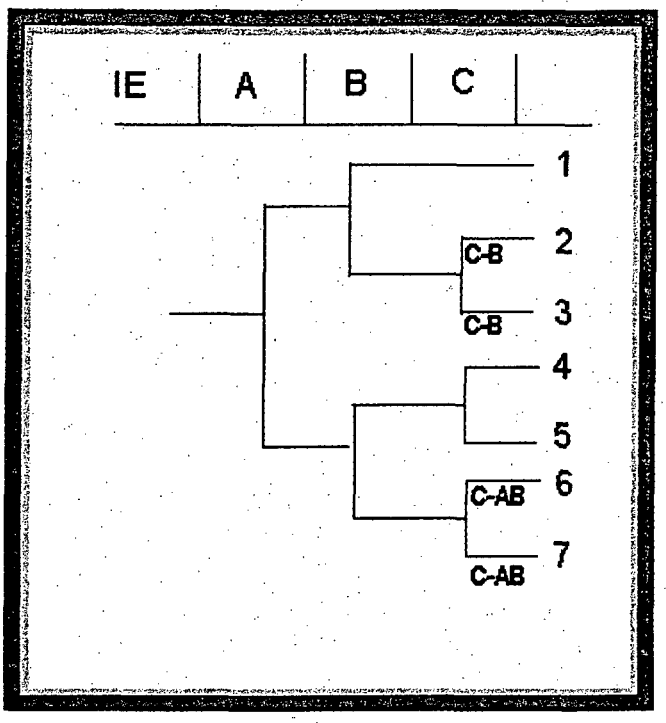
```
| The "if-then-elsif" structure:
| This rule replaces C with C-AB if
| A and B are failed, and replaces
| C with C-B if only B is failed.
if A * B then
/C = C-AB;

C = C-AB;

elsif B then
/C = C-B;

C = C-B;

endif
```



Only One Substitution Per Branch:

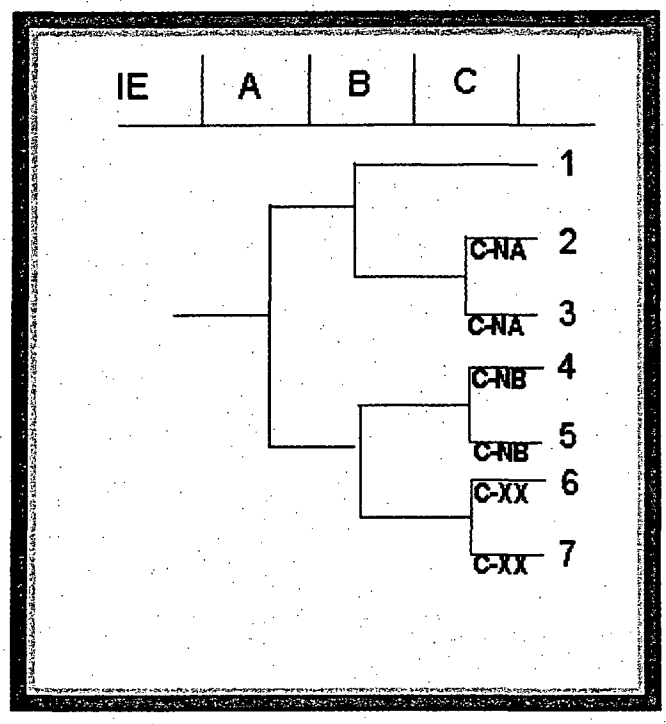
In the "elsif "rule structure, for every applicable branch, only the first substitution that applies is made. Subsequent substitutions are ignored.

In fact, the linkage rules as a whole work this way because only the first substitution for a branch is made. In other words, after a substitution has been assigned, no other rule will overwrite the substitution.

17.4.6 else

This keyword specifies that some action to be taken if all the search criteria are not met. The "else" should be the last condition in the rule.

```
| The "if-then-elsif-else" structure:
| This rule replaces C with
|  C-NA when A is successful
|  C-NB if B is successful
|  C-XX if any other case
if /A then
/C = C-NA;
C = C-NA;
elsif /B then
/C = C-NB;
C = C-NB;
else
/C = C-XX;
C = C-XX;
endif
```



17.4.7 Cast Operators

SAPHIRE provides type cast functions for the following:

- event trees
- initiators
- fault trees
- flag sets
- end states
- macros

With these functions you can define substitutions for the end state or flag set of a sequence, or define a new transfer event tree for the sequence.

For instance, if the name "SYS5" was really the name of an initiating event then the user could change its default type with the following syntax; "init(SYS5)". The editor will then treat this name as an initiating event.

OTHER CAST OPERATORS

True ()- Cast operator that sets basic events to house event TRUE.
False - Cast operator that sets basic events to house event FALSE.
Ignore - Cast operator that sets basic events to house event IGNORE.

17.4.8 init()

This keyword is used to indicate a particular initiating event.

```
if init(INITIATOR-NAME) then
perform some action ...;
endif
```

See Changing the Transfer Tree Name for a specific example of how to use this keyword.

17.4.9 eventree()

This keyword is used to indicate a particular event tree. Make note of the spelling of this keyword.

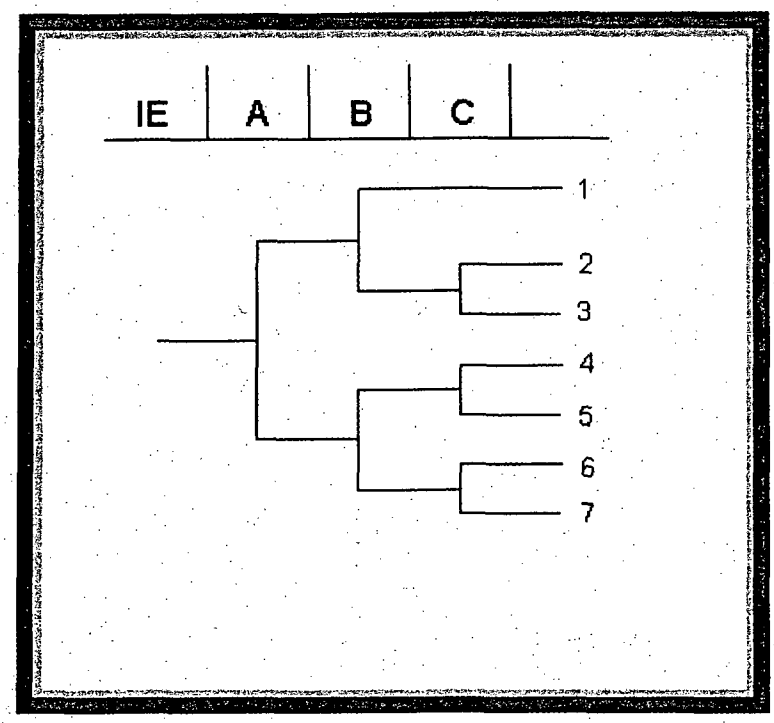
```
if "search criteria" then
eventree(EVENTREE1) = eventree(EVENTREE2);
endif
```

See Changing the Transfer Tree Name for a specific example of how to use this keyword.

17.4.10 Skip()

This keyword indicates that the specified item should be ignored. Usually, this keyword is used to "skip" sequences in the event tree logic. No data will be generated for any "skipped" sequences. The sequences are not generated when the rule is applied, however, the sequence names (numbers) are left unchanged.

```
| This rule "skips" C given the failure of B.
if B then
/C = Skip(C);
C = Skip(C);
endif
```



For this rule only sequences 1, 4, and 5 will be generated, since all sequences where B fails have been skipped.

17.4.11 True()

This rule structure is used to set the basic event(s) in parenthesis to house event TRUE.
In this example, the rule could alternatively set the basic events in parenthesis to house events FALSE or IGNORE by replacing the keyword "True" with either "False" or "Ignore".

```
| This rule sets E-MOV-A and E-PUMP-A
| to TRUE given ECS fails in the LOSP event tree.
if ECS then
eventree(LOSP) = True(E-MOV-A, E-PUMP-A);

endif
```

17.4.12 flag()

This rule structure can be used to add an existing flag set to a sequence. The flag set needs to be created using the **Modify | Flag Set** option for the information to be applied to the sequence.

```
| This rule adds the flag set "FLAG-SET-1" to
| the sequence that meets the criteria specified.
if ECS then
eventree(LOSP) = flag(FLAG-SET-1);

endif
```

17.4.13 endstate()

This rule structure can be used to add or create an end state for the sequence cut sets.

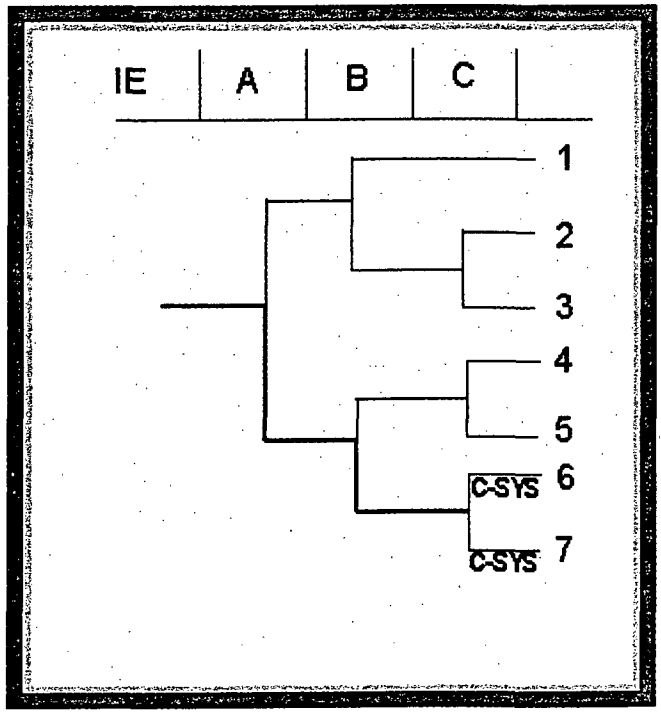
```
| This rule adds the end state "ECS-END"
| to the sequence meeting the criteria specified.
if ECS then
eventree(LOSP) = endstate(ECS-END);
endif
```

17.4.14 Macro Structures

Macros can streamline complex rules. A macro is simply a statement to define a search criteria (or variable) and assign a name to the search criteria. The macro name must be all uppercase, must be 24 characters or less, and must not include any of the restricted characters (e.g., a space, *, ?, \, /). The macro line can wrap around to more than one line, but must end with a semicolon.

Once a macro is defined, it can be used in an "if" test.

```
| Define a macro named AB-FAIL
AB-FAIL = A * B;
|
| Use the macro in a rule
if AB-FAIL then
/C = C-SYS;
C = C-SYS;
endif
```



SEE ALSO

Using ~ (never) with a Macro

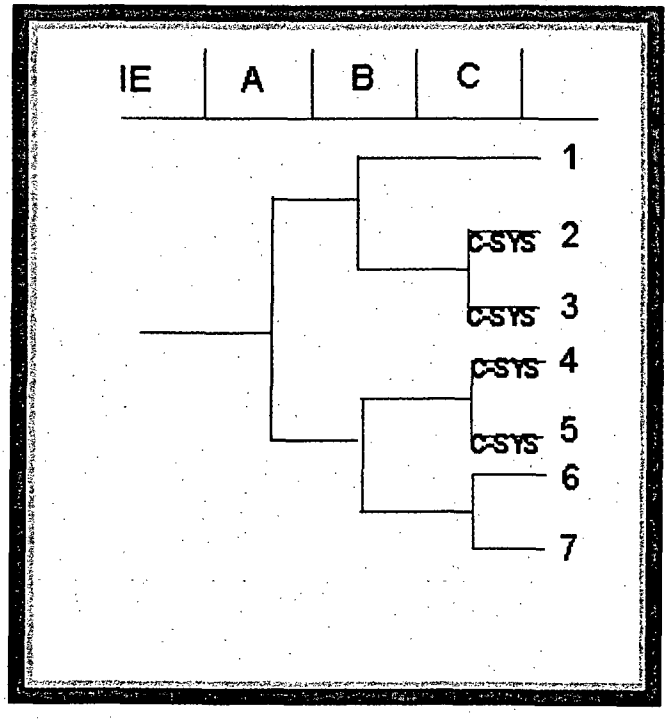
17.4.15 Using ~ (never) With a Macro

When creating a rule where the events in the macro do not occur, use the ~ (i.e., never) symbol. Do not "complement" a macro.

```
| Using the ~macro as the search criteria:
| The rule applies when A and B
| have not both failed.
|
| Define a macro named AB-FAIL
AB-FAIL = A * B;
|
| Use the macro in a rule
if ~AB-FAIL then
/C = C-SYS;

C = C-SYS;

endif
```



17.5 Transfer Tree Name

17.5.1 Changing the Transfer Tree Name

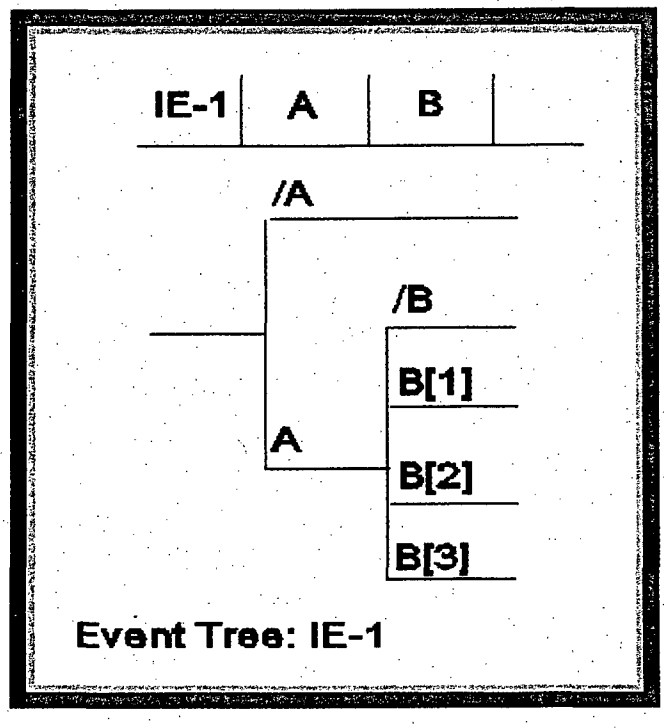
This is an example of how the transfer tree can be changed by using the init() and eventree() rules.

```
| This rule is for a transfer tree named SHARED.
|
| Two different event trees, each having a unique
| initiating event, transfer to the SHARED event
| tree. The first event tree has initiating event IE-A,
| and after it transfers to SHARED, it should transfer
| to an event tree named A-PRT. The second event
| tree has initiating event IE-B, and after it transfers
| to SHARED, it should transfer to an event tree
| named B-PRT.
|
| The transfer name on the SHARED event tree
| graphic is A-PRT. This rule changes the
| specified transfer tree to B-PRT when the
| initiator is IE-B.
if init(IE-B) then
eventree(A-PRT) = eventree(B-PRT);
endif
```

17.6 Binary and Multiple-Split Branches

17.6.1 Rules for Binary and Multiple-Split Branches

Several important modeling conventions are provided in the following example. The language for specifying a particular event tree branch is exemplified. The event tree in the figure here will be used for all rules in this discussion.

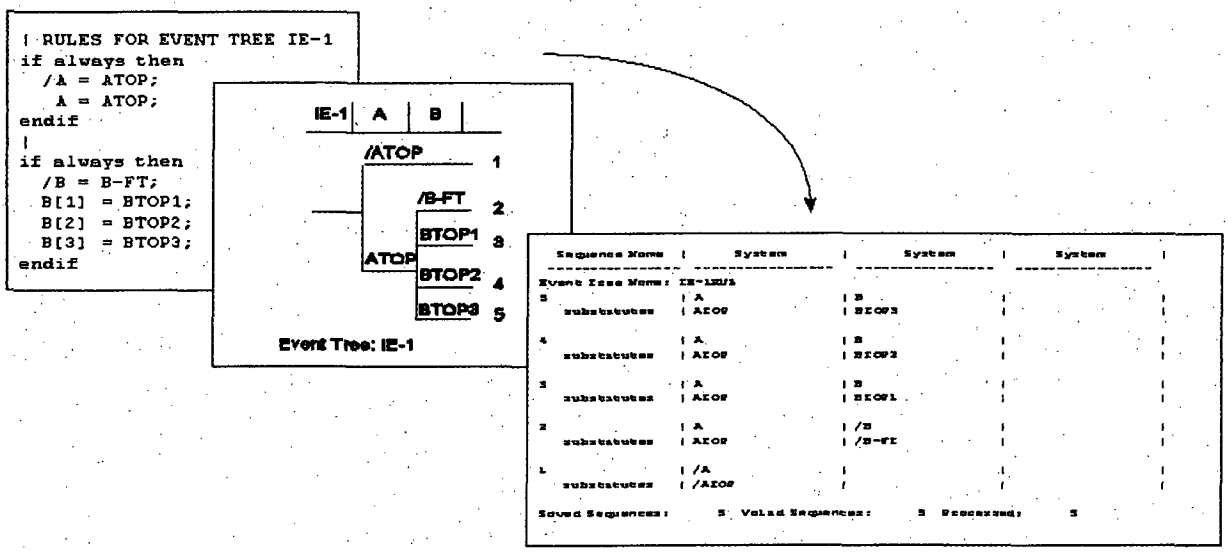


For binary branching, the success branch for a top is denoted with the complement symbol "/". SAPHIRE computes the probability for /A as (1-ATOP).

For multiple-split branching, the failed branches are designated with the top event name and the branch number in brackets.

```
| A rule for a binary branch
| Note: DO NOT specify "/ATOP"
|
if always then
/A = ATOP;
A = ATOP;
endif
|
| A rule for a multiple-split branch
|
if always then
/B = B-FT
B[1] = BTOP1;
B[2] = BTOP2;
B[3] = BTOP3;
endif
```

When you generate sequences using the event tree rules (from the **Event Tree | Link Trees** option), the sequences are reported as shown below.



To include complemented events in sequence cut sets, you must specify the **Y Process Flag** (in the **Modify | Basic Events** option) for the applicable fault tree. In this example, you would set the Y Process Flag for ATOP and B-FT.

For multiple-split branches, you may want to construct a fault tree with the name that corresponds to the substituted success branch name. The fault tree would consist of the failed branch fault trees, BTOP1, BTOP2, and BTOP3 "ORed" together. SAPHIRE will automatically complement the fault tree result. You will need to solve the fault tree and use the **S Calculation Type** (in the **Modify | Basic Events** option).

18. SLICE OPTION

18.1 Slice By Event Option

18.1.1 Displaying Cut Sets Using The Slice By Event Option

PURPOSE

This section describes the "Slice By Event" cut set display feature that allows you to divide cut sets for viewing and reporting. The "Slice By Event" option is available for fault tree, sequence, and end state cut sets.

With this option, the user can quickly subdivide a list of cut sets into two lists based upon user-defined qualification criteria. The first list contains cut sets matching the current qualification criteria. Initially, this list contains all the generated cut sets. The second list, called the unqualified cut sets, contains all the cut sets *not* matching the qualification criteria.

