Appendix B

General MAR-D Data Interchange Formats

B. General MAR-D Data Interchange Formats

This appendix enumerates the formats for each of the various data interchange formats as of August 2005.

Except where noted, file formats are the same for SAPHIRE versions 6 and 7. The primary version differences occur in the name and description files. This is because version 7 has the capability to provide an alternate name and description for each data type. In addition, descriptions in version 6 are 60 characters long, whereas in version 7, they are 120 characters long.

The file formats are backward compatible: version 6 files can be successfully loaded into version 7. It is not recommended that version 7 files be loaded into version 6, due to the presence of subtle format and content changes.

General Format Rules

1. 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.

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

3. 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.

4. Text rules:

    1. File is standard ASCII text, single spaced, upper- and lower-case.
    2. ^EOS signals the End of Section so that multiple names in the same project can be collected in one file.

These rules apply to all files unless specifically stated otherwise.

Project (Plant) Information

Project Names and Descriptions (Version 6)

File Name:
xxxxxx.FAD
File Format:
name,description
where

Name	24 character	Project name
description	60 character	Project description

Project Names and Descriptions (Version 7)

File Name:
xxxxxx.FAD
File Format:
name,description[,A]
where

name	24 character	Project name
description	120 character	Project description
A	1 character	If included indicates alternate description

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

Project Recovery Rules

File Name:
xxxxxxxx.FAY
File Format:
project =
-- recovery rule text --
where

project	24 character	Project name

System (Fault Tree) Recovery Rules

File Name:
xxxxxxxx.FAS
File Format:
project =
-- recovery rule text --
where

project	24 character	Project name

Project Partition Rules

File Name:
xxxxxxxx.FAP
File Format:
project =
-- partition rule text --
where

project	24 character	Project name

Project Textual Information (Version 6)

File Name:
xxxxxx.FAT
File Format:
Project =
-- text --
where

project	24 character	Project name

Project Textual Information (Version 7)

File Name:
xxxxxx.FAT
File Format:
Project [,A] =
-- text --
where

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

Basic Event Information

Event Names and Descriptions (Version 6)

File Name:
xxxxxx.BED
File Format:
project =
name,description
. . . , . . .
where

project	24 character	Project name
name	24 character	Event primary name
description	120 character	Alphanumeric description

Event Names and Descriptions (Version 7)

File Name:
xxxxxx.BED
File Format:
project =
name,description[,A]
. . . , . . .
where

project	24 character	Project name
name	24 character	Event primary name
description	120 character	Alphanumeric description
A	1 character	If included indicates alternate description

Basic Event Rate Information (Version 6)

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	same as type 3
3	1 Exp(Lambda * Mission Time)
4	same as type 5
5	Operating component with full repair
6	same as type 7
7	1+(EXP(Lambda*Tau) 1.0)/(Lambda*Tau)
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 min cut upper bound
E	Use end state min cut upper bound
G	Seismic event - Enter g level for screening
H	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)
D	Dirichlet, b of Dirichlet(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
S	Seismic Log Normal
O	Constrained non-informative

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:

1. The name field is mandatory.

Basic Event Rate Information (Version 7)

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
V	Value event (input to compound event)
3	1 Exp(Lambda * Mission Time)
5	Operating component with full repair
7	1+(EXP(Lambda*Tau) 1.0)/(Lambda*Tau)
T	Set to House Event (Failed, Prob=1.0)
F	Set to House Event (Successful,Prob=0.0)
I	Set to ignore
C	Compound event
S	Use fault tree min cut upper bound
E	Use end state tree min cut upper bound
G	Seismic event - Enter g level for screening
H	Seismic event - Use medium site hazard curve for screening

UdC	24 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)

D	Dirichlet, b of Dirichlet(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
S	Seismic Log Normal
O	Constrained non-informative
T	Triangular, mode, upper end of Triangular(m, u)

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:

1. The name field is mandatory.

Basic Event Attribute Codes

Basic event attributes are entered through MODIFY--Basic Event and stored in Event.

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	16 values of Y or N (yes or no) indicate whether the

	flags	attribute described in the class attribute file is applicable.

General Rules:

1. The name field is mandatory.

Basic Event Transformations (Version 6)

In SAPHIRE version 6.0, both transformation and compound information are extracted into and loaded from this file type.