The *Slice By* **Event** button is available from the *Selected Cut Sets* dialog. (Via the **Display | Cut Sets** pop-up menu from fault trees, sequences or end states lists.) Once the slice by event option is selected, the *Event List* dialog is displayed. This is the dialog used to define the criteria upon which cut sets will be qualified.

STEPS

To define qualification criteria:

1. Choose the slice type (Event, Cutoff, or Rule)

2. If Event, then

- Specify the logic option. Choose either "AND" or "OR" from the *Logic* drop-down list.

  Choosing AND tells SAPHIRE that all of the selected events must appear in the cut set to meet the qualification criteria.

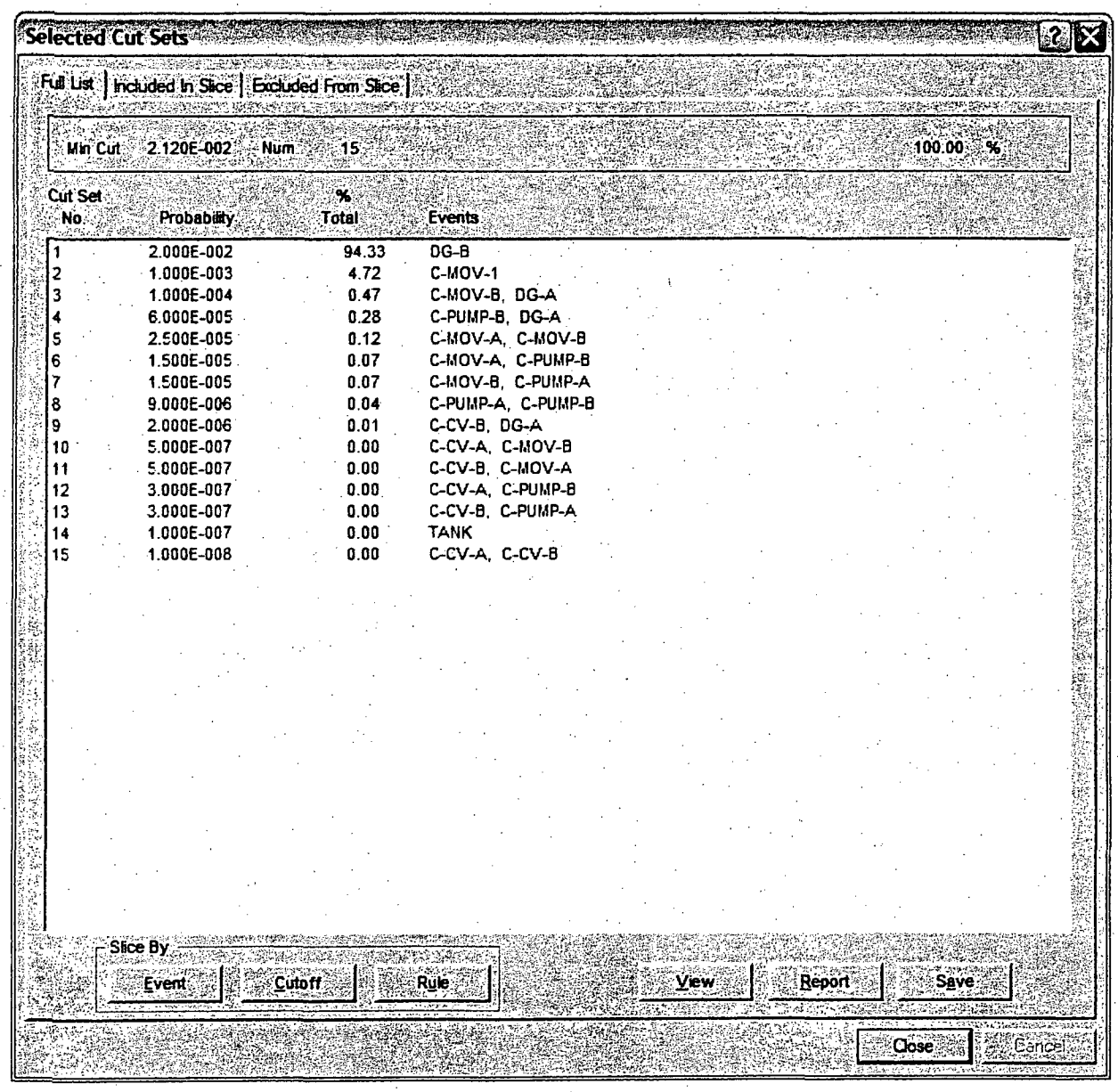
  Choosing OR tells SAPHIRE that any of the selected events can appear in a cut set for the cut set to meet the qualification criteria.

- Specify the events on which the qualification criteria are based. Three ways of specifying basic events are provided:

  Choose the Wild Card Mark option,

  Highlight the desired event(s) in the *Cut Set Events* list and right-click to invoke the pop-up menu. Then choose from one of the four types: (Normal) Event, Failure Event, Success Event, or NOT Event.

  Highlight the desired event(s) in the *Cut Set Events* list and choose the → button. This method adds (Normal) Event(s) to the *Selected Events* list. Conversely, choosing the ← button removes the highlighted event(s) from the *Selected Events* list.



As an example, the qualification criteria events are the four events in the *Selected Events* list and the *Logic* to be performed is the "OR" operation. This tells SAPHIRE that any cut set:

containing the failure of event "C-CV-A" (+C-CV-A) or
containing the success of event "C-MOV-1" (/C-MOV-1) or
not containing the event "C-PUMP-A" (~C-PUMP-A) or
containing the event "DG-A" (DG-A)

will meet the qualification criteria.

Thus, the evaluation can be defined in terms of successes, failures, exclusion (not contained in), or inclusion (contained in) for any of the basic events in the model.

Choose the **Apply** button. SAPHIRE searches the cut sets for the qualified events (i.e., those matching the criteria). The *Event List* dialog will be closed and only the qualified cut sets (i.e., those containing events matching the criteria) will be displayed in the *Selected Cut Sets* dialog.

18.1.2 Slice By Event Examples

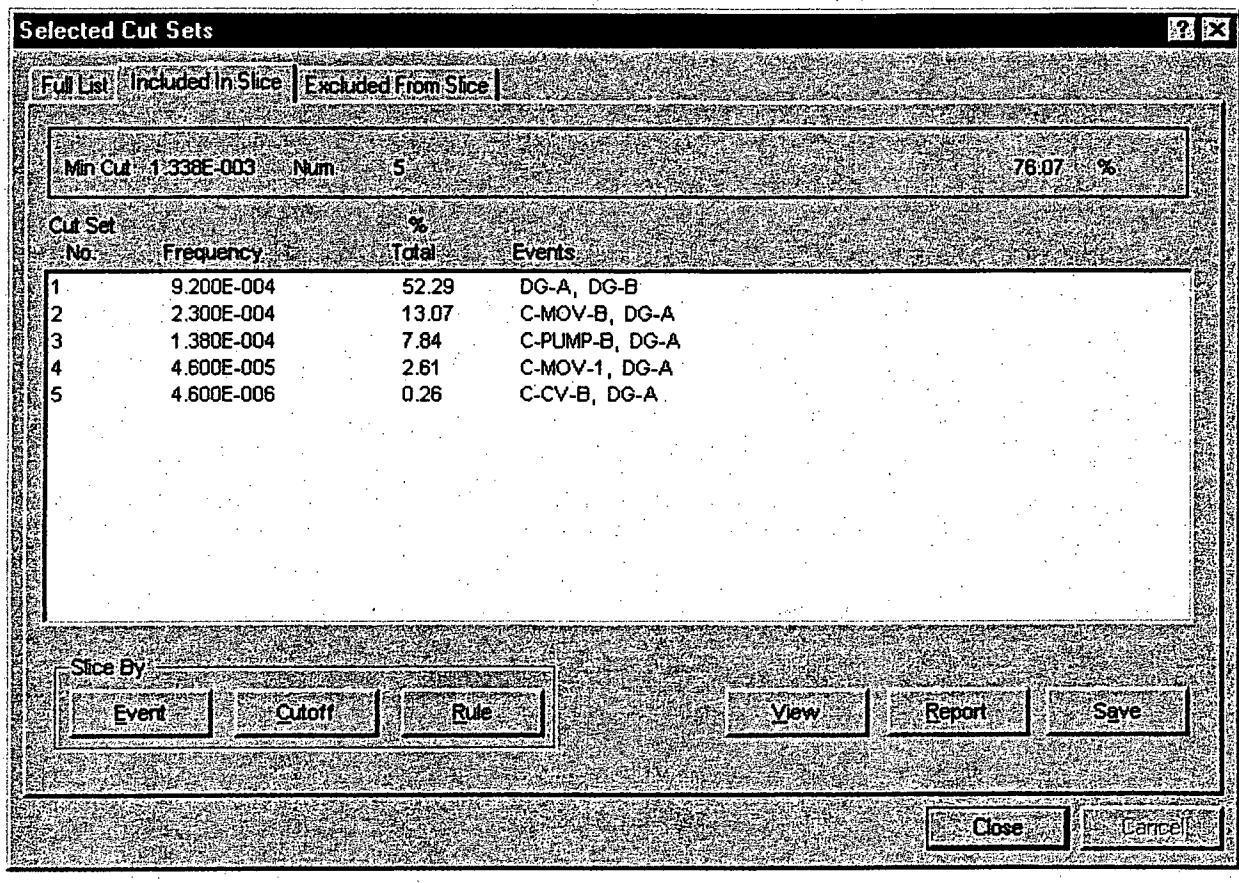
18.1.2.1 Viewing Cut Sets Containing DG-A

PURPOSE

This example demonstrates a simple use of the cut set slice option. A single event will be selected with the qualification criteria defined in terms of *either* failure or success for the event (i.e., the state of the event is irrelevant).

STEPS

1. Select **Sequence** from the SAPHIRE menu.
2. The *Sequences* dialog will be displayed. Highlight LOSP 3.
3. Right-click to invoke the pop-up menu and choose **Display | Cut Sets**.
4. The *Selected Cut Sets* dialog will be displayed with all of the cut sets generated for the LOSP 3 sequence. Choose the *Slice By* **Event** button.
5. The *Event List* dialog will be displayed. Highlight DG-A in the *Cut Set Events* list.
6. Right-click to invoke the pop-up menu and choose **Add Event**.
7. Choose the **Apply** button. The *Event List* dialog will be closed and the qualification criteria will be evaluated.
8. The *Included In Slice* tab will be displayed in the *Selected Cut Sets* dialog, showing only those cut sets meeting the qualification criteria. The *Excluded From Slice* tab will show the unqualified cut sets.



Notice that in this example, only five cut sets have been included that contain the basic event DG-A. The number of cut sets, their percent contribution, and subtotal are shown on the bottom right hand side of the dialog.

If you select the ***Excluded From List*** tab, the cut sets that do not contain DG-A will be displayed.

18.1.2.2 Using the Slice Option to Select DG-A or DG-B

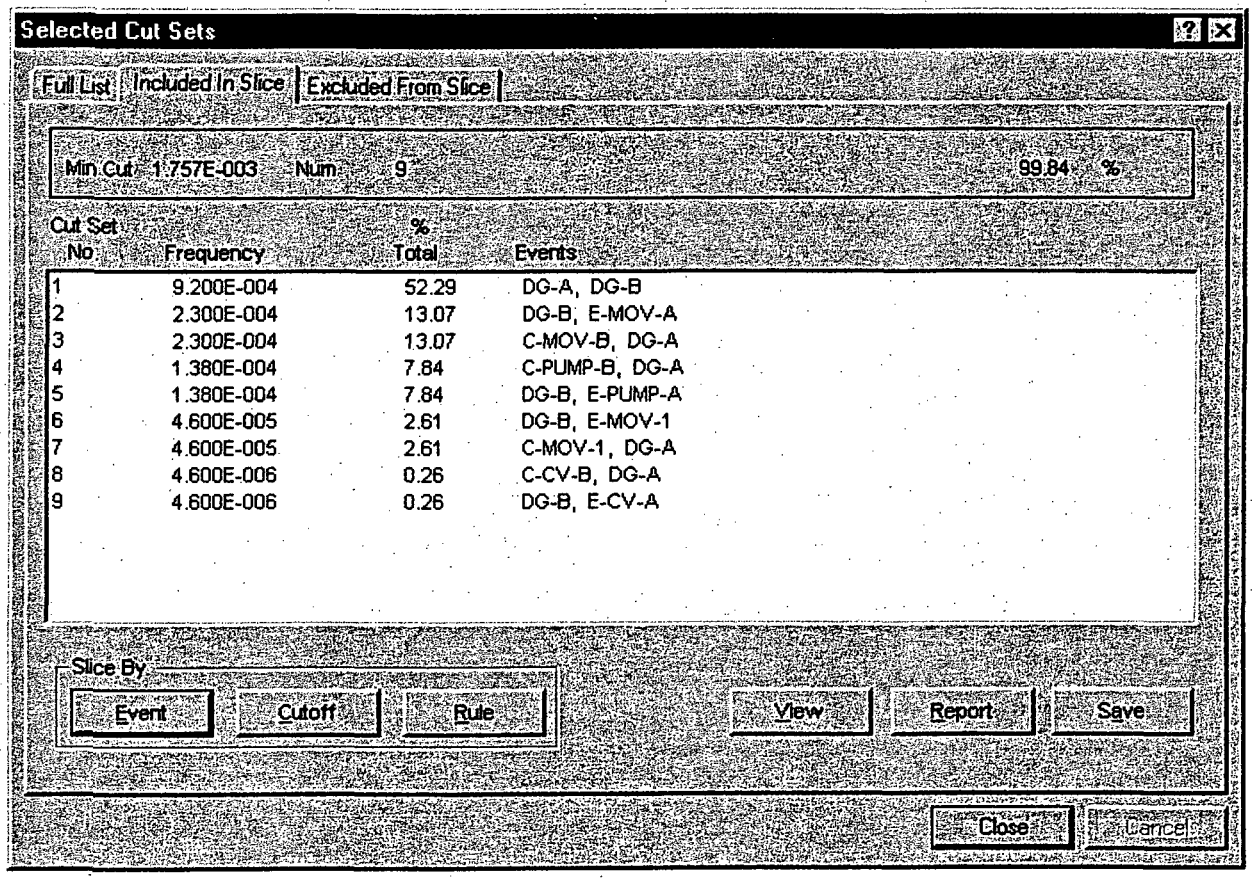
PURPOSE

This example demonstrates how to use the **Wild Card Mark** feature to select multiple basic events.

Notice when using this option that the events matching the event class mask are highlighted in the *Cut Set Events* list, but are not automatically added to the *Selected Events* list. This provides the analyst with the opportunity to choose the state of the highlighted events (e.g., failed, success, not in), via the pop-up menu, before they are added to the *Selected Events* list.

STEPS

1. To start a fresh slice session for LOSP 3, choose the *Slice By* **Event** button (from the *Selected Cut Sets* dialog).
2. On the *Event List* dialog, choose the **Restart** button. The original cut set list for LOSP 3 will be restored in the *Selected Cut Sets* dialog.
3. Choose the *Slice By* **Event** button again.
4. On the *Event List* dialog, select the "OR" **Logic** option.
5. Right-click to invoke the pop-up menu and choose the **Wild Card Mark** option.
6. The *Event Class Mask* dialog will be displayed. Enter "DG" in the *Type* field.
7. Choose the **Ok** button. The *Event Class Mask* dialog will be closed and the DG-A and DG-B events will be highlighted in the *Cut Set Events* list.
8. Choose the **Add Event** option from the pop-up menu. Both events will be included in the *Selected Events* list.
9. Choose the **Apply** button. The *Event List* dialog will be closed and the qualification criteria will be evaluated.
10. The *Included In Slice* tab will be displayed in the *Selected Cut Sets* dialog, showing only those cut sets meeting the qualification criteria. The *Excluded From Slice* tab will show the unqualified cut sets.



Now there are nine cut sets displayed. Each cut set contains either DG-A or DG-B (or both).

If you select the *Excluded From List* tab the cut sets that do not contain either DG-A or DG-B will be displayed.

If you leave the **Logic** option set to "AND" (the default), only one cut set will meet the qualification criteria. This cut set includes both DG-A *and* DG-B.

Alternate ways to select both DGs:

- Highlight the basic events in the *Cut Set Events* list on the *Event List* dialog.
- Enter "DG-?" in the *Primary Name* field on the *Event Class Mask* dialog.

18.2 Slice By Rule Option

18.2.1 Displaying Cut Sets Using The Slice By Rule Option

PURPOSE

This section describes the "Slice By Rule" cut set display feature that allows you to divide cut sets for viewing and reporting. This "Slice By Rule" option is available for fault tree, sequence, and end state cut sets.

With this option, the user can quickly subdivide a list of cut sets into two lists based upon user-defined qualification criteria. The first list contains cut sets matching the current qualification criteria. Initially, this list contains all the generated cut sets. The second list, called the unqualified cut sets, contains all the cut sets *not* matching the qualification criteria.

The *Slice By* **Rule** button is available from the *Selected Cut Sets* dialog (via the **Display | Cut Sets** pop-up menu from fault trees, sequences or end states lists). Once the slice option is selected, the *Slice Rules* dialog is displayed. This is the dialog used to manage the list of slice rules upon which cut sets will be qualified. Right clicking in this dialog displays a menu to Add, Copy, Modify, and Delete slice rule name and descriptions. Once a slice name has been defined and highlighted, the **Edit Rule** button is used to display the *Slice Rule Editor*, where search criteria for the rule is specified.

The slice rules are "free-form" logic rules that allow for the inclusion or exclusion of cut sets from a "keep" list. They follow a format similar to the structure that is found in traditional programming languages (e.g., BASIC or PASCAL). As such, the ability exists to define "macros" and "if...then" types of structures. The rules are entered in a free-form text editor within SAPHIRE. They can also be exported and imported through the SAPHIRE **Utility | Load and Extract** option.

Upon saving and exiting the rule logic, SAPHIRE compiles the rules to check their validity. Errors will be reported to the user if found; otherwise the editor will exit normally. To apply a slice rule to the full list of displayed cut sets, highlight the desired rule and press the **Apply** Button. Cut sets will be divided into the *Included in List* tab and *Excluded From List* tab according to the rule logic.

To clear the Included and Excluded lists, choose the **Reset** option from the *Slice By Event* option, or, apply a different slice option to override the current option.

EXAMPLES

Viewing Cut Sets Containing DG-A
Using the Slice Option to Select DG-A and DG-B

18.2.2 Search Criteria Examples - Slice Rules

These examples are used based on basic events X, Y, and Z.

Search Criteria	Meaning of Search Criteria
X	Basic event X appears in the cut set
/X	Success of event X appears in the cut set
~X	Basic event X never occurs in the cut set
X + Y	Either event X or Y appear in the cut set
X * Y	Both events X and Y appear in the cut set
~X * Y	Y does appear and X does not appear in the cut set
X * (Y + Z)	Either X and Y or X and Z appear in the cut set
always	This pre-defined macro name means the criteria is always met

18.2.3 Rule Editor Symbols

| (comment)	Denotes a comment line
* (AND)	Logical AND operator
+ (OR)	Logical OR operator
~ (never)	Logical operator for "never" or "not present"
/ (complement)	Complement
; (end macro)	Indicates the end of a macro line or a line that modifies the criteria being evaluated.
( )	Parentheses for grouping terms

18.2.4 Slice Rule Actions

The slice utility provides the ability to divide a list of cut sets into two groups, or slices: the *Included in Slice* group, and the *Excluded From Slice* group. This feature allows the analyst to locate and separate out cut sets containing a specified combination of events. As such, there are two actions a rule can perform. The "keep" action will include qualified cut set in the *Included in Slice* group, and the "discard" action will include unqualified cut sets in the *Excluded From Slice* group.

Note that when a cut set is evaluated, and does not meet the specified search criteria, the default action is to "discard". Therefore, if the slice action is "discard" (which may sometimes be a desirable way to frame the search logic), you must include a "keep" action in order to separate unqualified cut sets from those the qualification is discarding.

To demonstrate the potential uses of the slice rules, the example here shows how the rules could be used to remove cut sets relating to loss of off-site power and diesel generator failures. As shown, the rule will place any cut sets having the initiating event LOSP and either diesel generator A or B into the *Excluded from Slice* List. Those cut sets that do not meet this criteria will be placed into the *Included In Slice* list.

| Search on LOSP initiator and failure of
| diesel generators A or B and remove them from display.
if init(LOSP) * (DG-A + DG-B) then
discard;

else
keep;

endif

When this rule is applied to the displayed cut sets, SAPHIRE will evaluate each cut set and then automatically separate each one into the appropriate list. The "if...then" line can be as complex as needed.

As written, the above rule would only discard cut sets for sequence cut sets, since fault tree cut sets do not contain initiating events.

18.2.5 Keywords and Symbols

18.2.5.1 Keywords and Symbols - Slice Rules

Below is a list of keywords and symbols used in the SAPHIRE slice rules. The search criteria and rule structure of the slice rules is parallel to that of recovery rules; only the recovery and slice actions differ.

Note the spelling of each keyword.

if-then - Indicates a search criteria is being specified.
endif - Indicates the end of a particular rule.
else - Specifies some action to be taken if all the search criteria are not met. The "else" should be the last condition in the slice rule.
elsif - Specifies an alternative search criteria.
always - Indicates that every cut set that is being evaluated satisfies the search criteria.
init()- Used in the search criteria to indicate that a sequence cut set has a particular initiating event.
~ (never) - Used in the search criteria to indicate that a particular event ***will not*** be in the cut set that is being evaluated.
/ (complement) - Represents a complemented event (i.e., the success of a failure basic event).
| (comment) - Represents a comment contained in the rules.
; (end macro) - Indicates the end of a macro line or a line that directs the destination of the cut set being evaluated.
*** (AND)** - Indicates the logical AND command.
+ (OR) - Indicates the logical OR command.
( ) - Indicate a specific grouping of items.
keep; - Indicates that the cut set being evaluated will be placed in the *Included In Slice* list.
discard; - Indicates that the cut set being evaluated will be placed in the *Excluded From Slice* list.
MACRO - A user-definable keyword that specifies a search criteria.

18.2.5.2 if-then

(Basic Rule syntax)

This keyword indicates that a search criteria is being specified.

if "search criteria" **then**
perform some action on each cut set…;

endif

18.2.5.3 endif

(Basic Rule syntax)

This keyword indicates that the end of a particular rule.

if "search criteria" then
perform some action on each cut set…;

endif

18.2.5.4 else

(Basic Rule syntax)

This keyword specifies that some action to be taken if all the search criteria are not met. The "else" should be the last condition in the recovery rule.

if "search criteria" then
perform some action on each cut set…;

else
perform some other action on each cut set if "search criteria" not met…;

endif

18.2.5.5 elsif

(Basic Rule syntax)

This keyword specifies an alternative search criteria. Any number of "elsifs" can be used within a recovery rule.

if "search criteria" then
perform some action on each cut set…;

elsif "2nd search criteria" then

perform some other action on each cut set …;

elsif "3rd search criteria" then
perform some other action on each cut set …;

endif

18.2.5.6 always

(Basic Rule syntax)

This keyword indicates that every cut set that is being evaluated satisfies the search criteria.

if **always** then
perform some action on each cut set…;

endif

18.2.5.7 init()

(Basic Rule syntax)

This keyword is used in the search criteria to indicate that a sequence cut set has a particular initiating event.

if **init**(INITIATOR-NAME) "and optional other search criteria" then
perform some action on each cut set…;

endif

18.2.5.8 ~ (never)

(Basic Rule syntax)

This symbol is used in the search criteria to indicate that a particular event ***will not*** be in the cut set that is being evaluated.

if (~SEARCH-CRITERIA) "and optional other search criteria" then
…

The search criteria will be satisfied for cut sets that do not contain SEARCH-CRITERIA (and also contains the optional ("other search criteria"). SEARCH-CRITERIA may be either an initiating event, basic event, macro, or logic expression.

18.2.5.9 | (comment)

(Basic Rule syntax)

This symbol is used to represent a comment contained in the rules. Everything on a line to the right of this symbol will be ignored by the rule compiler. Blank lines are encouraged in the editor to improve readability and comments can be entered following the "|" character.

```
| Place your comments here!

| Note that blank lines are also permissible.
```

18.2.5.10 / (complement)

(Basic Rule syntax)

This symbol is used to represent a complemented event (i.e., the success of a failure basic event).

```
if (/BASIC-EVENT) "and optional other search criteria" then
```

The search criteria will be satisfied for all cut sets that contain the complement of BASIC-EVENT (and also contains the optional "other search criteria").

18.2.5.11 ;(end macro)

(Basic Rule syntax)

This symbol is used to indicate the end of a macro line or a line that modifies the cut set being evaluated.

```
| usage for a macro command
MACRO-NAME = "search criteria";

| usage for a cut set modification line
recovery = RECOVERY-EVENT;
```

*18.2.5.12 * (AND)*

(Basic Rule syntax)

This symbol is used to indicate the logical AND command.

```
if SEARCH-CRITERIA1 * SEARCH-CRITERIA2 then
```

The search criteria will be satisfied for all cut sets that match SEARCH-CRITERIA1 and SEARCH-CRITERIA2. The SEARCH-CRITERIA# may be either an initiating event, macro, or logic expression.

18.2.5.13 + (OR)

(Basic Rule syntax)

This symbol is used to indicate the logical OR command.

if SEARCH-CRITERIA1 + SEARCH-CRITERIA2 then

The search criteria will be satisfied for all cut sets that match either SEARCH-CRITERIA1 or SEARCH-CRITERIA2. The SEARCH-CRITERIA# may be either an initiating event, macro, or logic expression.

18.2.5.14 ( ) (group)

(Basic Rule syntax)

These symbols together indicate a specific grouping of items.

if (A + B) * (C + D) then

The search criteria above would return all cut sets that contain:
[A * C], [A * D], [B * C], or [B * D].

18.2.6 Slice By Rule Examples

18.2.6.1 Slice Rule Example: View MOV pairs

PURPOSE

This example demonstrates how to write a slice rule containing mixed logic to view only cut sets which have one or more groups of events in them. The ability to specify one or more groups of events is a feature available only through the *Slice by* **Rule** option.

The example also makes use of the "discard" keyword. Whenever the "discard" keyword is used, the "keep" keyword must also be used, because the default action for unqualified cut sets is to discard.

STEPS

1. From the *Selected Cut Sets* dialog, choose the *Slice By* **Rule** button. The *Slice Rules* dialog will appear.

2. Right click to invoke the pop-up menu and choose the **Add** menu option.

3. Give the rule a name, such as "MOV-RULE", and, optionally, a description, such as "find MOV groups".

4. Choose the **OK** button to close the *Add Rule* dialog and save the new rule .

5. With the new rule still highlighted, choose the **Edit Rules** button. The *Rules Editor* window will appear.

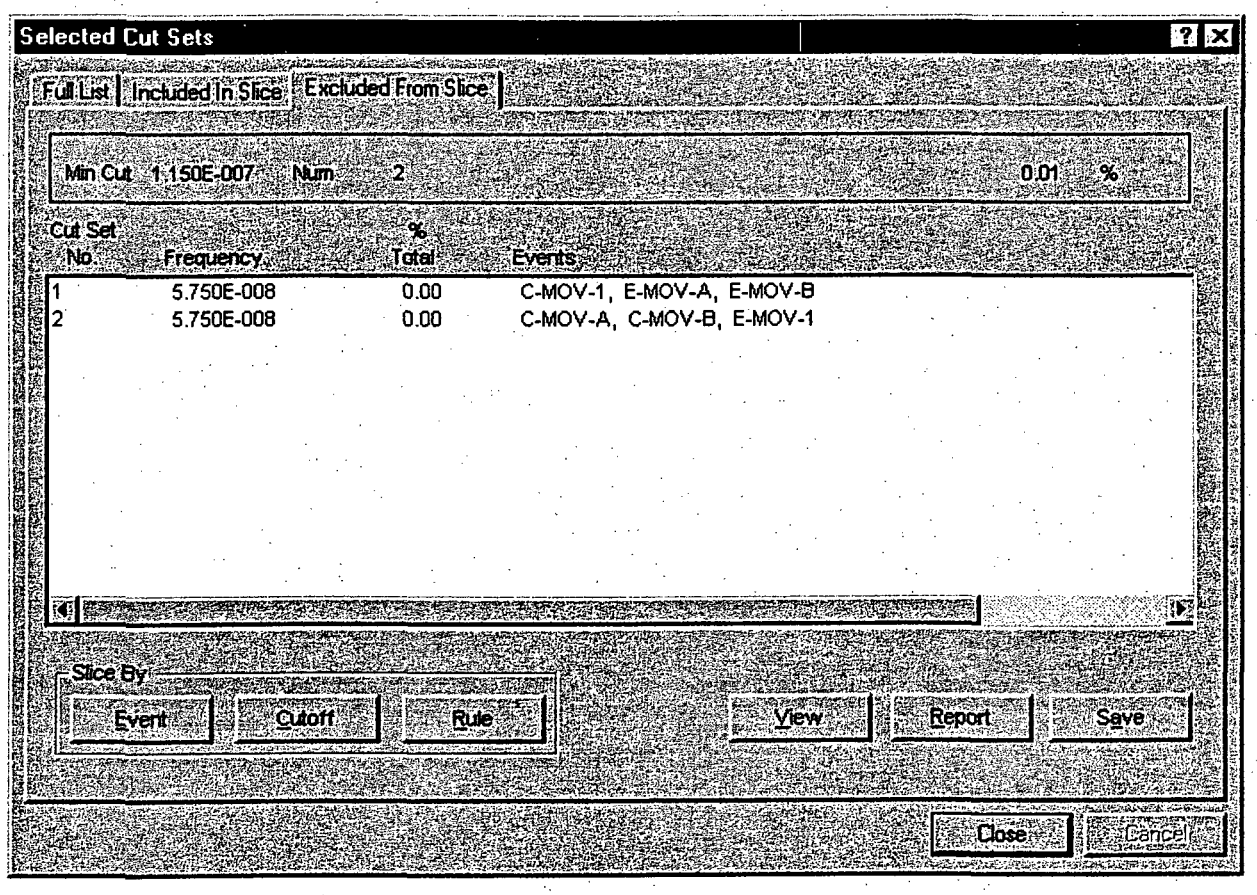
6. Type the following rule:

```
if (C-MOV-A * C-MOV-B) + (E-MOV-A * E-MOV-B) then
  discard;
else
  keep;
endif
```

7. From the *Edit Rules* menu, choose **File | Exit** to close the *Rules Editor*. When prompted to save the file, choose **Yes**.

8. From the *Slice Rules* dialog (with the new rule still highlighted), choose the **Apply** button. The *Slice Rules* dialog will be closed and the cut sets will be evaluated against the rule.

9. The *Included In Slice* tab will be displayed in the *Selected Cut Sets* dialog, showing only those cut sets which *did not* meet the qualification criteria. Because the "discard" keyword was used, it is the *Excluded From Slice* tab that shows the qualified cut sets.



The alternate form of the slice rule that will reverse the contents of the *Included In Slice* and *Excluded From Slice* tabs is as follows:

```
if (C-MOV-A * C-MOV-B) + (E-MOV-A * E-MOV-B) then
  keep;
endif
```

18.2.6.2 Slice Rule Example: View DG-A or DG-B cut sets

PURPOSE

This example demonstrates how to write a slice rule to view only cut sets which have one or more specified events in them.

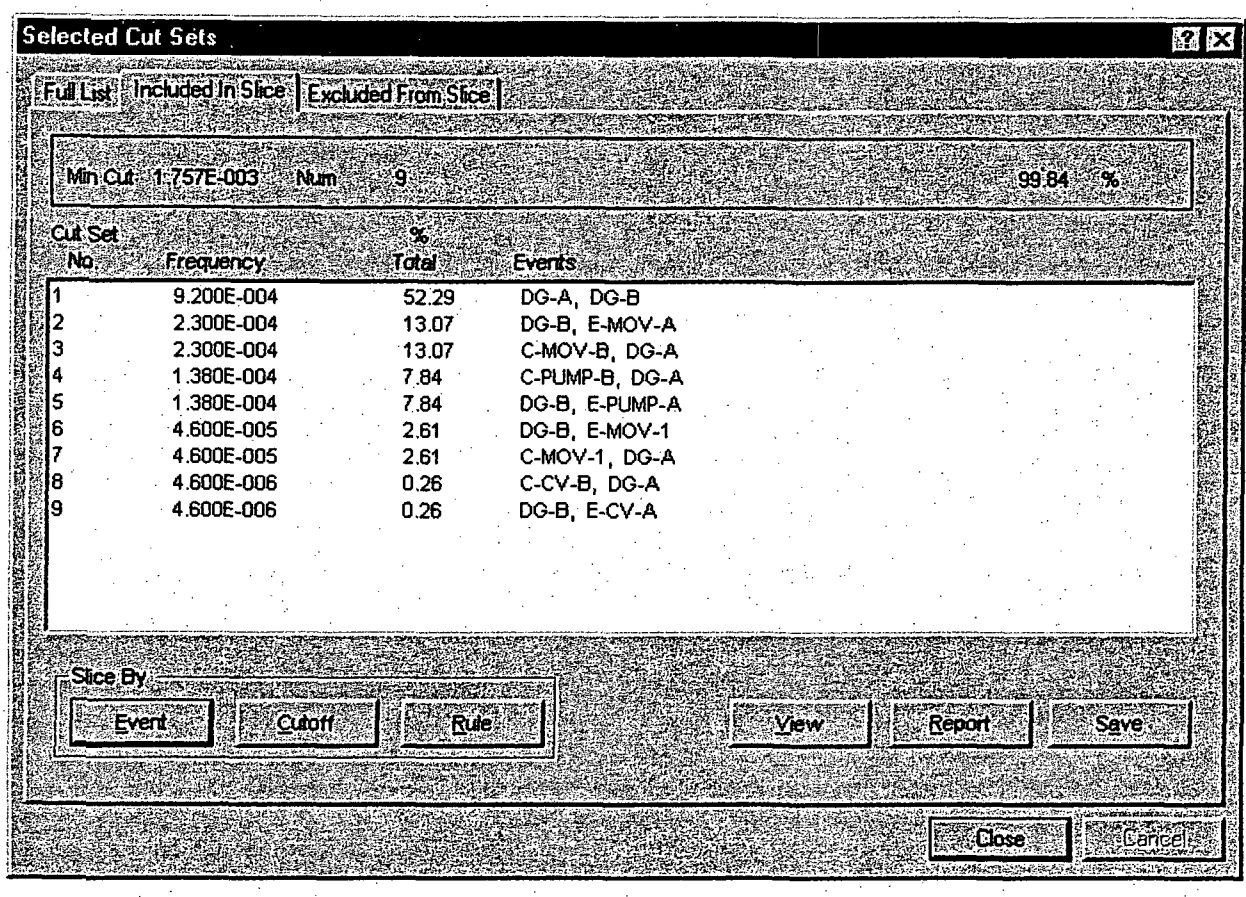
In this example, the cut sets in the DEMO project will be "sliced" according to whether or not they contain DG-A or DG-B. See the *Slice By* **Event** example for an alternate method of performing the same task.

STEPS

1. From the *Selected Cut Sets* dialog, choose the *Slice By* **Rule** button. The *Slice Rules* dialog will appear.
2. Right click to invoke the pop-up menu and choose the **Add** menu option.
3. Give the rule a name, such as "DG-RULE", and, optionally, a description, such as "find any diesel generators".
4. Choose the **OK** button to close the *Add Rule* dialog and save the new rule .
5. With the new rule still highlighted, choose the **Edit Rules** button. The *Rules Editor* window will appear.
6. Type the following rule:

   ```
   if DG-A + DG-B then
   keep;
   endif
   ```

7. From the *Edit Rules* menu, choose **File | Exit** to close the *Rules Editor*. When prompted to save the file, choose **Yes**.
8. From the *Slice Rules* dialog (with the new rule still highlighted), choose the **Apply** button. The *Slice Rules* dialog will be closed and the cut sets will be evaluated against the rule.
9. The *Included In Slice* tab will be displayed in the *Selected Cut Sets* dialog, showing only those cut sets meeting the qualification criteria. The *Excluded From Slice* tab will show the unqualified cut sets.



Now there are nine cut sets displayed. Each cut set contains either DG-A or DG-B (or both).

If you select the *Excluded From List* tab the cut sets that do not contain either DG-A or DG-B will be displayed.

The following variation of the above rule would reverse the results; The *Excluded From List* tab would contain the cut sets having either DG-A or DG-B (or both). The *Included In List* tab would contain the cut sets that do not contain either DG event. The "else keep" portion of the rule is necessary, because by default, any cut sets not meeting the rule criteria are placed in the *Excluded From List*, (i.e., discarded).

```
if DG-A + DG-B then
  discard;
else
  keep;
endif
```

19. DYNAMIC FLAG SETS

19.1 Dynamic Flag Sets

Dynamic flag sets are a special type of flag set that are assigned to sequences by the use of event tree rules. The dynamic flag set is assigned to a sequence or sequences if they meet the criteria contained in the rule.

The advantage of using dynamic flag sets is evident when event tree logic changes are made which causes changes to the number of sequences. Since dynamic flag sets are assigned to sequences defined in the event tree rule, the dynamic flag set will automatically reassign itself to the correct sequence. Conversely, if regular flag sets were used, the analyst would have to modify each sequence individually to make sure the correct flag set name is assigned to the correct sequence. This makes the use of dynamic flag sets more efficient than regular flag sets if flag sets are assigned to specific sequences.

Dynamic flag sets can contain only individual basic event changes. No "Class Changes" are allowed.