File Name:
xxxxxx.BET
File Format:
project =
name1,level,type, library, procedure
bename1, bename2, . . . ,
. . . , benameN
^EOS
name2,level,type, library, procedure
bename1, bename2, . . . ,
. . . , benameN
^EOS
where

project	24 character	Project name
name	24 character	Event name
level	3 character	Transformation level (0..99)
type	4 character	Transformation type (AND, OR, ZOR, COM, blank)
library	60 character	name of plug in library (for COM events)
procedure	60 character	name of procedure in plug in library (for COM events)
bename1..N	24 character	Event name

Basic Event Transformations (Version 7)

File 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
level	3 character	Transformation level (0..99)
type	4 character	Transformation type (AND, OR, ZOR, blank)
bename1..N	24 character	Event name

Basic Event Compound Information (Version 7 only)

In SAPHIRE version 7.0, compound information is extracted into its own file type. Compound events can still be loaded from .BET files (where version 6.0 extracts compound information).

File Name:
xxxxxx.BEC
File Format:
project =
name1,level,type
bename1, bename2, . . . ,
. . . , benameN
^EOS
name2,level,type, library, procedure
bename1, bename2, . . . ,
. . . , benameN
^EOS
where

project	24 character	Project name
name	24 character	Event name
level	3 character	0 or blank
type	4 character	COM
library	60 character	name of plug in library
procedure	60 character	name of procedure from plug in library
bename1..N	24 character	Event name

Event Attribute Descriptions

Failure Mode Descriptions (Version 6)

File Name:
xxxxxx.FMD
File Format:
project =
fail, description
. . . , . . .
where

project	24 character	Project name
Fail	3 character	Failure mode primary identifier
description	60 character	Failure mode description

Failure Mode Descriptions (Version 7)

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	120 character	Failure mode description
A	1 character	If included indicates alternate description

Component Type Descriptions (Version 6)

File Name:
xxxxxx.CTD
File Format:
project =
comp, description
. . . , . . .
where

project	24 character	Project name
comp	3 character	Component type primary identifier

description	60 character	Component type description

Component Type Descriptions (Version 7)

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	120 character	Component type description
A	1 character	If included indicates alternate description

System Type Descriptions (Version 6)

File Name:
xxxxxx.STD
File Format:
project =
sys,description
. . . , . . .
where

project	24 character	Project name
sys	5 character	System primary identifier
description	60 character	System description

System Type Descriptions (Version 7)

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	120 character	System description
A	1 character	If included indicates alternate description

Location Descriptions (Version 6)

File Name:
xxxxxx.LCD
File Format:
project =
loc,description
. . . , . . .
where

project	24 character	Project name
loc	3 character	Location primary identifier
description	60 character	Location description

Location Descriptions (Version 7)

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	120 character	Location description
A	1 character	If included indicates alternate description

Class Attribute Descriptions (Version 6)

File Name:
xxxxxx.TTD
File Format:
project =
attr,description
. . . , . . .
where

project	24 character	Project name
Attr	3 character	Class attribute primary name

description	60 character	Class attribute description

Class Attribute Descriptions (Version 7)

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	120 character	Class attribute description
A	1 character	If included indicates alternate description

Fault Tree Information

Fault Tree Names and Descriptions (Version 6)

File Name:

xxxxxx.FTD

File Format:

project =
name,description[,s]
. . . , . . .

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

Fault Tree Names and Descriptions (Version 7)

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	120 character	Fault tree description
S	1 character	If included indicates fault tree is a sub-tree
A	1 character	If included indicates alternate description

Fault Tree Graphics

Fault tree graphics are stored in the block data file of the Graphics 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 6.0/7.0 Fault Tree Graphics file (DLS format)

Fault Tree Logic (Version 6)

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:

1. A gate definition cannot exceed 255 characters. (Use the CONT gate to break up definitions.)

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

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

Fault Tree Logic (Version 7)

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		120 character	gate name descriptions included as comment

General Rules:

1. A gate definition cannot exceed 255 characters. (Use the CONT gate to break up definitions.)

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

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

Fault Tree Cut Sets

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:

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

2. A plus sign (+) separates cut sets.

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

4. A slash (/) precedes complemented events.

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

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

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	

Fault Tree Recovery Rules

File Name:
xxxxxxxx.FTY
File Format:
project =
-- recovery rule text --
where

project	24 character	Project name

Fault Tree Textual Information (Version 6)

File Name:
xxxxxx.FTT

File Format:

project, fault tree =
-- text --
^EOS
project, fault tree2 =
. . .

where

project	- 24 character	Project name
fault tree	- 24 character	Fault tree name

Fault Tree Textual Information (Version 7)

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

Fault Tree Graphical P&ID

The piping and instrumentation diagrams is a graphics file in binary format. It will be loaded and output as-is: no conversion will be performed.

File Name:
xxxxxxxx.PID
File Format:

IRRAS 4.0/5.0, SAPHIRE 6.0 and 7 .0 P&ID Graphics file (PID Format)

Event Tree Information

Event Tree Names and Descriptions (Version 6)

File Name:
xxxxxx.ETD

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

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 transfer tree

Event Tree Names and Descriptions (Version 7)

File Name:
xxxxxx.ETD
File Format:
project =
name,description[,s][,A]
. . . , . . .
where

Project	24 character	Project name
Name	24 character	Event tree name
Description	120 character	Event tree description
S	1 character	If included indicates event tree is a transfer tree
A	1 character	If included indicates alternate description

Event Tree Attributes

File Name:
xxxxxx.ETA
File Format:
project =
name,init
. . . , . . .
where

project	24 character	Project name

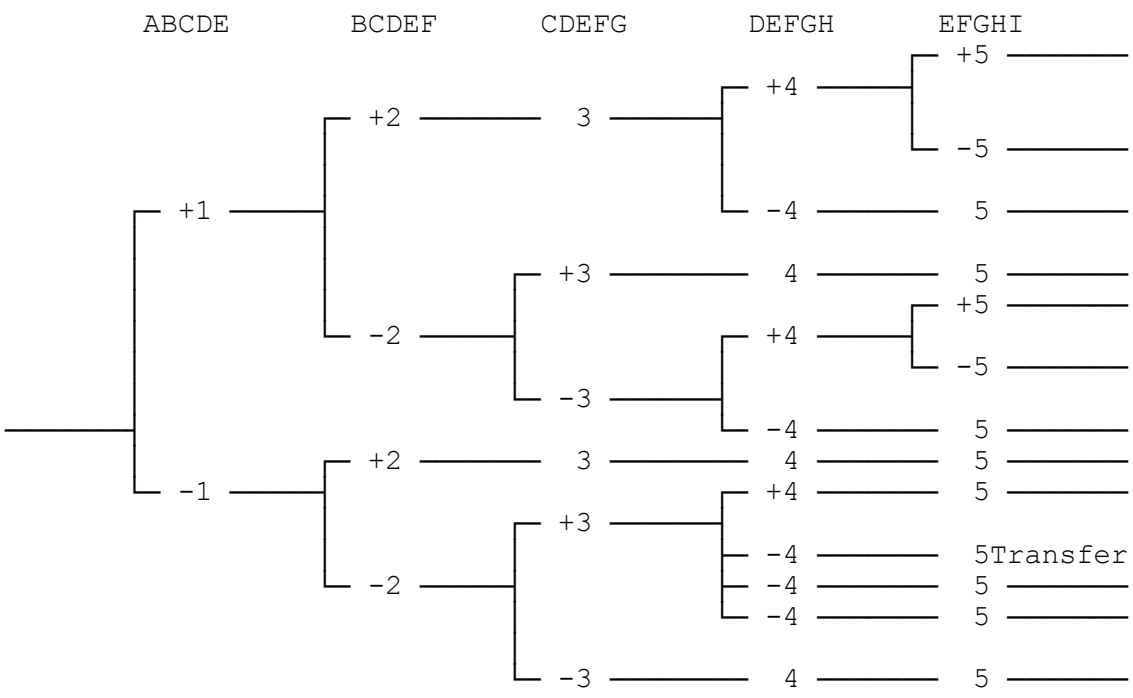
name	24 character	Event tree name
init event	24 character	Initiating Event