Dynamic flag sets can contain only house event settings (calculation type T, F, or I changes) or top event process flag changes. No probability changes can be made.

Dynamic flag sets will appear in the same list as regular Flag Sets, however, the name given to a dynamic flag set (e.g., ET-000001-000001) is based upon the event tree, sequence name, and number of dynamic flag sets already created.

TOPICS

Search Criteria Examples
Symbols
Keywords and Structures
Making a Dynamic Flag Set
Example Dynamic Flag Set
Flag Sets and Subtrees

19.2 Rule Structures

19.2.1 Rule Editor Symbols

| (comment)	Denotes a comment line
* (AND)	Logical AND operator
+ (OR)	Logical OR operator
~ (never)	Logical operator for "never" or "not present"
/ (complement)	Complement
; (end macro)	Indicates the end of a macro line or a line that modifies the criteria being evaluated.
( )	Parentheses for grouping terms

19.2.2 Linkage Rules Keywords and Structure

if-then -	Indicates that a search criteria is being specified.
if-then using wildcards -	A search criteria can be specified using wildcards.
always -	Indicates that the search criteria is always satisfied.
elsif -	Specifies an alternative search criteria.
else -	Specifies some action to be taken if all the search criteria are not met.
Cast Operators -	Define substitutions for the end state or flag set of a sequence, or define a new transfer event tree
Skip -	Indicates that the specified item should be ignored.
Macro -	A user-definable keyword that specifies a search criteria.

19.2.3 Flag Sets and Sub-trees

SAPHIRE has an enhanced feature that will append a flag set to the sequence meeting the search criteria even if the event tree stated in parenthesis transfers to a subtree. Therefore, either rule stated below will append a flag set to the same sequence.

if init(initiating-event) then
eventree(main-event-tree) = True(event1);

endif
or
if init(initiating-event) then
eventree(subtree) = True(event1);

endif

19.3 Making a Dynamic Flag Set

19.3.1 Making a Dynamic Flag Set

PURPOSE

Describe the steps for creating a dynamic flag set.

STEPS

1. Select **Event Tree** from the SAPHIRE menu.
2. In the *Event Tree List*, highlight the desired event tree.
3. Right-click to invoke the pop-up menu and select **Edit Rules**.
4. Using the rule structures discussed, construct a rule that will modify a basic event's calculation type and exit the editor.
5. Generate the event tree sequences by selecting **Link Trees** from the pop-up menu.
6. The flag set will automatically be assigned to the sequence without manually modifying the sequence.
7. The event tree sequences are ready to be analyzed from the **Sequence** menu option.

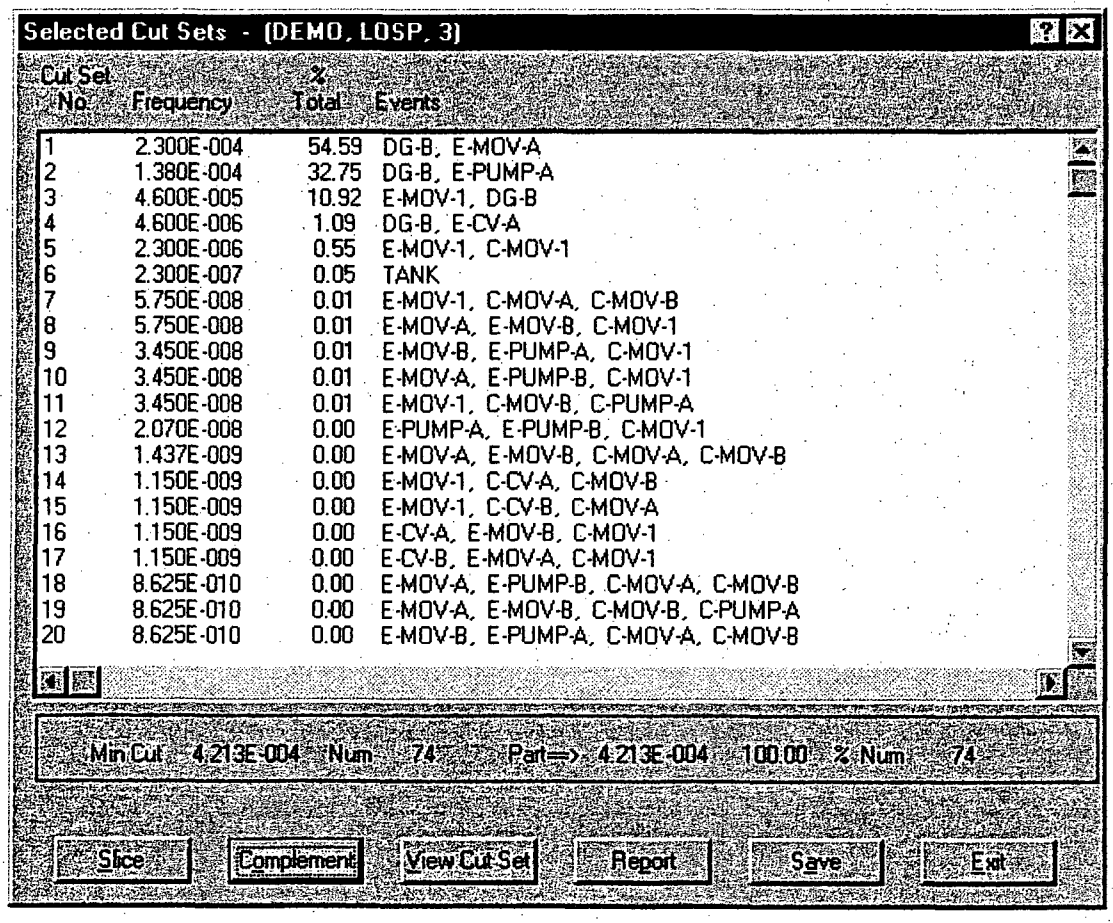
19.3.2 Example Dynamic Flag Set

To illustrate the use of dynamic flag sets, the DEMO project will be used.

A rule was written to set DG-A and C-PUMP-B to a house event FALSE given CCS fails in the LOSP event tree.

```
if CCS then
eventree(LOSP) = False(DG-A, C-PUMP-B);
endif
```

The dynamic flag set will append itself to the sequences meeting the criteria specified in the rule. For this particular rule, Sequence 3 will have a flag set associated with it that will set DG-A and C-PUMP-B to a house event FALSE.



The resulting cut sets for Sequence 3 are shown above. Notice that DG-A and C-PUMP-B do not show up in the list of cut sets.

20. MUTUALLY EXCLUSIVE EVENTS

20.1 Mutually Exclusive Events

PURPOSE

This section presents the topic of mutually exclusive events. The term is defined here, a fault tree logic example is provided, and the methods to remove these events from SAPHIRE PRA results are discussed.

The term "mutually exclusive events" refers to two or more basic events that appear in a single cut set (either for fault trees or sequences) which logically should not appear together. Generally, mutually exclusive events should not appear together in the list of cut sets for one of two reasons:

Plant technical specifications or other operating restrictions may prevent two components from being out of service (e.g., tested or maintained) at the same time. An example is not allowing two AFW pumps to be simultaneously out of service for testing and maintenance.

Other general logical modeling concerns may lead the analyst to remove specific combinations of events. An example of this involves the practice of including multiple initiating events in the fault tree logic. Given this case, sequence cut sets can e generated that include multiple initiating events.

Most mutually exclusive groups include only two or three components.

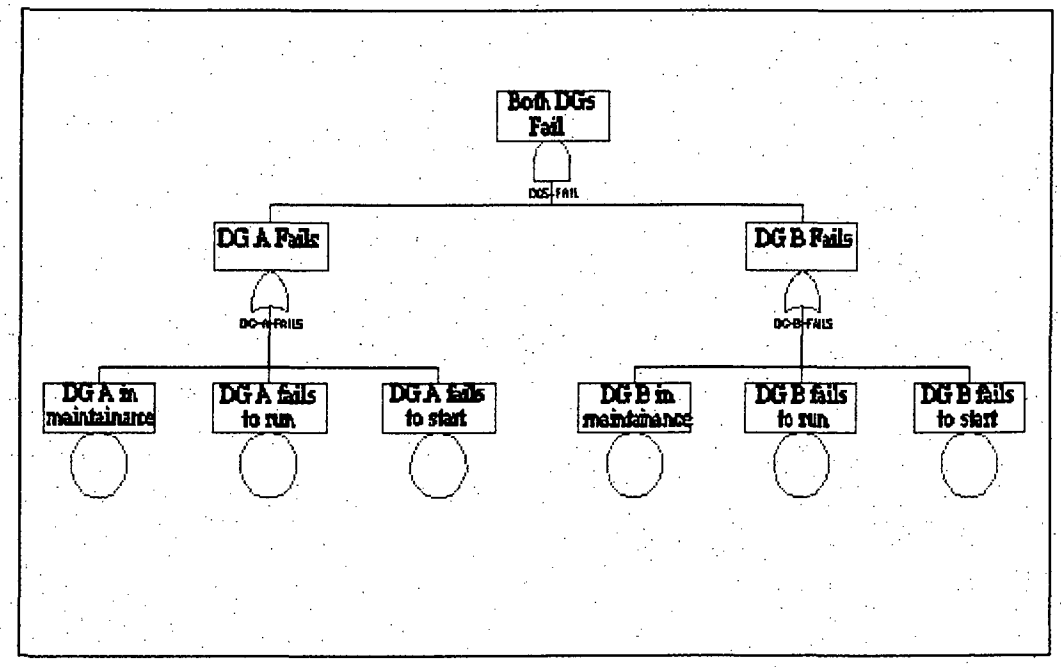
During the logic modeling phase, the analyst may recognize that certain combinations of mutually exclusive events will appear just by knowing how the fault tree and event tree logic modeling was performed. However, some unrecognized mutually exclusive events may not be evident until the analyst generates and evaluates the fault tree or sequence cut sets.

TOPICS

Mutually Exclusive Event Example
Removing Mutually Exclusive Events

20.2 Mutually Exclusive Event Example

As an example of how fault tree logic modeling can produce mutually exclusive events, the fault tree shown below will be used.



Generating cut sets for this fault tree will produce a cut set containing the two maintenance events:

DG-A-MAINTENANCE * DG-B-MAINTENANCE

Assuming that the plant technical specifications restrict both diesel generators from being in maintenance simultaneously while at power, this cut set is an example of mutually exclusive events. These mutually exclusive events would be handled by removing any cut sets that contain the mutually exclusive events (even if the cut set includes additional events).

Generally, mutually exclusive events fall under the "restriction case" (e.g., two AFW pumps out for testing at the same time) or under the "modeling concerns case" (e.g., initiating events in the fault trees lead to multiple initiating events in sequence cut sets).

20.3 Removing Mutually Exclusive Events

Mutually exclusive events are treated by removing any cut sets that contain the mutually exclusive events (even if the cut set includes additional events). In SAPHIRE, several methods exist that could perform this removal operation.

Editing the cut sets manually using the cut set editor to "weed-out' the mutually exclusive events.

Using the "mutually exclusive top event" feature when generating sequence logic so that impossible events can be defined at the sequence level.

Modify logic models (via NOT gates ore complemented events) to effectively remove impossible combinations of events.

Pruning cut sets to remove impossible events that are defined by a simple logic structure (i.e., fault tree).

Using recovery rules to define impossible combinations of events so that cut sets containing these events would be deleted.

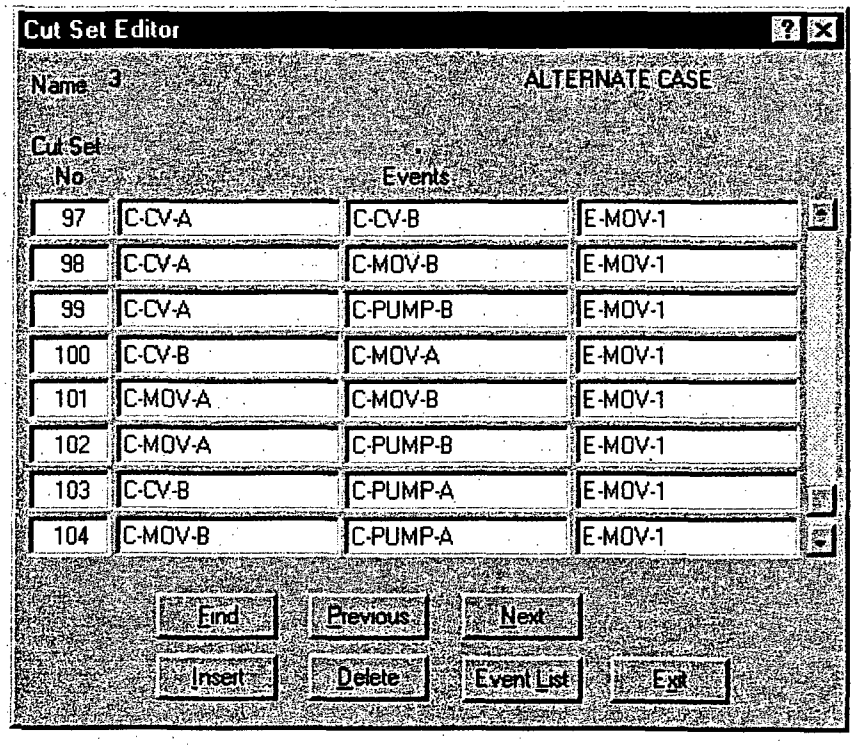
20.4 Manual Cut Set Editor Method

PURPOSE

Remove mutually exclusive events from a cut set using the cut set editor. This method can also be applied to fault tree cut sets in a similar manner.

STEPS

1. After sequences cut sets have been solved, select **Sequence** from SAPHIRE menu.
2. From the *Sequences* dialog, highlight the desired sequence. LOSP 3 was highlighted in this example.
3. Right-click to invoke the pop-up menu and select **Cut Sets | Edit | Current**.
4. Place the cursor in the *Cut Set No.* column of the desired cut set and choose the **Delete** button.



This process is not recommended since it is both error prone and time consuming.

20.5 Mutually Exclusive Top Method

PURPOSE

Remove mutually exclusive events from sequence cut sets using the "mutually exclusive top event" feature when generating sequence logic. Impossible events can be defined at the sequence level using this method. Using the "mutually exclusive top" method is only applicable to sequence cut sets.

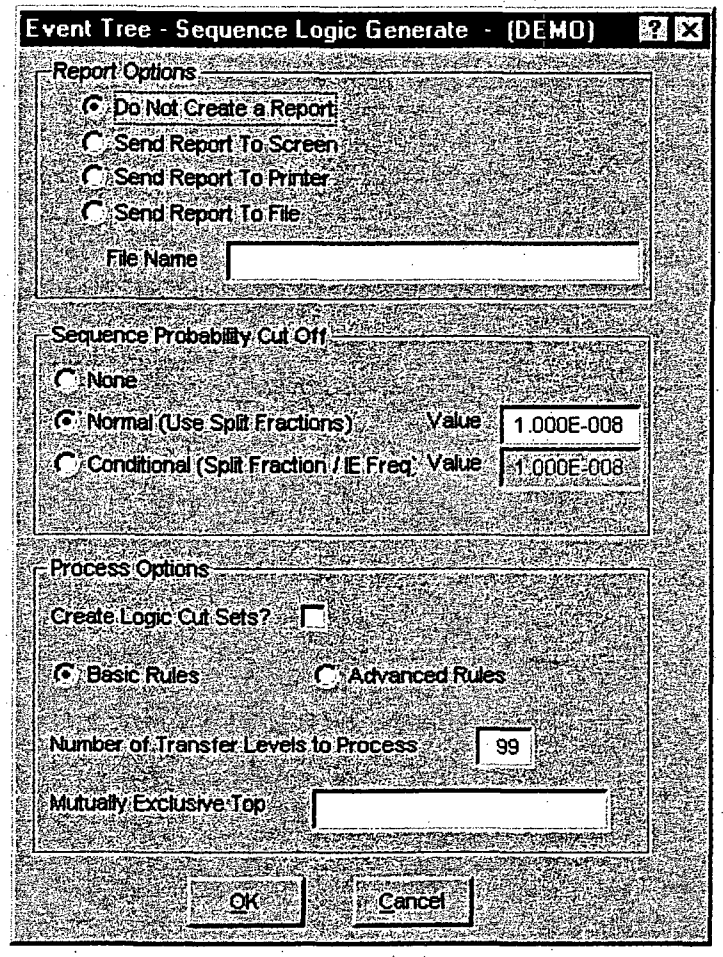
STEPS

1. Define fault tree logic that represents the combination of events that are mutually exclusive.

   Using the example, the analyst would make a fault tree that had the two maintenance events "ANDed" together.

   ME-TOP AND DG-A-MAINTENANCE DG-B-MAINTENANCE

2. The fault tree name for the fault tree containing the mutually exclusive combinations would then be typed in the space labeled "Mutually Exclusive Top Name" when generating sequence logic (via the **Event Tree | Link Trees** option).

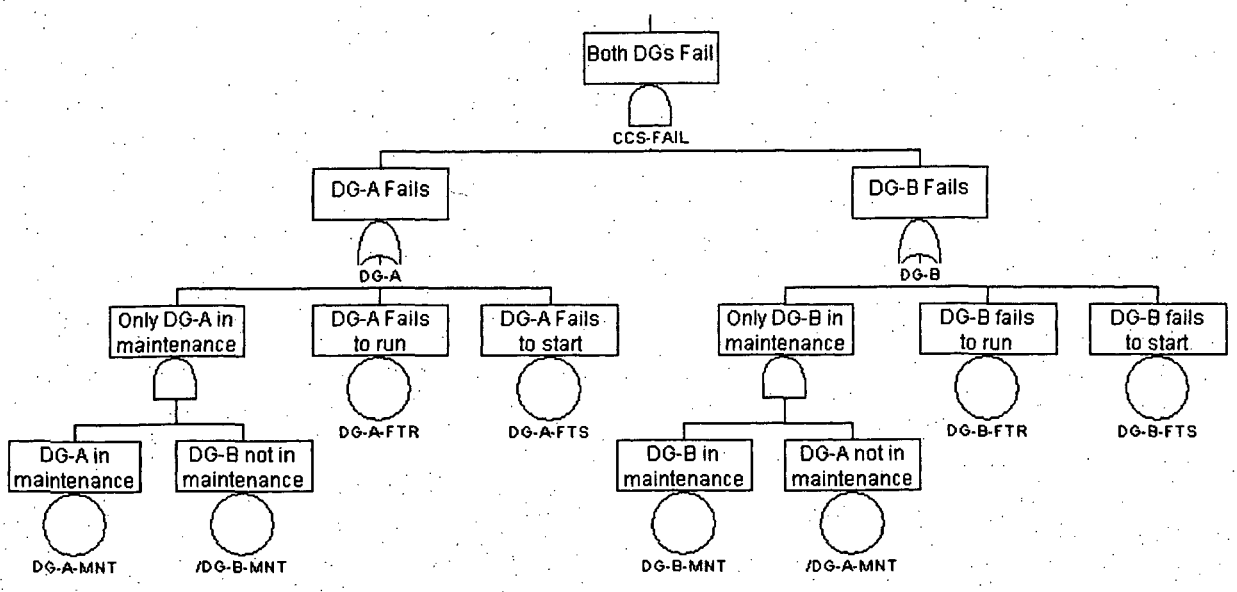


20.6 Logic Modification Method

PURPOSE

Remove mutually exclusive events by modifying logic models (via NOT gates or complemented events).