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

SAMPLE GRAPHICAL EVENT TREE



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
"120 character line of text"
XY xvalue, yvalue
```

"120 character line of text"
"120 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:

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

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

3. 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.

4. 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.

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

6. The ^PARMS command allows input of program control parameters.

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

Event Tree Textual Information (Version 6)

File Name:
xxxxxx.ETT

File Format:
project, event tree =
-- text --
^EOS
project, event tree2 =
-- text --

where

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

Event Tree Textual Information (Version 7)

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

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

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

End State Information

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 SARA *.PDS file.

End State Names and Descriptions (Version 6)

File Name:
xxxxxx.ESD

File Format:
project =
name,description
. . . , . . .

where

project	24 character	Project primary name
name	24 character	End state primary name
description	60 character	End state description

End State Names and Descriptions (Version 7)

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	120 character	End state description
A	1 character	If included indicates alternate description

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

End State Textual Information (Version 6)

File Name:
xxxxxx.EST

File Format:
project, end state =
-- text --
^EOS
project, end state2 =

where

project	24 character	Project name
end state	24 character	End state name

End State Textual Information (Version 7)

File Name:
end-state.EST
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

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:

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

2. A plus sign (+) separates cut sets.

3. A period (.) denotes the end of the end state cut sets.

4. A slash (/) precedes complemented events.

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

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

Sequence Information

Sequence Names and Descriptions (Version 6)

File Name:

xxxxxx.SQD

File Format:

project,eventree =
name,description
. . . , . . .
^EOS

where

project	24 character	Project name
event tree	24 character	Event tree name
name	24 character	Sequence name
description	60 character	Sequence description

Sequence Names and Descriptions (Version 7)

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	120 character	Sequence description
A	1 character	If included indicates alternate description

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 +hjn
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:

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

2. A plus sign (+) separates cut sets.

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

4. A slash (/) precedes complemented events.

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

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

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	5^{th} 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

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

Sequence Textual Information (Version 6)

File Name:
xxxxxx.SQT

File Format:
project, event tree, sequence=
--- 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

Sequence Textual Information (Version 7)

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

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

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

Gates

Gate Description (Version 6)

File Name:

xxxxxx.GTD

File Format:

project=
name,description

where

Project	24 character	Project name
Name	24 character	Gate name
description	120 character	Gate description

Gate Description (Version 7)

File Name:

xxxxxx.GTD

File Format:

project=
name,description[,A]

where

Project	24 character	Project name
Name	24 character	Gate name
description	120 character	Gate description
A	1 character	If included indicates alternate description

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

Change Sets

Change Set Description (Version 6)

File Name:

xxxxxx.CSD

File Format:

project=
name,description
...,...

where

project	24 character	Project name
name	24 character	Change set name
description	60 character	Change set description

Change Set Description (Version 7)

File Name:

xxxxxx.CSD

File Format:

project=
name,description[,A]
...,...

where

project	24 character	Project name
name	24 character	Change set name
description	120 character	Change set description
A	1 character	If included indicates alternate description

Change Set Information (Version 6)

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,..att24
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	same as type 3
3	1 Exp(-Lambda * Mission Time)
4	same as type 5
5	Operating component with full repair
6	same as type 7
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 min cut upper bound
E	Use end state min cut upper bound

	G	Seismic event - Enter g level for screening
	H	Seismic event - Use medium site hazard curve for screening

udT	1 character	Uncertainty distribution type

P	use point estimate
L	Log normal, error factor
N	Normal, standard deviation
B	Beta, b of Beta(a,b)
D	Dirichlet, b of Dirichlet
G	Gamma, a Gamma(a)
C	Chi-squared, degrees of freedom
E	Exponential, none
U	Uniform, Upper end pt.
H	Histogram
M	Maximum entropy
S	Seismic log normal, betaR, betaU
O	Constrained non-informative

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

Change Set Information (Version 7)

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
3	1 Exp(-Lambda * Mission Time)
5	Operating component with full repair
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 min cut upper bound
E	Use end state min cut upper bound
G	Seismic event - Enter g level for screening
H	Use medium site hazard curve
B	Use base case (even if prior marked change sets have altered the value)

udT	1 character	Uncertainty distribution type

P	Use point estimate
L	Log normal, error factor
N	Normal, standard deviation
B	Beta, b of Beta(a,b)
D	Dirichlet, b of Dirichlet(a,b)
G	Gamma, a of Gamma(a)
C	Chi-squared, degrees of freedom
E	Exponential, none
U	Uniform, Upper end pt.
H	Histogram
M	Maximum entropy
S	Seismic log normal, betaR, betaU
O	Constrained non-informative

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	24 characters	Uncertainty correlation class. Events in same class are 100% correlated.
mission	Floating point	Mission time
init	Boolean (T/F)	Initiating event

Change Set Attributes (Version 7 only)

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

Histograms

Histogram Description (Version 6)

File Name:

xxxxxxxx.HID

File Format:

project =
name,type,subtype,description

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	

Histogram Description (Version 7)

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	120 character	Histogram description	
A	1 character	If included indicates alternate description	

Histogram Information

File Name:
xxxxxxxx.Hll
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	

Histogram Attributes (Version 7 only)

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	

Slices

Slice Descriptions (Version 6)

File Name:
xxxxxxxx.SLD
File Format:
project =
name, description
where

project	24 character	Project name
name	24 character	Slice name
description	60 character	Slice description

Slice Descriptions (Version 7)

File Name:
xxxxxxxx.SLD
File Format:
project =
name, description[, A]
where

project	24 character	Project name

name	24 character	Slice name
description	120 character	Slice description
A	1 character	If included indicates alternate description

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:

1. 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.

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

3. A slash (/) precedes complemented events.

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

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

Slice Basic 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

Slice Basic Attributes (Version 7 only)

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

SETS FORMAT

Sequences (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 SAPHIRE at this time.

File Format:
sequence-name =
eventName * eventName +
eventName * eventName.
where
. . . . .

General Rules:

1. An asterisk (*) separates event names. Spaces are ignored.
2. A plus sign (+) separates cut sets.
3. A period (.) denotes the end of a sequence.
4. An asterisk (*) in the first column denotes a comment.

Fault Trees (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.

Fault Tree Cut sets

The fault tree cutsets are stored in the System relation in the block data file. The format of the cutset file (.DNF) is given above.

Basic Events (SETS)

Basic Event Descriptions

File Name:
xxxxxxx.DES.
File Format:
name $ description $
name $ description $
where

name	event name
name list	description of event

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

Appendix C

MAR-D Files for Sample Database

C. MAR-D Files for Sample Database

SAPHIRE Version 6 MAR-D formats for the Sample Database are presented. Version 6 results were selected for presentation since they can be loaded into both versions 6 and version 7.

Note that these examples are shown in a document created by a word processor. Actual MAR-D files should be edited in a text editor, such as Notepad, so that formatting codes are not embedded into the text. SAPHIRE handles only ASCII text characters.

In this document, some line wrapping occurs so that entire lines can be displayed. Where this occurs in this document, the wrapped line will appear indented.

PROJECT FILES

These are examples of files (or partial files) in MAR-D formats for the Sample database. These formats are as of August 2005.

Project Names and Description File (.FAD)

```
SAMPLE          ,This is a sample data base
```

Project Attribute File (.FAA)

```
SAMPLE                  , 0001 =
*  Name          , Mission   , NewSum    , Company  , Location       ,Typ,
      Design   ,Vendr, Arch Eng , OpDate   , QualDate
SAMPLE                  , 2.400E+001,+0.000E+000,STANDARD  ,HOMETOWN        ,
      ,          ,     ,          ,----/--/--,----/--/--
```

Project Text File (.FAT)

```
SAMPLE           =
      A simple example that models the probability of getting to work on time.
SAMPLE           =
 A simple example that models the probability of getting to work on time.
```

BASIC EVENT FILES

Basic Event Names and Description File (.BED)

```
SAMPLE                =
ALARM                ,ALARM CLOCK FAILURE
ALM-BPF              ,Alarm fails due to battery failure
ALM-CPF              ,Alarm fails due to commercial power failure
ALM-FTS              ,Alarm fails because worker fails to set
ALM-MECH             ,Alarm fails due to mechanical failure
ALM-SWT              ,Alarm fails because worker set wrong time
MEDICINE             ,Recovery for sick failure preventing attending work
OTHER                ,Other personal reasons that cause a failure to get to work
PER-TRNS             ,Personal transportation
PERSONAL             ,PERSONAL PROBLEMS
PUB-TRNS             ,Public transportation fails
PUB-TRNS-LATE        ,Public transportation fails late time frame
SICK                 ,Failed to get to work because of illness
SICK-FAM             ,Failed to get to work because of illness in project
TRNS-2               ,COMMERCIAL TRANSPORTATION FAILS AT A LATER TIME
TRNSPRT              ,PERSONAL AND COMMERCIAL TRANSPORTATION FAIL
WORK                 ,Event tree (WORK) initiating event
```

Basic Event Rate Information File (.BEI)

```
SAMPLE                    =
*  Name          ,FdT,UdC,UdT, UdValue , Prob       , Lambda     , Tau        ,
      Mission ,Cat,PF, UdValue2
ALARM            ,1,     ,L, 1.000E+000, 1.000E+000,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
ALM-BPF          ,1,     ,L, 3.000E+000, 9.000E-008,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
ALM-CPF          ,1,     ,L, 3.000E+000, 1.500E-002,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
ALM-FTS          ,1,     ,L, 1.000E+001, 5.500E-006,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
ALM-MECH         ,1,     ,L, 3.000E+000, 2.700E-008,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
ALM-SWT          ,1,     ,L, 1.000E+001, 2.700E-003,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
MEDICINE         ,1,     ,L, 5.000E+000, 5.000E-001,+0.000E+000,
      +0.000E+000,+0.000E+000,R, ,+0.000E+000
OTHER            ,1,     ,L, 1.000E+001, 8.100E-003,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
PER-TRNS         ,1,     ,L, 5.000E+000, 5.500E-003,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
PERSONAL         ,1,     ,L, 1.000E+000, 1.000E+000,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
PUB-TRNS         ,1,     ,L, 3.000E+000, 2.700E-003,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
PUB-TRNS-LATE ,1,     ,L, 3.000E+000, 2.000E-003,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
SICK             ,1,     ,L, 1.000E+001, 8.100E-003,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
SICK-FAM         ,1,     ,L, 1.000E+001, 4.000E-003,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
TRNS-2           ,1,     ,L, 1.000E+000, 1.000E+000,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
```

```
TRNSPRT       ,1,    ,L, 1.000E+000, 1.000E+000,+0.000E+000,
      +0.000E+000,+0.000E+000, , ,+0.000E+000
WORK          ,1,    ,L, 2.000E+000, 2.480E+002,+0.000E+000,
      +0.000E+000,+0.000E+000,I, ,+0.000E+000
```

Basic Event Attribute File (.BEA)