The illustration here demonstrates how the example problem could be modified so that combinations of diesel generator maintenance events will not appear in the list of cut sets.



Using this method requires that the analyst modify the fault tree logic in order to directly remove impossible combinations of events.

Drawbacks to this method include the time needed to modify the fault tree logic and the realization that complemented basic events will not appear in the list of cut sets.

20.7 Pruning Method

PURPOSE

Remove mutually exclusive events by pruning cut sets using the Pruning feature.

To use this method, the analyst must first define fault tree logic that indicates (via existing cut sets) combinations of events that you want to keep. Thus, the tree

ME-PRUNE OR DG-A-MAINTENANCE DG-B-MAINTENANCE

would delete cut sets containing DG-A-MAINTENANCE * DG-B-MAINTENANCE, but would keep any single occurrence of the events DG-A-MAINTENANCE or DG-B-MAINTENANCE.

20.8 Recovery Rules Method

PURPOSE

Use Recovery Rules to define groups of events that, if appearing together, result in the deletion of the cut set. For most cases, this is the preferred method of removing cut sets for the following reasons:

The rules are automatically implemented on the cut sets when recovery rules are applied.

No changes to logic models are needed.

No manual manipulations to cut sets are required.

The recovery rules for removing mutually exclusive events may be developed for a particular fault tree, all fault trees, a particular sequence, a single event tree, or all sequences.

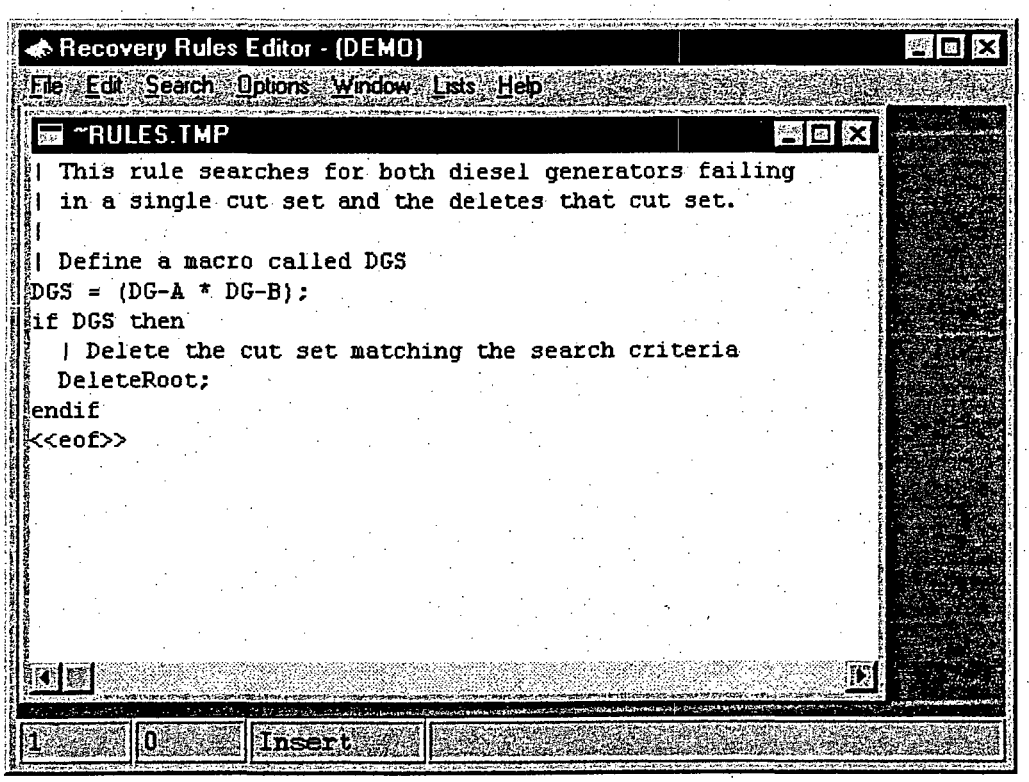
Mutually exclusive event rules are generally typed into the fault tree project and sequence project rules.

SEE ALSO

Recovery Rules Method Example

20.9 Recovery Rules Method Example

To demonstrate the potential use of the recovery rules, the example here shows how the rules could be used to remove the cut set containing both diesel generators failing from the demo database.



The rule shown above was entered in the sequence project rules.

The recovery rules were then applied to both sequences 2 and 3 (using no truncation).

Sequence 3 changed from an original value of 1.8E-3 to a value of 8.4E-4.

Only one cut set was removed from Sequence 3, but it happened to be the dominant cut set.

21. INTERNATIONAL SAPHIRE

21.1 Setup of International SAPHIRE

This section applies to the English versions of the Windows operating systems. It is intended to guide you through the steps of enabling foreign language support while operating the *SAPHIRE for Windows* program. If your Windows operating system is the native version for the foreign language of your choice, it is not necessary for you to follow these instructions.

The following instructions are applicable only for the specified versions Windows 95 and Windows NT operating systems.

Because you will be required to restart your system, it is recommended that you close all other applications before beginning installation of the foreign language(s). You will also need to have your Windows operating system CD-ROM or floppy disks available.

21.2 Windows NT, Version 4.0

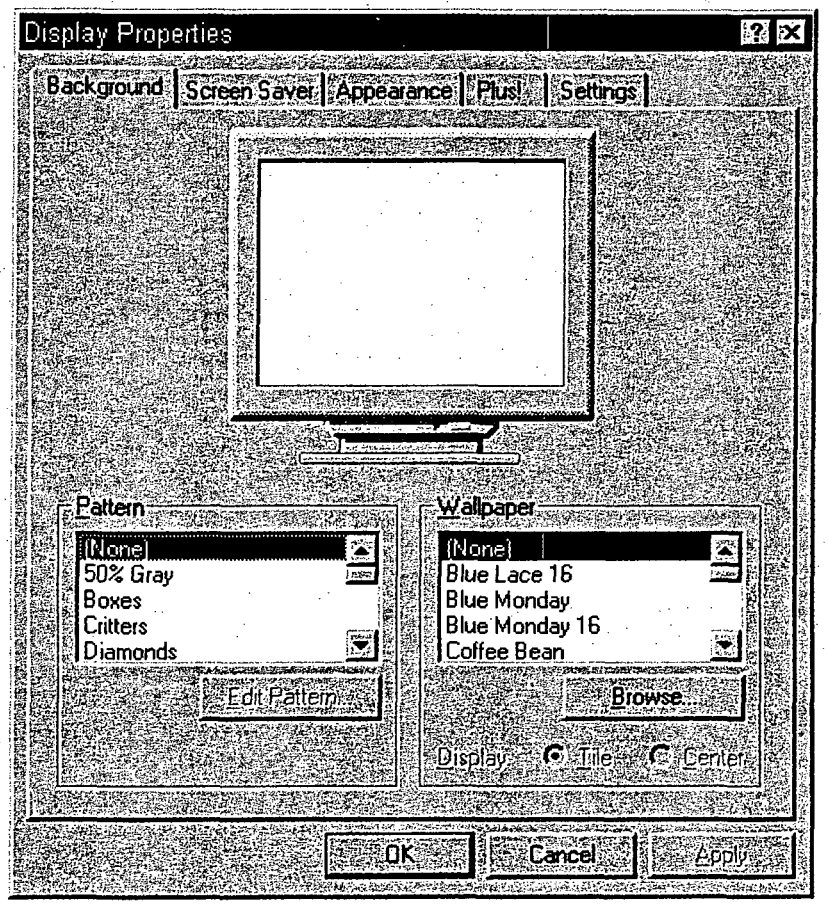
PURPOSE

This section guides you through the steps of enabling foreign language support for the Windows NT, Version 4.0 operating system.

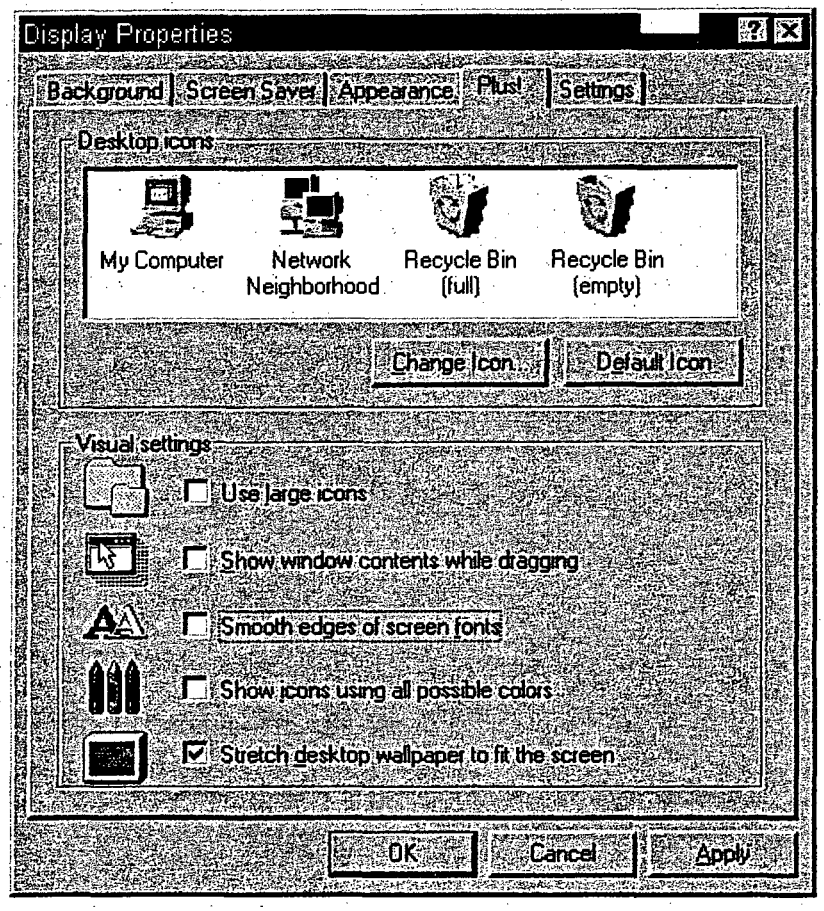
SAPHIRE International Support is not available for earlier versions of Windows NT (e.g., Version 3.51 or earlier). SAPHIRE has been tested only with NT 4.0, Service Pack 3, but should run with prior versions of NT 4.0. It is strongly recommended that you run SAPHIRE with the most current NT Service Pack.

STEPS

1. Double-click on the *My Computer*  icon on your desktop. In the *My Computer* window double-click on the *Control Panel*  folder. In the *Control Panel* window double-click on the *Display*  icon. The *Display Properties* dialog will be displayed.

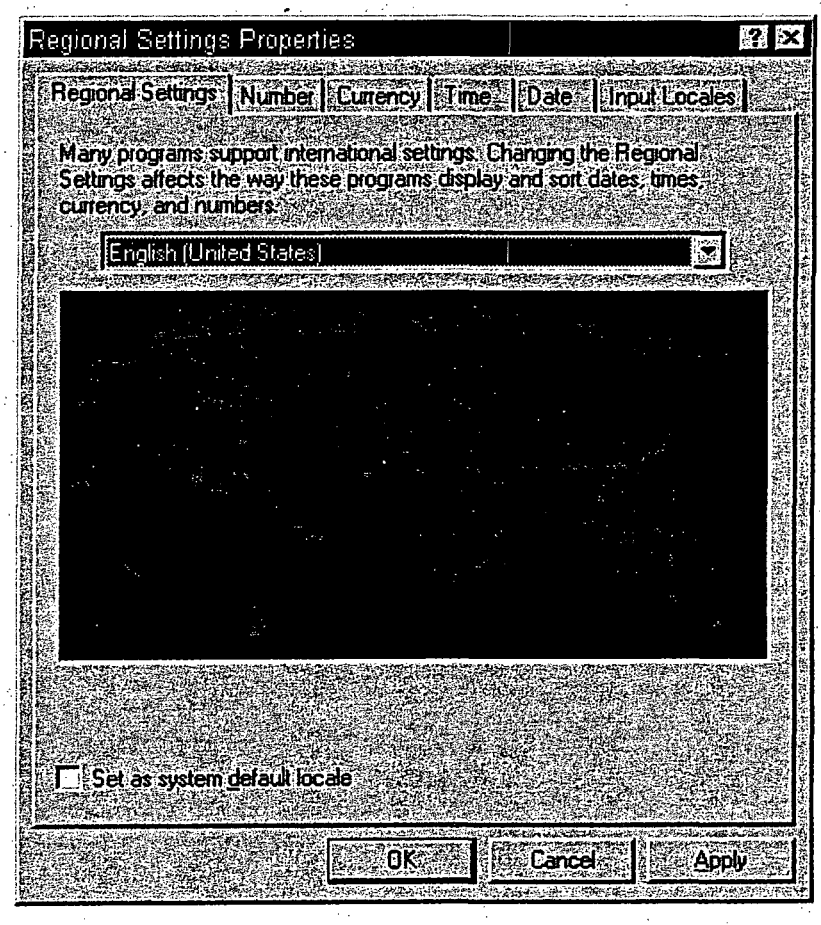


2. Select the *Plus!* tab.

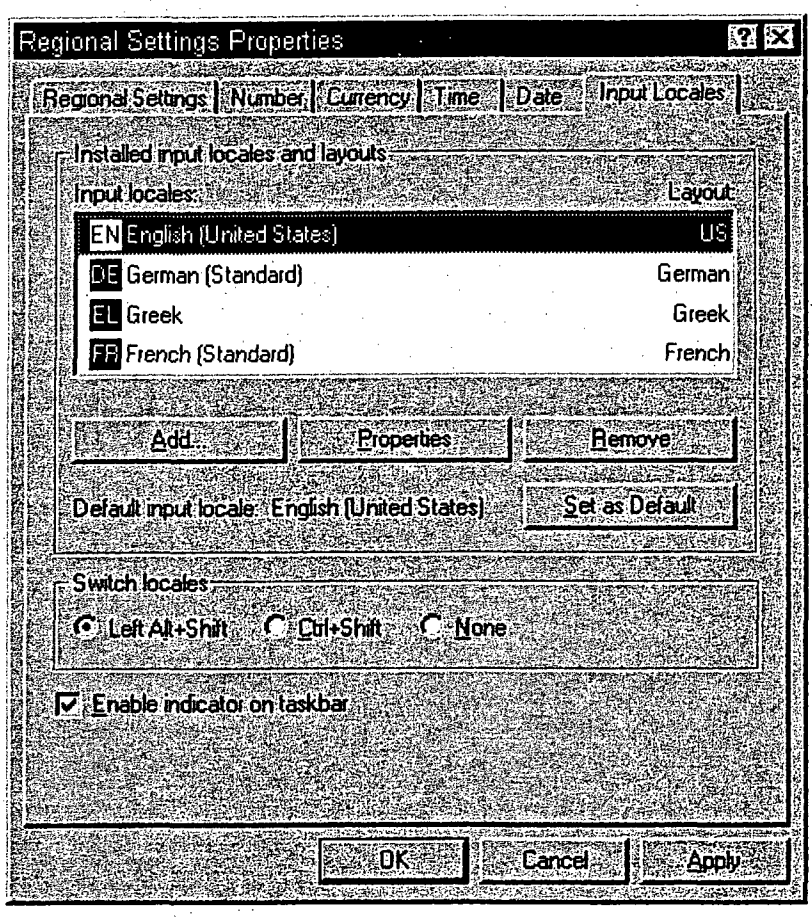


3. Ensure that the **Smooth edges of screen fonts** option is ***NOT*** checked. Choose the **Apply** button. Choose the **OK** button. The *Display Properties* dialog will be closed.

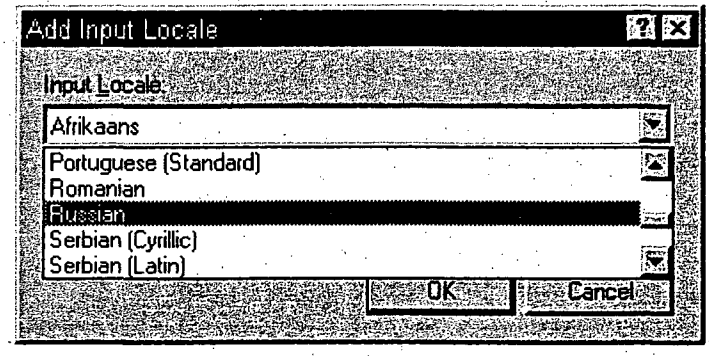
4. In the *Control Panel* window double-click on the *Regional Settings* icon. The *Regional Settings Properties* dialog will be displayed.



5. Choose the **Input Locales** tab. The currently installed input locales will be displayed in the list. If the language you wish to use is already in the list, skip to Step 7. If the language you wish to use is not in the list, choose the **Add** button.



6. The *Add Input Locale* dialog will be displayed. Select the desired locale from the drop-down list and choose the **OK** button.

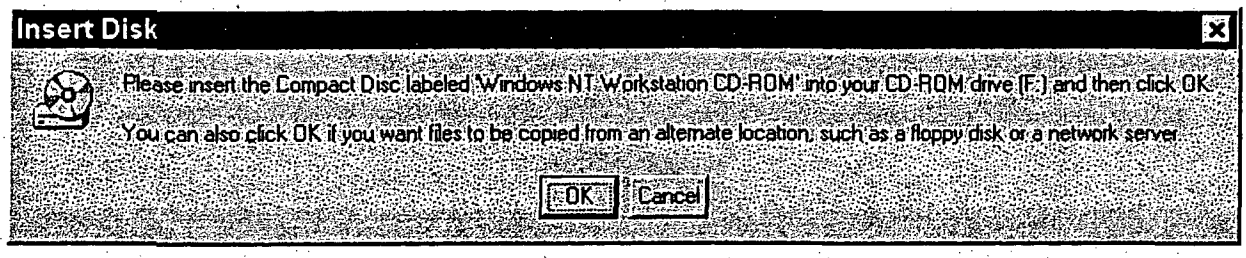


The *Add Input Locale* dialog will be closed and the selected locale will be displayed in the list of input locale choices on the *Regional Settings Properties* dialog, *Input Locales* tab.

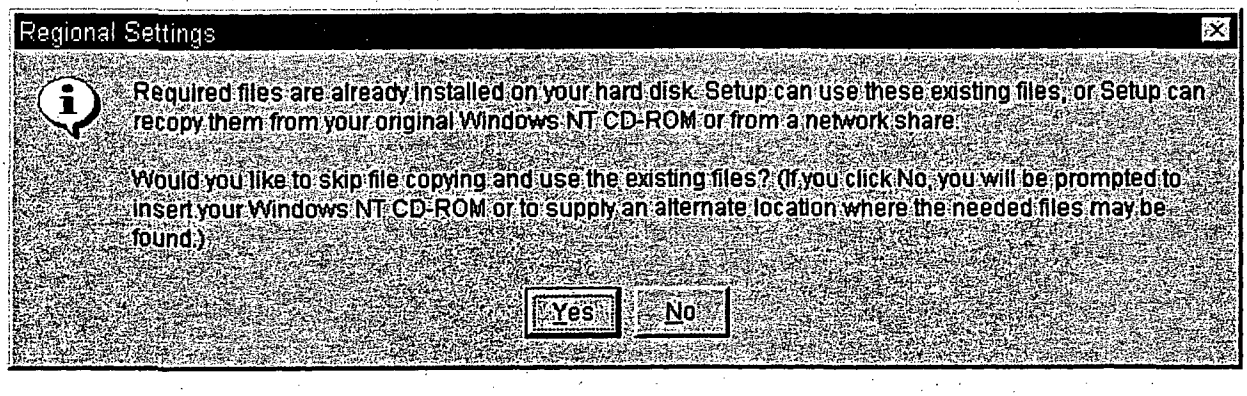
7. To allow switching on-the-fly between installed input locales, choose either the **Left Alt+Shift** or **Ctfl+Shift** radio buttons. Ensure that the **Enable indicator on taskbar** check box is checked. Choose the **Apply** button. You may be instructed to install some required files on your computer. If necessary, follow the directions for copying the requisite files.

8. On the *Regional Settings Properties* dialog choose the **Regional Settings** tab. Select the desired language from the drop-down list. Choose the **Set as system default locale** check box. Choose the **OK** button.

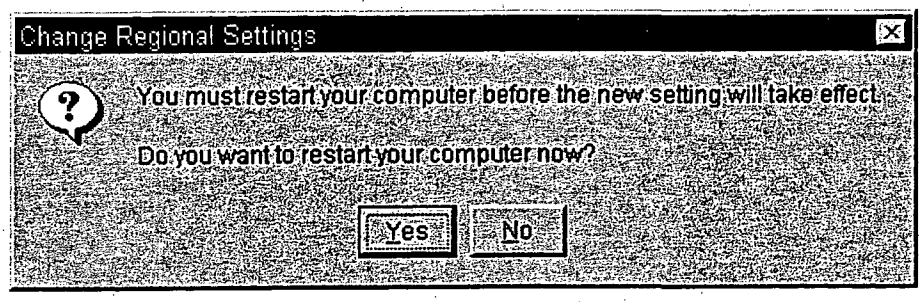
9. If you have not previously installed the required files, the *Insert Disk* dialog will be displayed. Follow the directions and choose the **OK** button.



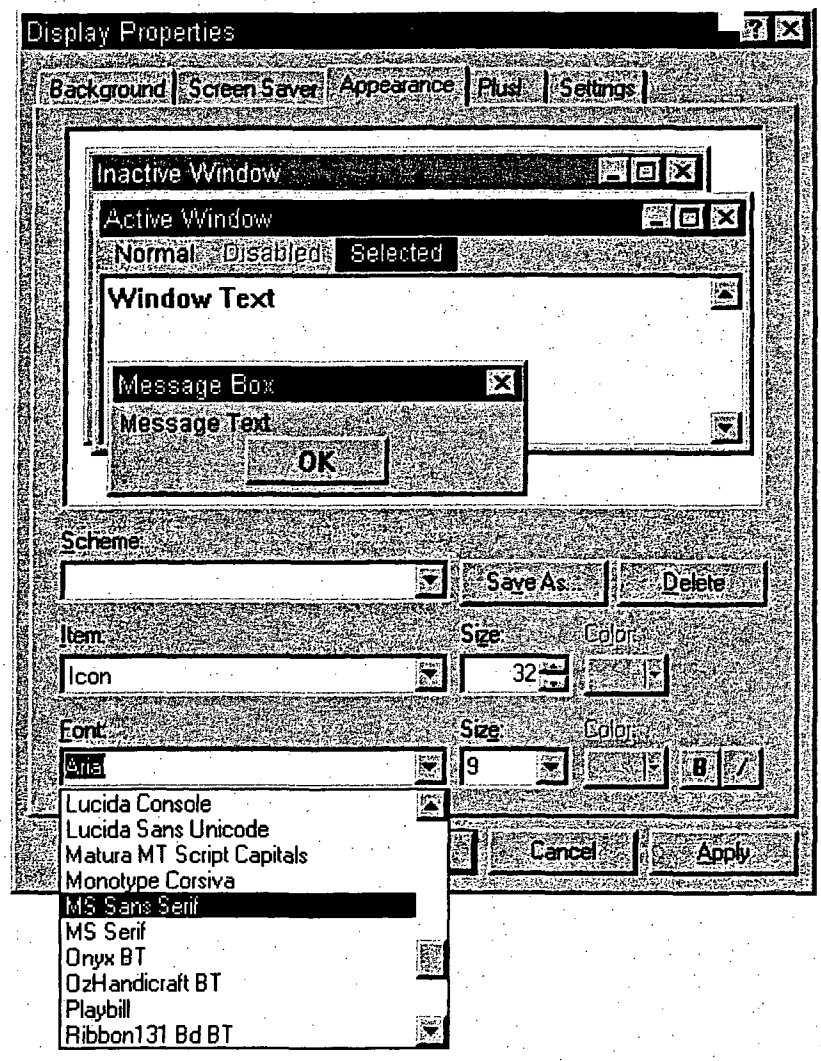
10. If you have previously installed the required language files, the *Regional Settings* dialog will be displayed. Choose the **Yes** button.



11. The *Change Regional Settings* dialog will be displayed. In order for the change to take place, you must restart your computer.



After your computer has restarted, you will need to change your system's "icon font." Open the *Display Properties* dialog. Select the *Appearance* tab.



12. In the *Item* drop-down list, select the **Icon** choice. In the *Font* drop-down list, select the **MS San Serif** font. Choose the **Apply** button. Choose the **OK** button. The *Display Properties* dialog will be closed.

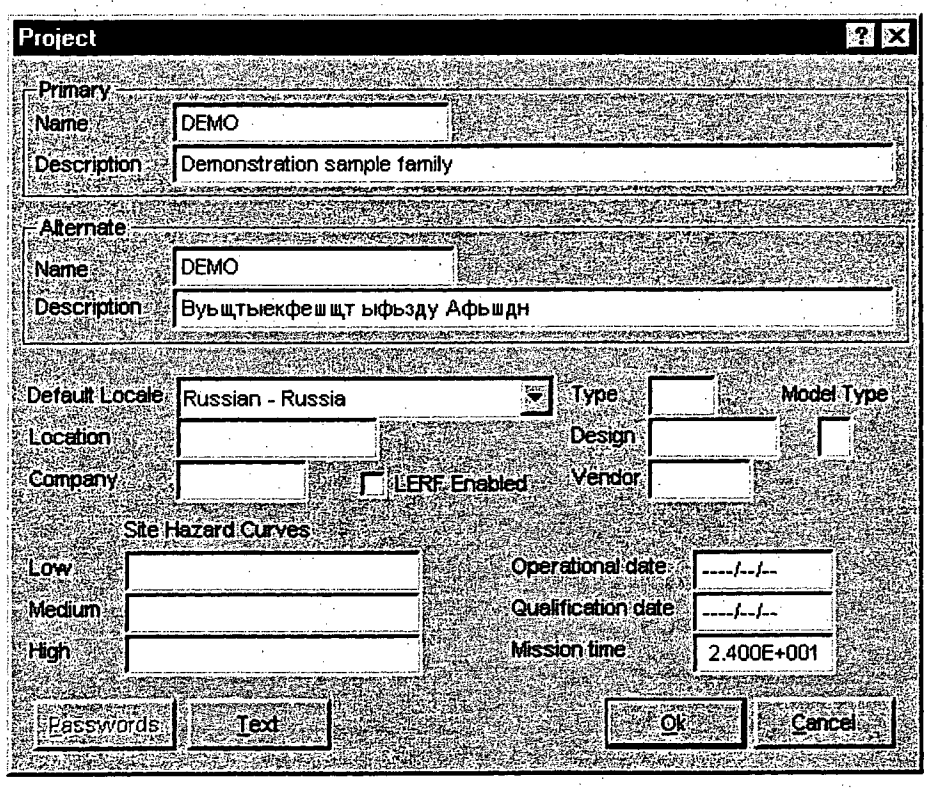
13. You may want to change the regional setting of your operating system's "current" locale to English, but leave the "default" locale set to Russian. On the *Regional Settings Properties* dialog, *Regional Settings* tab select *English* from the drop-down list. Choose the **Apply** button, then the **OK** button. By doing this you set the "current" locale to English which enables SAPHIRE to display certain controls using the English alphabet. **DO NOT** select the **Set as system default locale** check box since by doing this you will change the operating system's "default" locale from Russian back to English.

    You can switch between languages (code pages) anytime after installing more than one locale by using the shortcut keys selected in Step 7, above, or by clicking on the language indicator on the system taskbar and selecting the desired locale from the list.

To enter data using the English character set, you must choose the *English* locale, to enter data using the Cyrillic character set, you must choose the *Russian* locale, etc.

This on-the-fly switching of code pages does not affect the way programs display dialog controls, times, dates, etc. It simply changes the way keyboard input is interpreted by the program, allowing entry using different character sets.

14. Start SAPHIRE and choose **Modify | Project** from the menu or choose the  buttons on the toolbar. The *Project* dialog will be displayed.



15. Choose the desired language from the *Default Locale* drop-down list. Choose the **Ok** button. You can now enter and view foreign language characters on SAPHIRE dialogs and reports. Switch between locales (described in Step 13) to enter data using the different character sets on subsequent SAPHIRE dialogs.

21.3 Windows 95, Version 4.00.950

PURPOSE

SAPHIRE does not run with this version of Windows 95. You **MUST** install Service Pack 1 for this version of Windows 95 in order to run SAPHIRE and use any language other than English.

After installing Service Pack 1, you will be running Windows 95, version 4.00.950a (Service Pack 1). Follow the instructions below for Windows 95 version 4.00.950 b.

21.4 Windows 95, OEM Version 4.00.950 b (OSR2)

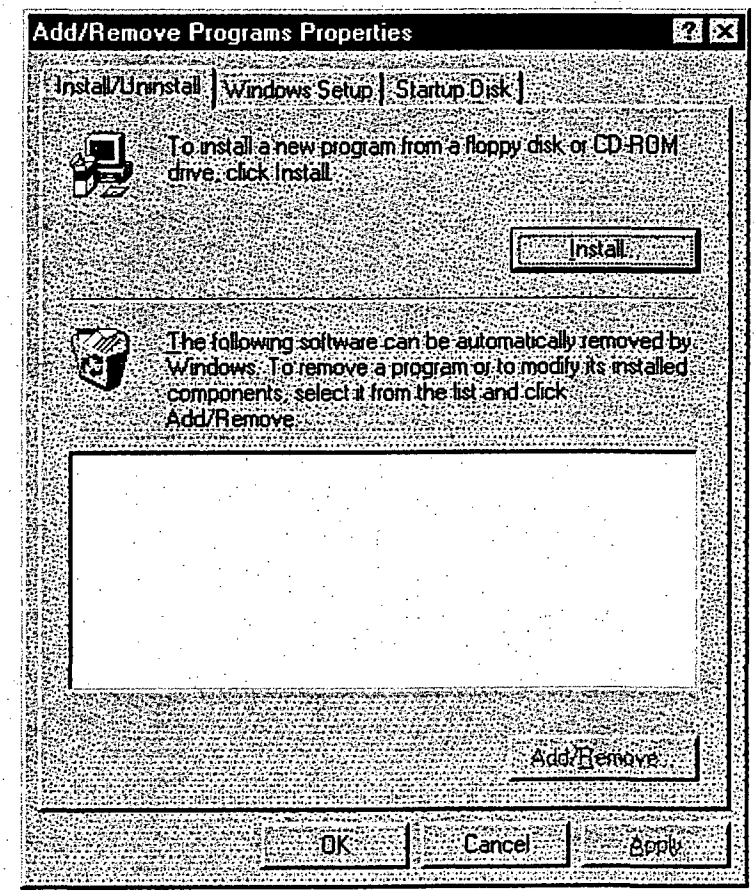
PURPOSE

This section guides you through the steps of enabling foreign language support for the Windows 95, OEM Version 4.00950 b (OSR2) operating system.

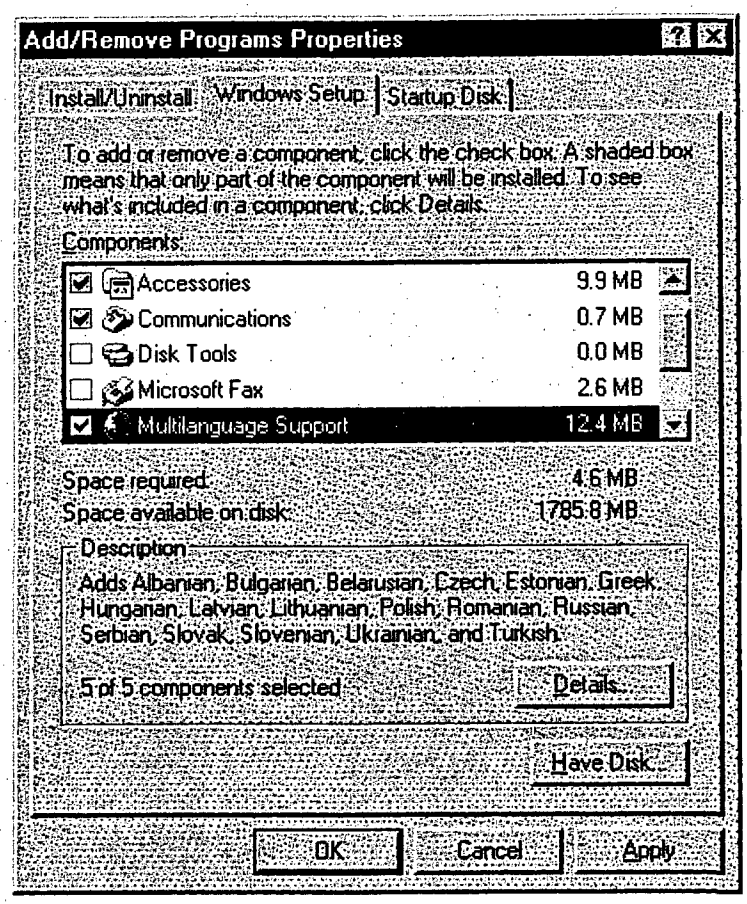
STEPS

1. Double-click on the *My Computer*  icon on your desktop. In the *My Computer* window double-click on the *Control Panel*  folder. If you have previously installed Multilanguage support, skip to Step 6, otherwise, continue with Step 2.

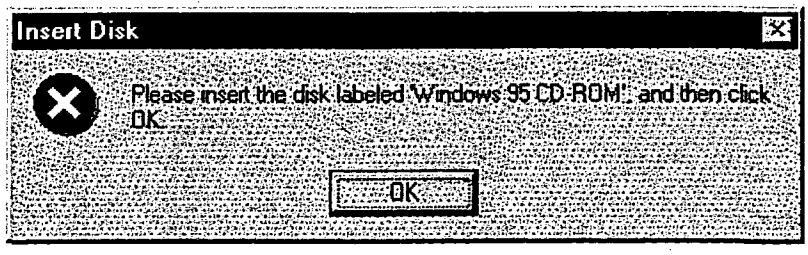
2. In the *Control Panel* window double-click on the *Add/Remove Programs*  icon. The *Add/Remove Programs Properties* dialog will be displayed.



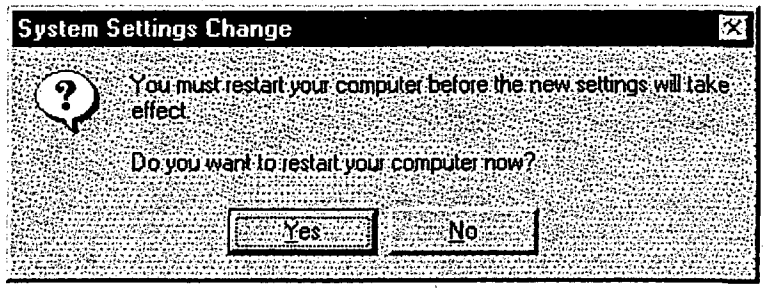
3. Select the *Windows Setup* tab. From the *Components* list check the **Multilanguage Support** option. Choose the **Details** button and ensure that a check mark appears next to the language(s) you want to use. Choose the **Apply** button.



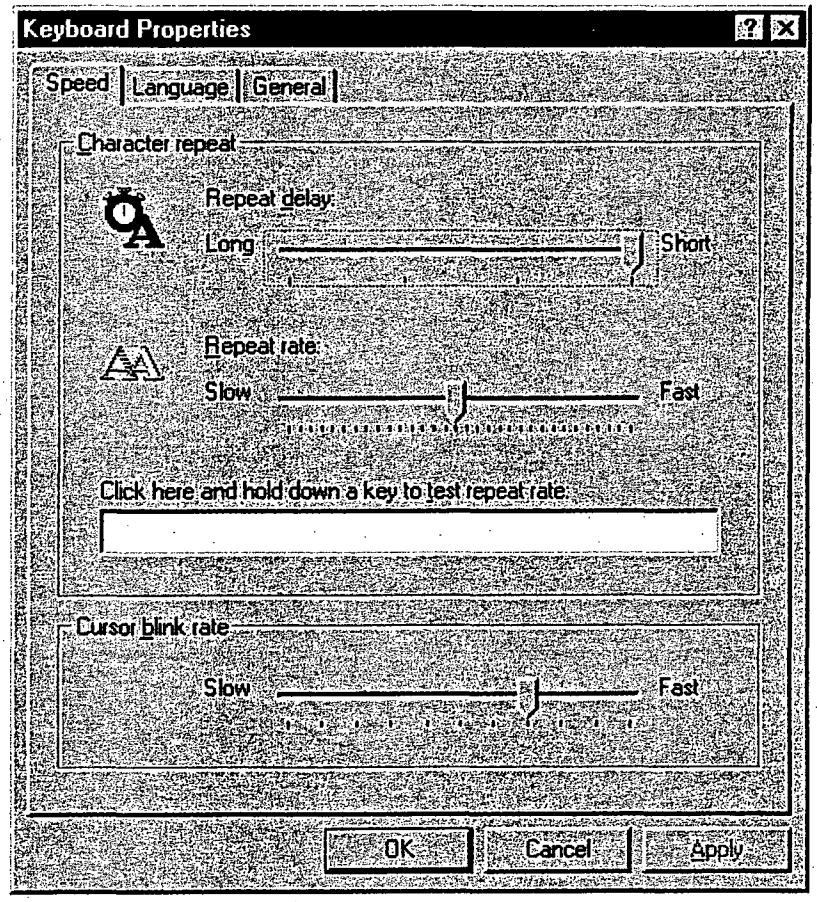
4. If the required files have not been previously installed, the *Insert Disk* dialog will be displayed. Follow the directions and choose the **OK** button.