```
SAMPLE         =
*  Name        ,AltName                ,Typ,Sys,Fail,Loc,CompId,Train,
      Attributes                     ,Template Name           ,Use
      TemplateFlags                                                ,Shape
ALARM          ,ALARM                    ,DE ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
ALM-BPF        ,ALM-BPF                  ,   ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
ALM-CPF        ,ALM-CPF                  ,   ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
ALM-FTS        ,ALM-FTS                  ,   ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
ALM-MECH       ,ALM-MECH                 ,   ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
ALM-SWT        ,ALM-SWT                  ,   ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
MEDICINE       ,MEDICINE                 ,   ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
OTHER          ,OTHER                    ,   ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
PER-TRNS       ,PER-TRNS                 ,   ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,B
PERSONAL       ,PERSONAL                 ,DE ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
PUB-TRNS       ,PUB-TRNS                 ,   ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
PUB-TRNS-LATE ,PUB-TRNS-LATE            ,   ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N, ,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,B
SICK           ,SICK                     ,   ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
SICK-FAM       ,SICK-FAM                 ,   ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
TRNS-2         ,TRNS-2                   ,DE ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
TRNSPRT        ,TRNSPRT                  ,DE ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
```

```
WORK          ,WORK                    ,   ,   ,   ,   ,       ,
      ,Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,
      ,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,
```

FAULT TREE FILES

Fault Tree Names and Description File (.FTD)

```
SAMPLE                   =
ALARM       ,ALARM CLOCK FAILURE
PERSONAL    ,PERSONAL PROBLEMS
TRNS-2      ,COMMERCIAL TRANSPORTATION FAILS AT A LATER TIME
TRNSPRT     ,PERSONAL AND COMMERCIAL TRANSPORTATION FAIL
```

Fault Tree Logic File (.FTL)

```
SAMPLE, ALARM =
ALARM                    OR   ALARM-1 ALARM-2 ALM-MECH
ALARM-1                  OR   ALM-FTS ALM-SWT
ALARM-2                  AND  ALM-BPF ALM-CPF
^EOS
SAMPLE, PERSONAL =
PERSONAL                 OR   OTHER SICK SICK-FAM
^EOS
SAMPLE, TRNS-2 =
TRNS-2                   AND  PER-TRNS PUB-TRNS-LATE
^EOS
SAMPLE, TRNSPRT =
TRNSPRT                  AND  PER-TRNS PUB-TRNS
```

Fault Tree Graphics File (.DLS)

```
NOT IN ASCII FORMAT
```

Fault Tree Cut Sets File (.FTC)

```
SAMPLE, ALARM, 0001=
ALM-BPF * ALM-CPF +
ALM-FTS +
ALM-MECH +
ALM-SWT .
^EOS
SAMPLE, PERSONAL, 0001=
OTHER +
SICK +
SICK-FAM .
^EOS
SAMPLE, TRNS-2, 0001=
PER-TRNS * PUB-TRNS-LATE .
^EOS
SAMPLE, TRNSPRT, 0001=
PER-TRNS * PUB-TRNS .
```

Fault Tree Attribute File (.FTA)

```
SAMPLE, 0001 =
* Name             , Level, Mission    , MinCut     , Def ProCut,Used
      ProCut,Sample,Seed,Siz,Sys,Cuts,Events, UdValues,
      Def Flags,               Used Flags,      S QMethod, S QPasses, R
      QMethod, R QPasses
ALARM      ,0, 2.400E+001, 2.706E-003,------E----,------E----, -----,-----,--,
      ,------,-----,------E----,------E----,------E----,------E----,------E---
      -,------E----,------E----,------E----,------E----,
      ,                         , ,----,M,   0
PERSONAL   ,0, 2.400E+001, 2.007E-002,------E----,------E----, -----,-----,--,
      ,------,-----,------E----,------E----,------E----,------E----,------E---
      -,------E----,------E----,------E----,------E----,
      ,                         , ,----,M,   0
TRNS-2     ,0, 2.400E+001, 1.100E-005,------E----,------E----, -----,-----,--,
      ,------,-----,------E----,------E----,------E----,------E----,------E---
      -,------E----,------E----,------E----,------E----,
      ,                         , ,----,M,   0
TRNSPRT    ,0, 2.400E+001, 1.485E-005,------E----,------E----, -----,-----,--,
      ,------,-----,------E----,------E----,------E----,------E----,------E---
      -,------E----,------E----,------E----,------E----,
      ,                         , ,----,M,   0
```

Fault Tree Text File (.FTT)

```
SAMPLE, ALARM=
The ALARM fault tree is a simple representation modeling alarm clock failure.
Some common reasons for alarm clock failure include setting the wrong time,
mechanical failure, or power failure (either battery or commercial).
```

EVENT TREE FILES

Event Tree Names and Descriptions File (.ETD)

```
SAMPLE           =
WORK            ,WORK EVENT TREE
```

Event Tree Graphics File (.ETG)

```
SAMPLE, WORK, WORK =
^WINVER1.0
^TOPS
ALARM, PERSONAL, TRNSPRT
^LOGIC
 +1 +2 +3
       -3
    -2  3
 -1  2 +3
       -3
^SEQUENCES
Y, SEQUENCE, N, Y, N, STATE, N, N,
N, N, N, OK, Y, A, Y, OK,
Y, B, Y, LATE-TO-WORK, Y, , Y, ,
Y, C, Y, MISS-WORK, Y, , Y, ,
Y, D, Y, LATE-TO-WORK, Y, , Y, ,
Y, E, Y, LATE-TO-WORK, Y, , Y, ,
^TOPDESC
 "INITIATING EVENT"
!
 "ALARM FAILURE"
!
 "PERSONAL FAILURE"
!
 "TRANSPORTATION",
 "FAILURE"
!
^PARMS
START 52.00, 809.60 WINDOW 132.00, 363.50, 1043.00, 1274.50
ASPECTRATIO 0.74
HEADER 972.00, 1224.00, 1476.00, 1728.00
STRING E
DEFFONT 5
TOPWIDTH 16
TOPSIZE -15.00
TOPFONT 1
TOPFACE Times_New_Roman
TOPPITCH 2
TOPCOLOR 15
DESHITE 3
DESSIZE -10.00
DESFONT 5
DESFACE Times_New_Roman
DESCOLOR 15
DESPITCH 2
NODEHITE 20.00
ENDSIZE -15.00
ENDFONT 1
ENDFACE Times_New_Roman
ENDPITCH 2
ENDCOLOR 15
```

```
BACKCOLOR 1
TOPBACKCOLOR 1
LINECOLOR 15
HILITECOLOR 1
LOCALE 1033
MODDATE 2003/09/23
```

Event Tree Logic File (.ETL)

```
SAME AS THE .ETG FILE SECTION C.5.2
```

Event Tree Attribute File (.ETA)

```
SAMPLE           =
*  Name         , Init Event
WORK            , WORK
```

Event Tree Rules File (.ETR)

```
SAMPLE, WORK=
| rule to substitute TRNS-2 for TRNSPRT
if ALARM then
  TRNSPRT = TRNS-2;
endif
```

Event Tree Recovery Rules (.ETY)

```
SAMPLE, WORK=
| rule to add recovery potential to the cut sets
if SICK then
  recovery = MEDICINE;
endif
```

Event Tree Text File (.ETT)

```
SAMPLE, WORK=
A FAIL-SUCCESS LOGIC WAS USED TO DEVELOP AN EVENT TREE TO CALCULATE THE
FREQUENCY THAT THE AVERAGE PERSON WILL ARRIVE ON TIME, BE LATE, OR MISS A DAY
OF WORK.
```

END STATE FILES

End State Names and Description File (.ESD)

```
SAMPLE           =
LATE-TO-WORK      , This end state represents being late to work
MISS-WORK         , This end state represents missing work
```

End State Text File (.EST)

```
SAMPLE, LATE-TO-WORK=
THIS IS THE LATE TO WORK END STATE.
```

SEQUENCE FILES

Sequence Names and Description File (.SQD)

```
SAMPLE, WORK=
2               ,LATE TO WORK
3               ,MISS WORK
4               ,LATE TO WORK
5               ,LATE TO WORK
```

Sequence Cut Set File (.SQC)

```
SAMPLE, WORK, 2, 0001=
PER-TRNS * PUB-TRNS .
^EOS
SAMPLE, WORK, 3, 0001=
OTHER +
SICK * MEDICINE +
SICK-FAM .
^EOS
SAMPLE, WORK, 4, 0001=
ALM-BPF * ALM-CPF +
ALM-FTS +
ALM-MECH +
ALM-SWT .
^EOS
SAMPLE, WORK, 5, 0001=
ALM-BPF * ALM-CPF * PER-TRNS * PUB-TRNS-LATE +
ALM-FTS * PER-TRNS * PUB-TRNS-LATE +
ALM-MECH * PER-TRNS * PUB-TRNS-LATE +
ALM-SWT * PER-TRNS * PUB-TRNS-LATE .
```

Sequence Cut Set Attribute File (.SQA)