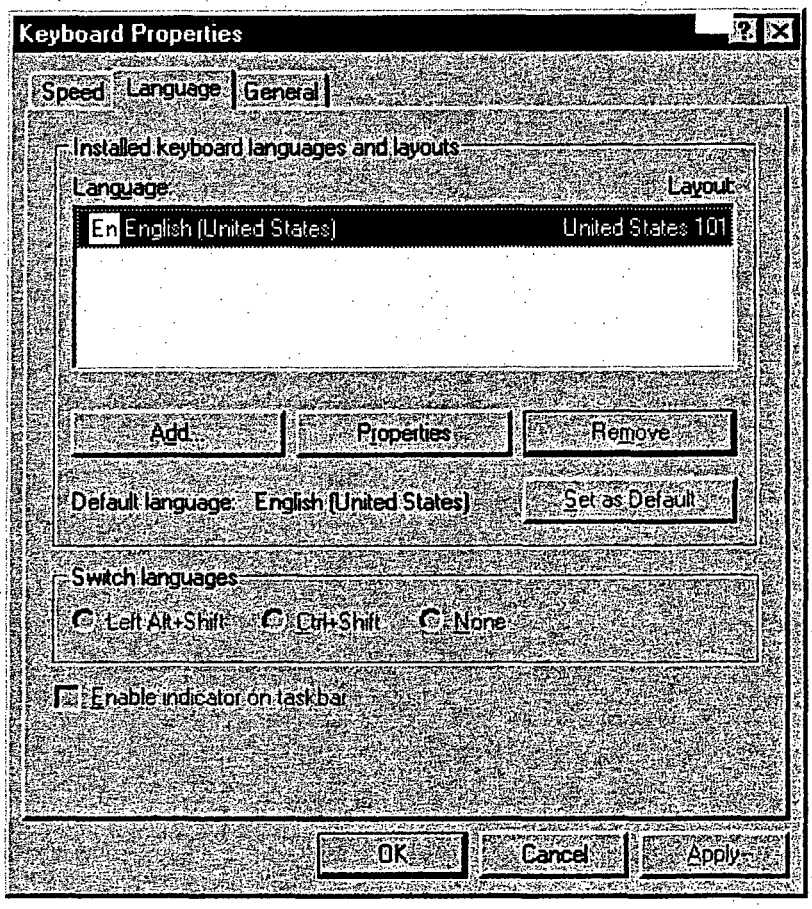
5. After the required files have been copied, the *System Settings Change* dialog will be displayed. Choose the **Yes** button. In order for the change to take place, you must restart your computer



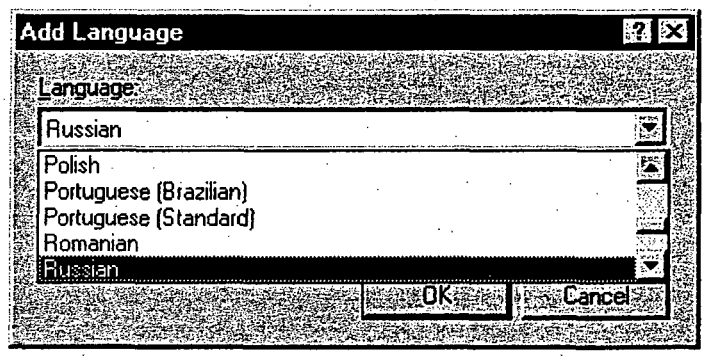
6. After your computer has restarted, you will need to change your system's keyboard properties. In the *Control Panel* window double-click on the *Keyboard* icon. The *Keyboard Properties* dialog will be displayed.



7. Choose the **Language** tab. The currently installed languages are displayed in the list. If the language you wish to use is already in the list, skip to Step 9. If the desired language is not in the list, choose the **Add** button.

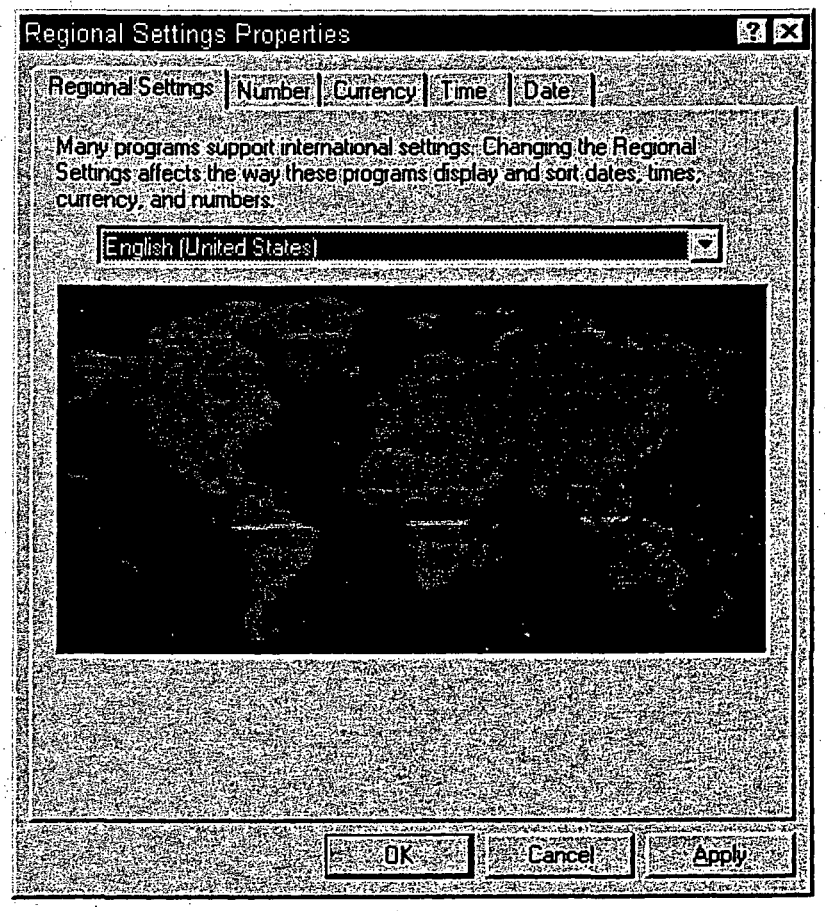


8. The *Add Language* dialog will be displayed. Select the desired language from the drop-down list and choose the **OK** button.



9. The *Add Language* dialog will be closed and the selected language will be displayed in the list of language choices on the *Keyboard Properties* dialog, *Language* tab (refer to Figure 17). To allow for switching on-the-fly between installed languages, choose either the **Left Alt+Shift** or **Ctrl+Shift** radio buttons. Ensure that the **Enable indicator on taskbar** check box is checked. Choose the **Apply** button then choose the **OK** button. You may be instructed to install some required files on your computer. If necessary, follow the directions for copying the requisite files.

10. In the *Control Panel* window double-click on the *Regional Settings* icon. The *Regional Settings Properties* dialog will be displayed.



11. Select the desired language from the drop-down list. Choose the **Apply** button.
12. The *Change Regional Settings* dialog will be displayed. In order for the change to take place, you must restart your computer.
13. After your computer has restarted, you can switch between languages by using the shortcut keys selected in Step 9 above, or by clicking on the language indicator on the system taskbar and selecting the desired language from the list.
14. Go to Steps 14 and 15 under the Windows NT instructions.

21.5 MAR-D Format

When editing MAR-D output files that contain data entered using other languages, you must use an editor that supports multiple languages (i.e., language scripts other than English) such as Multi-Edit for Windows and WordPad.

21.6 Definitions

21.6.1 MAR-D Format

When editing MAR-D output files that contain data entered using other languages, you must use an editor that supports multiple languages (i.e., language scripts other than English) such as Multi-Edit for Windows and WordPad.

21.6.2 OEM

Original Equipment Manufacturer

21.6.3 OSR2

Operating System Revision 2

21.6.4 Code Page

An internal table that the operating system uses to relate the keys on the keyboard to the characters displayed on the screen. Usually defined to support specific languages or groups of languages which share common writing systems. In Windows 95 and NT 4.0, code pages can be changed on-the-fly by the user without changing the default language system in use.

21.6.5 Font

A set of graphical representations of characters.

21.6.6 Keyboard layout

A standard arrangement of characters on a keyboard that defines which keys produce particular characters.

21.6.7 Locale

The features of the software environment that are dependent on language, country, and cultural conventions (e.g., sort order; keyboard layout; character sets; and time, date, and currency formats). In Windows, the locale information includes the name of the spoken language, the script used to write the language, and the cultural conventions.

21.6.8 Script

A system of characters used to write one or several languages. The characters of the script denote sounds, syllables, or word elements and are governed by a general set of rules for creating text, such as default writing direction.

22. MACRO SCRIPTS

22.1 Macro-Scripts Usage And Examples

SAPHIRE (Version 7) is equipped with a feature that allows for the usage of Macro-scripts to run specified menu or analysis steps in a manner outside of SAPHIRE. This feature is useful for repetitive analysis procedures in databases or several databases. Macro-scripts in SAPHIRE are different than the Macros used in the linking, recovery and partition rule editors. Those Macros are used to define a combination of rule based operands, systems, or basic event for linking event trees or adding recovery actions or end state partitions. "Macro-scripts" are Keywords in text file format that SAPHIRE uses to automatically perform "analysis menu" functions. Though not directly designed to perform model modifications (i.e. modify basic event data, fault tree logic and event tree logic). Most modify functions can be used using macro-scripts using the MAR-D extract and load feature.

Macro-scripts are "programmed" using a standard text editor that SAPHIRE interprets to perform specified functions. The Macro-scripts are coded with *Keywords* that SAPHIRE recognizes as a specific menu or analysis functions.

The embedded macro code uses tags, similar to those in an "HTML" or "XML" type file to run the actual associated scripts. The tags are classified as verbs or "actions" which are the main actions or operations performed on a plant model; classes of "objects" which are acted upon by the verbs; and "parameters" which are passed to the actions. Each action, class, and parameter has opening and closing tags that mark the series of commands performed during a macro scenario. This code contains the declarations for all the tags utilized by the macrocode. The module also contains functions and procedures that parse the macro script into its actions and verbs, initializes the routines and parameters in the tool from the macro specified inputs, runs the code and outputs results into a text file. The macro scripts themselves can be run in normal or debug mode. All the commands executed can be submitted into an output ASCII text log file. Results are output into a separate ASCII text file. Time, date, and user identification information is logged with the test results. Key modules within the API and core functional areas have had macrocode hooks added to support this interface. Comment capability is available in the Macro-script program.

TOPICS

Macro-Script Keywords

The following table illustrates the Keywords and their usage for creating a macro-script in SAPHIRE. Each keyword having a opening tag and closing tag (i.e. <tag> being the opening and </tag> being the closing). The table also contains parameter setting commands that are used by the tags to perform specific functions.

Keyword	Use/Example	Notes
<add>	<add> . . . </add>	This keyword will add a new item to a list. Using this keyword, a basic event could be added to a MAR-D cut set listing or a change set added to a database.
<analysis type>	<analysis type>4 </analysis type>	This keyword is used to specify analysis type that SAPHIRE is to perform calculation in. Random = 1, Fire = 2, Flood = 3, Seismic = 4, etc. For this example the analysis type would be "seismic".
<analysis>	<analysis>random </analysis>	This keyword is used to specify specific analysis type during a base case update. For this example, during the base case update, the analysis type would be "random."
<base case update>	<base case update> (analysis type specified) </base case update>	This keyword is used to specify the start and end a base case update action.
<basic event>	<basic event> . . . </basic event>	This keyword is used to specify the start and end of a basic event action. Generally, basic event actions involves extracting or loading MAR-D files or adding/deleting a basic event from a MAR-D file.
<calc type>	<calc type>1</calc type>	This keyword is used to specify the calculation type. For this example, calculation type 1 would be use in the change set.
<case>	<case>base</case>	This keyword is used to specify the base or current case. For this example, the base case would be invoked.
<change set>	<change set> . . . </change set>	This keyword is used to specify the start and end of change set operations. For example, a change set can be added, deleted, marked or unmarked.
<class>	<class> . . . </class>	This keyword is used to specify the start and end of an event class operation (i.e., class type basic event change in a change set). Within this keyword function, the class of the basic events to be changed and their desired changes is specified.
<comment>	<comment> information </comment>	This keyword is used to specify the start and end of a macro comment block. Any text can be located within the comment block.
<compare file>	<compare file> <input 1>*FILE1* </input1> <input 2>*FILE2* </input2>	This keyword is used to specify the start and end of a file comparison. Used with <input 1>, <input 2> tags to specify the two files to compare (report files). The results of the comparison can be viewed in a file using the <output>*filename*</output> keyword.

Keyword	Use/Example	Notes
	</compare file>	
<condition>	<condition> . . . </condition>	This keyword is used to specify the start and end of a selected GEM condition assessment process. Within the condition block, a GEM condition assessment can be added, deleted, marked or unmarked.
<delete>	<delete> . . . </delete>	This keyword is used to specify the start and end of a list item to be deleted. Using this keyword, a basic event could be deleted to a MAR-D cut set listing and then reloaded or a change set deleted from a database.
<description>	<description> text description </description>	This keyword is used to specify a text description to change set, flag set, or scenario. For example if a change set is created, <description>`Class change - All events`</description>, the text between the description block would be added as the change set's description.
<duration>	<duration>72 </duration>	This keyword is used to specify a duration time for a GEM condition assessment. For this example, the duration time would be 72 hours.
<end state>	<end state> . . . </end state>	This keyword is used to specify the start and end of an end state operation. Within the end state keyword blocks, end states can be marked, gathered, solved, and an uncertainty analysis performed.
<end>	<end> </end>	This keyword is used to specify the end of a scenario.
<event name>	<event name> BE NAME </event name>	This keyword is used to specify an event name to be adjusted during a change set operation. For this example, the basic event BE NAME, could be adjusted within a change set block.
<event tree>	<event tree> . . . </event tree>	This keyword is used to specify the start and end of an event tree operation. Within the event tree keyword block, an event tree can be marked, unmarked, and linked.
<exclude>	<exclude> FT-1 </exclude>	This keyword is used to specify the name of a event tree, fault tree, or sequence to be excluded in an operation. For this example, fault tree FT-1, would be excluded from any operation.
<extract>	<extract> . . . </extract>	This keyword is used to specify the start and end of a MAR-D extract operation. Within the extract keyword block, the MAR-D file type and file name is specified.

Keyword	Use/Example	Notes
<fault tree>	<fault tree> . . . </fault tree>	This keyword is used to specify the start and end of a fault tree operation. Within the fault tree keyword block, cut sets can be solved, flag sets applied, uncertainty analysis and importance measures can be performed.
<file name>	<file name> TEMP.BEI </file name>	This keyword is used to specify a file name for MAR-D extract and load functions. For this example, the file name is MAR-D file with a name "TEMP.BEI". This keyword is also used to specify report output files names.
<flag set>	<flag set> . . . </flag set>	This keyword is used to specify the start and end of flag set operations. For example, a flag set can be added, deleted, and applied to fault trees and sequences.
<gather method>	<gather method>by sequence</gather method>	This keyword is used to specify the specific end state gather operation. For this example, the cut sets would be gathered "by sequence."
<generate>	<generate> </generate>	This keyword is used to specify the generate option. The generate command implements the changes in a change set or flag set and moves from the base case to the current case for an analysis.
<group>	<group>yes</group>	This keyword is used to specify the use of group importance measures or not. For this example, the group importance is "on".
<ie name>	<ie name>IE-LOSP </ie name>	This keyword is used to specify the initiating event name. For this example, the initiating event "IE-LOSP" would be selected during a GEM initiating event assessment.
<importance>	<importance> . . . </importance>	This keyword is used to specify the start and end of an importance measure operation.
<include>	<include>FT-1 </include>	This keyword is used to specify the name of a event tree, fault tree, or sequence to be included in an operation. For this example, fault tree FT-1, would be included in the operation.
<init event>	<init event> . . . </init event>	This keyword is used to specify the start and end of a GEM initiating event assessment..
<initial prompt>	<initial prompt> yes </initial prompt>	This keyword is not necessary to run a macro. However, if "yes" is specified (as in this example) in the initial prompt block, a confirmation box is displayed that ask the analyst do confirm the operation of the macro-script.

Keyword	Use/Example	Notes
<input 1>	<input 1> FILENAME1</input 1>	This keyword is used to designate the first file in a compare operation. For this example, the first file is called FILENAME1.
<input 2>	<input 2> FILENAME2</input2>	This keyword is used to designate the second file in a compare operation. For this example, the second file is called FILENAME2.
<lambda>	<lambda> 1.0E-2 </lambda>	This keyword is used to specify the value. For this example, the lambda value would be 1.0E-2.
<link>	<link> </link>	This keyword is used to link event trees.
<load>	<load> . . . </load>	This keyword is used to specify the start and end of a MAR-D load operation. This operation can load in additional MAR-D files or files that have been modified
<logic show save>	<logic show save> </logic show save>	This keyword is used to show and save fault tree logic .
<loop class>	<loop class> grid </loop class>	This keyword is used to specify the type of loop classification being used during an analysis. For this example, the loop class would be "Grid-related".
<mark event tree mask>	<mark event tree mask> LOSP </mark event tree mask>	This keyword is used to specify the specific event tree mask and is used during sequence analysis. For this example, all sequences generated with the event tree LOSP would be marked.
<mark logic fault tree>	<mark logic fault tree>*</mark logic fault tree>	This keyword is used to select a specific sequence having a specific fault tree logic during sequence analysis.
<mark mask>	<mark mask> * </mark mask>	This keyword is used to specify the specific fault trees or event trees to mark. For this example, a "*" would mark all fault trees or event trees in the list (depending on if you were functioning under a <fault tree> or <event tree> function block).
<mark name>	<mark name> NAME1</mark name>	This keyword is used to specify the specific condition name in GEM. For this example, the condition assessment NAME1 would be marked.
<mark sequence mask>	<mark sequence mask> ATWS </mark sequence mask>	This keyword is used to specify the specific sequence mask. For this example, all ATWS sequences would be selected for analysis.

Keyword	Use/Example	Notes
<mark sequence name>	<mark sequence name> LOSP2 </mark sequence name>	This keyword is used to mark a specific sequence for sequence analysis. For this example, the sequence named LOSP2 would be marked..
<mask operation>	<mask operation>and </mask operation>	This keyword is used to specify the specific mask operation for masking/marking sequences. The logic mask operator can either be "OR" or "AND". This function block allows more flexibility in selecting event tree names, sequence names and fault tree names for sequence analysis.
<mask>	<mask> </mask>	This keyword is used in conjunction with others.
<method>	<method> mcs </method>	This keyword is used to specify the specific uncertainty method. For this example, the uncertainty method would be Monte Carlo. For Latin Hypercube, "lhs" would be used.
<mission time>	<mission time> 10 </mission time>	This keyword is used to specify the specific mission time value. Mission time can be in scientific notation i.e.<mission time>1.0E+1</mission time>. For this example the mission time would be 10 hours.
<name>	<name> test name </name>	This keyword is used to specify the name of a specific test passed in by a DOS script file, i.e. <name>%P-11</name>. It is also used to specify the name of a change set, flag set, fault tree, event tree, etc to be added or deleted during an operation.
<output>	<output> file name </output>	This keyword is used to specify the output to a specific file name. For example, <output>compare.rpt</output>, would output to a text file called "compare.rpt".
<page>	<page> </page>	This keyword is used to turn on the MAR-D paging option.
<partition>	<partition> </partition>	This keyword is used to turn on the partition option for event trees.
<probability>	<probability> 0.02 </probability>	This keyword is used to specify a specific probability. For this example the probability of 0.02 would be used in a change set.
<process>	<process> </process>	This keyword is used to specify the start and end of a condition assessment timed test.
<program exit>	<program exit> </program exit>	This keyword is used to specify the end of the macro script.
<project>	<project> </project>	This keyword is used to specify a "project level" uncertainty

Keyword	Use/Example	Notes
		operation.
<report destination>	<report destination>F </report destination>	This keyword is used to specify the destination of the report output. There are three options: "F" is a file, "V" is a screen view, and "P" is a default printer. It the report destination is not specified, the default is to a file.
<report format>	<report format>A </report format>	This keyword is used to specify the type of report format that is outputted. There are three options: "A" is text file, "R" is RTF format, and "H" is HTML format. The default is text format.
<report>	<report> . . . </report>	This keyword is used to specify a report creating operation.
<sample>	<sample> 5000 </sample>	This keyword is used to specify the number of samples in an uncertainty calculation. For this example, 5000 samples would be ran for the uncertainty operation.
<scenario>	<scenario> . . . </scenario>	This keyword is used to specify a specific scenario sequence of events. This keyword is used to help identify various macro-script functions in a file.
<seed>	<seed>1234</seed>	This keyword is used to specify the random number seed in an uncertainty measure calculation. If a "0" is used, then the clock number is implemented. For this example, the seed number would be "1234".
<seismic bin>	<seismic bin> 9 </seismic bin>	This keyword is used to specify the histogram bin number for a seismic analysis.
<sequence>	<sequence> . . . </sequence>	This keyword is used to specify the start and end of a sequence operation.
<set units>	<set units> PPerYear </set units>	This keyword is used to specify the project units. For this example "per year" would be used. Other units include per month, per week, and per day.
<setup>	<setup> . . . </setup>	This keyword is used to specify the test scenario state of operation. This keyword is used to during the validation and verification macro-script tests.
<show save>	<show save> </show save>	This keyword displays a fault tree graphic and saves it.

Keyword	Use/Example	Notes
<single>	<single> </single>	This keyword is used to specify a single event operation in a change set operation. To types of change sets are available, single and class changes.
<slice save>	<slice save> . . . </slice save>	This keyword is used to specify a cut set slice save operation. This feature allow for specific cut sets to be saved in a file (or end state).
<solve>	<solve> . . . </solve>	This keyword is used to specify a fault tree/sequence solve operation (i.e. solve for cut sets).
<sort order>	<sort order> name </sort order>	This keyword is used to specify the type of sorted order of the output. This is used for sorting using a specified importance measure. This function sorts the basic events in alphabetical order.
<start>	<start> </start>	This keyword is used to specify the start of a new validation and verification test.
<sub type>	<sub type>base only </sub type>	This keyword is used to specify a report sub-type. For this example, the sub-type report is for base case results only.
<suscept>	<suscept> 4 </suscept>	This keyword is used to specify an event susceptibility value. 1=random, 2=fire, 3=flood, 4=seismic, etc.
<tau>	<tau>4</tau>	This keyword is used to specify a specific tau value in a change set.
<truncation>	<truncation> 1.0E-12 </truncation>	This keyword is used to specify a probability cut off level, a truncation. This function block is used within the <solve> block.
<type>	<type>cut set</type>	This keyword is used to specify the report type. For this example, the report type would be for a cut set output report.
<uncertainty>	<uncertainty> . . . </uncertainty>	This keyword is used to specify an uncertainty measure operation.
<unmark>	<unmark></unmark>	This keyword is used to clear any marked change sets in a plant model prior to additional analysis.
<update>	<update></update>	This keyword is used to specify an update operation for a cut set.

Keyword	Use/Example	Notes
<verbose>	<verbose> . . . </verbose>	This keyword is used to specify the verbose option on for the verification and validation test procedure of SAPHIRE. This keyword is used to setup the level of detail in the test results output files.
affected	<sub type> affected </sub type>	Parameter setting for a change set report that shows affected events adjusted in the change set (i.e. shows current case data)
all	<sub type> all </sub type>	Parameter setting for a change set report that shows both affected and unaffected basic events (i.e. shows both current and base case).
and	<mask operation>and </mask operation>	Parameter setting for setting for filtering masking/marking selections for fault trees and sequences.
ascii	<report format> ascii </report format>	Parameter setting for setting the report file out format to delimited ascii text.
average	<loop class> average </loop class>	Parameter setting for use with GEM initiating event operations. Set the loop class to be a combined average of Grid related, Plant centered, severe weather, and extremely severe weather classes.
base	<case>base</case>	Parameter setting to specify the base case is to be used.
by partition	<gather method> by partition </gather method>	Parameter setting for selection of an end state gather technique to be "gather cut sets by partition".
by sequence	<gather method>by sequence</gather method>	Parameter setting for selection of an end state gather technique to be "gather cut sets by sequence".
combination	<type> combination </type>	Parameter setting to specify a combination report type (summary report) for fault tree, sequences and end states reports.
current	<case>current</case>	Parameter setting to specify the current case is to be used.
current base	<sub type>current base</sub type>	Parameter setting for selection of a report sub type to have output for both the current case and the base case.
current only	<sub type>current only</sub type>	Parameter setting for selection of a report sub type to be current case only.

Keyword	Use/Example	Notes
cut set	<type>cut set</type>	Parameter setting for selection of a report type to be for "cut sets." i.e. the output file will contain cut sets.
debug		Parameter setting for selecting the mode of the log file output used for validation and verification testing.
detail		Parameter setting for selecting the mode of the log file output in a high detail mode.
esw	<loop class> esw </loop class>	Parameter setting for use with GEM initiating event operations. Sets the loop class to "extreme severe weather."
file		Parameter setting for TBD
grid	<loop class> grid </loop class>	Parameter setting for use with Initiating Event operations. Set the loop class to "grid related."
group	<type>group</type>	Parameter setting for selection of an importance measure calculation type "group."
interval	<type>interval </type>	Setting for selection of an importance measure calculation, "interval." The other option is "ratio".
lhs	<method>lhs </method>	Parameter setting for Latin Hypercube Simulation method in an uncertainty calculation.
mcs	<method>mcs </method>	Parameter setting for Monte Carlo Simulation method in an uncertainty calculation.
name	<name>%P-10 </name>	Parameter setting for assigning the name of the database (or project) that the macro-script is being used on to the test scenario. For this example, if the project name was "TEST", then the name of the scenario would be "TEST-10".
no	<initial prompt>no </initial prompt>	Parameter setting for turning off a feature. For this example, the conformation prompt would not be implemented.
none		Parameter setting for TBD
or	<mask operation>or </mask operation>	Parameter setting for setting for filtering masking/marking selections for fault trees and sequences.
PPerDay	<set units> PPerDay </set units>	Parameter setting for <set units> to be "per day."
PPerDemand	<set units>	Parameter setting for <set units> to be "per demand."

Keyword	Use/Example	Notes
	PPerDemand </set units>	
PPerHour	<set units> PPerHour </set units>	Parameter setting for <set units> to be "per hour."
PPerMinute	<set units> PPerMinute </set units>	Parameter setting for <set units> to be "per minute."
PPerMonth	<set units> PPerMonth </set units>	Parameter setting for <set units> to be "per month."
PPerWeek	<set units> PPerWeek </set units>	Parameter setting for <set units> to be "per week."
PPerYear	<set units> PPerYear </set units>	Parameter setting for <set units> to be "per year."
plant centered	<loop class>plant centered</loop class>	Parameter setting for use with an Initiating Event operation. Sets the loop class to "plant centered."
print	<report destination> print </report destination>	Parameter setting for selecting a report output option to be to a system default printer.
project	<type>project</type>	Parameter setting for selection of an uncertainty calculation type to "project".
random	<analysis>random</analysis>	Parameter setting for an analysis type to random.
ratio	<type>ratio</type>	Parameter setting for selection of an importance measure calculation to be "ratio". The other option is "interval"
results	<type>results</type>	Parameter setting for selection of a report type for analysis results.
rtf	<report format> rtf </report format>	Parameter setting for setting the file output format of a report to be RTF format.
severe weather	<loop class>severe weather</loop class>	Parameter setting for use with initiating event operations. Set loop class to severe weather."

Keyword	Use/Example	Notes
single	<type>single</type>	Parameter setting for selection of an importance measure calculation to single.
unaffected	<sub type> unaffected </sub type>	Parameter setting for a change set report that shows unaffected events not adjusted in the change set (i.e. shows base case data)
uncertainty	<type>uncertainty </type>	Parameter setting for selection of an importance measure calculation to be by "uncertainty."
view	<report destination> view </report destination>	Parameter setting for selecting a report output option to be to screen for viewing.
yes	<group>yes</group>	Parameter setting for turning on a feature.

A Macro-Script Example

The follow is a simple Macro-script example designed for the DEMO project that accompanies the SAPHIRE program. The Macro-script can be created using any standard text editor. The following Macro-script, when ran, creates a change set called "Example" and adds a description to it. The change set adjusts the probability of basic event C-PUMP-A to 1.0E-1. The macro-script then marks the change set and generates the new data. Next, all the sequences are marked and new cut sets are solved. Finally, the new cut set listing is displayed on the screen.

```
<comment>

The following is an example MACRO-SCRIPT that can be run in the
DEMO project of SAPHIRE.  This simple MACRO-SCRIPT performs the
following funtions automatically when implemented using the "Run
Macro..." command:

        Creates a Change Set named "EXAMPLE".
        Adds a description to the change set.
        Changes the probability of basic event C-PUMP-A to 1.0E-1.
        Generates the new data.
        Marks all Sequences.
        Solves for cuts sets.
        Generates a report of the new cut sets to the screen.

</comment>

<change set>
 <add>
        <name>EXAMPLE</name>
        <description>Changes C-PUMP-A probability to 1.0E-1</description>
  <single>
        <event name>C-PUMP-A</event name>
        <probability>1.0E-1</probability>
  </single>
 </add>
 <mark name>EXAMPLE</mark name>
 <generate></generate>
</change set>

<sequence>
 <unmark></unmark>
 <include>
  <mark event tree mask>*</mark event tree mask>
  <mask operation>or</mask operation>
  <mark sequence mask>*</mark sequence mask>
  <mask operation>or</mask operation>
  <mark logic fault tree>*</mark logic fault tree>
 </include>

 <solve></solve>
  <report>
        <type>cut set</type>
        <report destination>view</report destination>
  </report>

</sequence>
```

22.2 Creating a Macro-Script

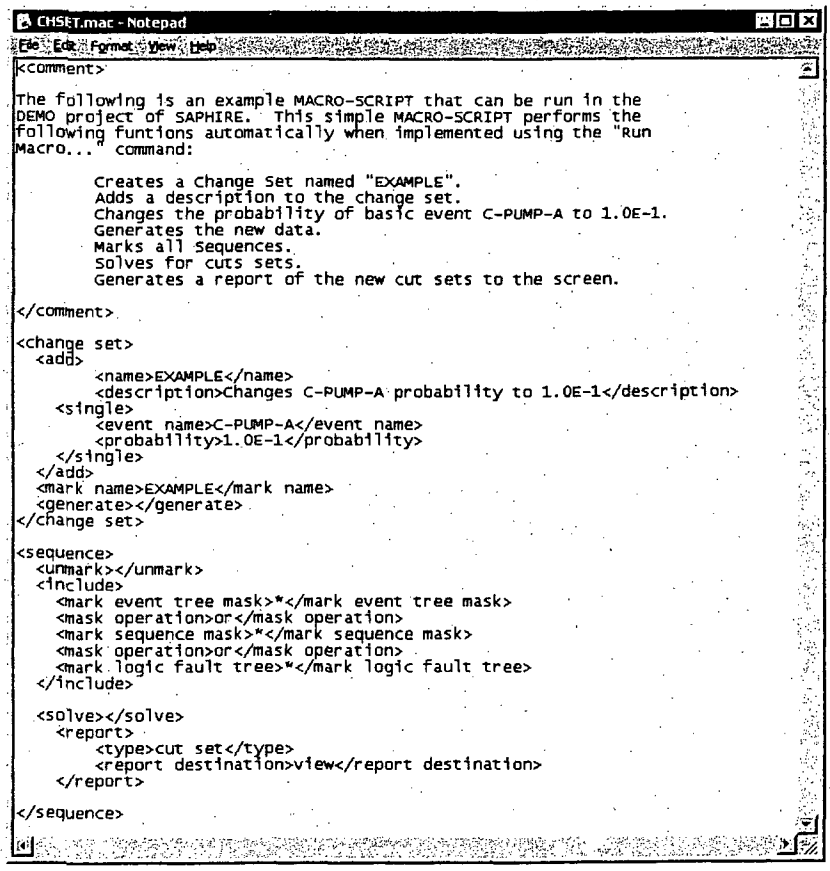
22.2.1 Making a Macro-Script

PURPOSE

Describe the steps for creating a Macro-Script.

STEPS

1. Select any text editor or program that saves files in a standard ASCII file format. For this example, Microsoft® NotePad will be used.
2. In NotePad, type in the desired Macro-Script using the Keywords and commands shown in the table above.



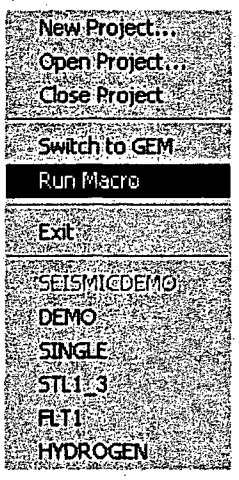
3. Save the file in NotePad by selecting **File → Save As...** and save the file in a specified directory or the Project directory. SAPHIRE reads the maco-script files with a "**.MAC**" file extention. For this example, the file was saved as CHSET.mac

The macro-script is ready to be accessed and implemented in SAPHIRE.

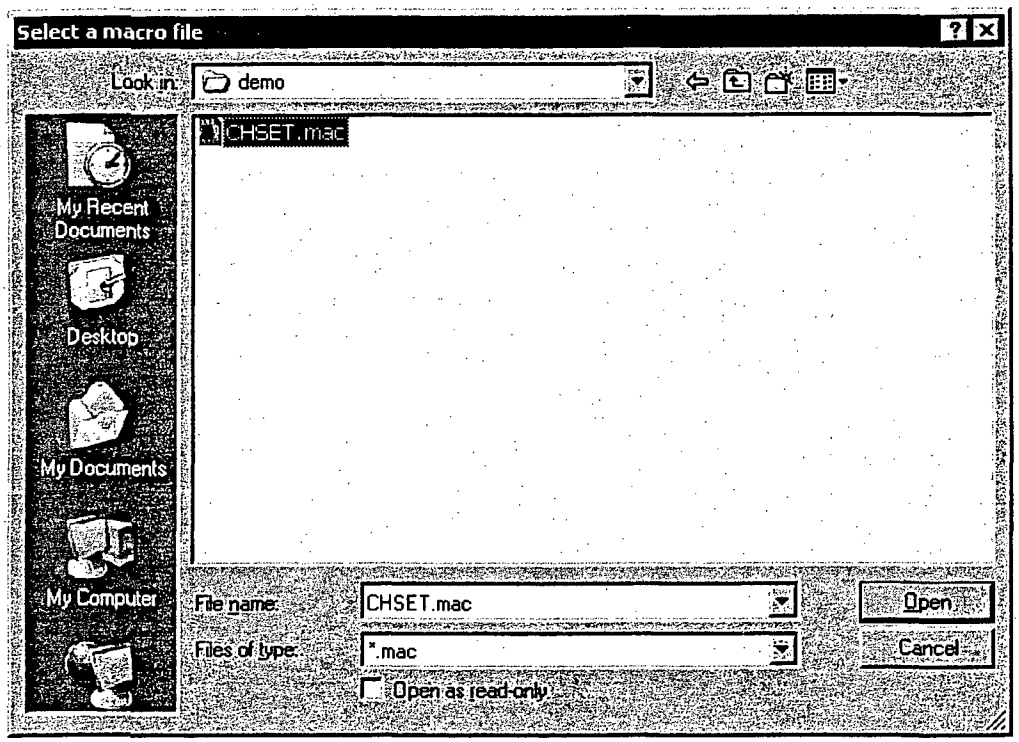
22.2.2 Running a Macro-Script.

To illustrate the use of a macro-script, the DEMO project will be used.

1. To run a macro-script, select **File** from the menu bar and use the drop-down menu to select **Run Maco**.



2. A "**Select a macro file**" pop-up window will be displayed. Use the "**Look in:**"drop box to select the location of the macro-script and select the desired file (macro-script files are read with a *.mac file extension). Click on the **Open** button to run the macro-script.



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-05-00644

2. TITLE AND SUBTITLE
Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Vol. 3 Code Reference Manual – Part B

3. DATE REPORT PUBLISHED
MONTH: September
YEAR: 2008

4. FIN OR GRANT NUMBER
N6203

5. AUTHOR(S)
K. J. Kvarfordt, C. L. Smith, S. T. Wood

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) is a software application developed for performing a complete probabilistic risk assessment (PRA) using a personal computer (PC) running the Microsoft Windows operating system. SAPHIRE is primarily funded by the U.S. Nuclear Regulatory Commission (NRC) and developed by the Idaho National Laboratory (INL). This reference guide will introduce the SAPHIRE Version 7.0 software. A brief discussion of the purpose and history of the software is included along with general information such as installation instructions, starting and stopping the program, and some pointers on how to get around inside the program. Next, database concepts and structure are discussed. Following that discussion are nine sections, one for each of the menu options on the SAPHIRE main menu, wherein the purpose and general capabilities for each option are furnished. Next, the capabilities and limitations of the software are provided. Finally, a series of appendices are provided that detail information about some of the more complex operations of the software.

12. KEY WORDS/DESCRIPTORS (List words or phrases that will assist researchers in locating the report.)
SAPHIRE, software, reliability, risk, safety, PRA

13. AVAILABILITY STATEMENT
Unlimited

14. SECURITY CLASSIFICATION
(This page)
Unclassified
(This report)
Unclassified

15. NUMBER OF PAGES

16. PRICE

NRC FORM 335 (2-89)

Federal Recycling Program

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, DC 20555-0001

OFFICIAL BUSINESS