```
SAMPLE, WORK, 0001=
* Name         , End State              , MinCut    , Mission   , ProCut
      ,Sample,Seed,Siz,Cuts,Events, UdValues, Def Flags, Used FlagsS QMethod,
      S QPasses, R QMethod, R QPasses
2    ,LATE-TO-WORK    , 3.683E-003, 2.400E+001,------E----, 1000,40777,--,
      1,    3,------E----,------E----,------E----,------E----, ------E----,---
      ---E----,------E----,------E----,------E----,     ,     , ,----,M,   0
3    ,MISS-WORK       , 3.985E+000, 2.400E+001,------E----, 1000,46267,--,
      3,    5,------E----,------E----,------E----,------E----, ------E----,---
      ---E----,------E----,------E----,------E----,     ,     , ,----,M,   0
4    ,LATE-TO-WORK    , 6.710E-001, 2.400E+001,------E----, 1000,52257,--,
      4,    6,------E----,------E----,------E----,------E----, ------E----,---
      ---E----,------E----,------E----,------E----,     ,     , ,----,M,   0
5    ,LATE-TO-WORK    , 7.381E-006, 2.400E+001,------E----, 1000,58407,--,
      4,    8,------E----,------E----,------E----,------E----, ------E----,---
      ---E----,------E----,------E----,------E----,     ,     , ,----,M,   0
```

Sequence Logic File (.SQL)

```
SAMPLE, WORK, 2=
/ALARM /PERSONAL TRNSPRT .
^EOS
SAMPLE, WORK, 3=
/ALARM PERSONAL .
^EOS
SAMPLE, WORK, 4=
ALARM /TRNSPRT .
^EOS
SAMPLE, WORK, 5=
ALARM TRNS-2 .
```

Sequence Text File (.SQT)

```
SAMPLE, WORK, 3=

Sequence 3 is the event tree sequence that is used to demonstrate the use of
recovery rules or recovery actions.
```

GATE FILES

Gate Description File (.GTD)

```
SAMPLE           =
ALARM          , ALARM CLOCK FAILURE
ALARM-1        , ALARM CLOCK SETTING FAILURE
ALARM-2        , ALARM CLOCK POWER FAILURE
PERSONAL       , PERSONAL PROBLEMS
TRNS-2         , COMMERCIAL TRANSPORTATION FAILS AT A LATER TIME
TRNSPRT        , PERSONAL AND COMMERCIAL TRANSPORTATION FAILURE
```

Gate Attributes File (.GTA)

```
SAMPLE                    =
*  Name           , Type
ALARM                     , OR
ALARM-1                   , OR
ALARM-2                   , AND
PERSONAL                  , OR
TRNS-2                    , AND
TRNSPRT                   , AND
```

This page left blank.

Appendix D

Seismic Data Loading

D. Seismic Data Loading

INTRODUCTION

This appendix discusses the features and basic data loading processes of the seismic module in SAPHIRE 5.0. The seismic data loading process assumes the availability of internal-events PRA or database (i.e. a SAPHIRE data base implementing analysis with random failures within a particular system). The procedures necessary for seismic data loading using the SAPHIRE code are described in the following subsections.

SAPHIRE SEISMIC CAPABILITIES

The SAPHIRE seismic analysis capabilities are designed to function directly from the internal-events PRA. Thus, internal basic events, system fault tree models, accident sequences, and initiating events have all been defined and developed for the system of interest. The SAPHIRE seismic analysis consists of taking the internal basic events (having random failures) and converting them into seismic basic events that represent seismic-induced failures. SAPHIRE performs transformations in the form of Boolean identities that allows the user to build on an internal-events analysis when developing a seismic model. After seismic vulnerabilities have been identified, they are incorporated into an existing internal-events analysis using a set of basic event transformations that substitute in seismic-induced failures that are used to generate seismic sequence or system cut sets.

BUILDING AND LOADING THE SEISMIC SAPHIRE MODEL

Hazard Curves

The hazard curve represents a range of possible earthquake magnitudes. The curve is usually found in the form of a probability of exceedence curve, with the earthquake ground acceleration on the horizontal axis and the probability of exceeding that acceleration on the vertical axis. (Sources of hazard curve data and information include NUREG-1488 and NUREG-4550.) SAPHIRE uses this information in the form of a histogram or a discreet probability density distribution. For a more detailed description of hazard curves and the methodology on their use during seismic analysis, see the SAPHIRE Technical Reference Manual.

The hazard curve (or histogram) that will be used in the seismic analysis is developed or modified by selecting the desired seismic hazard curve in the SAPHIRE program. This is done by selecting Modify → Project main menu option. Under the heading "Site Hazard Curves", there are three fields: "Low", "Medium", and "High". The histogram listed in the "Medium" field will be the one used during analysis. If a seismic hazard curve is not available, then one must be added in order to generate quantified cut sets. A seismic hazard curve (or histogram) can be added (or loaded) into the SAPHIRE database using two methods. The histogram can be added and the discrete data points input from the Modify→ Histograms main menu option or it can be loaded from a histogram flat file (.HII) through the Utility → Load and Extract main menu option. The procedures for both methods are discussed below.

Loading the Seismic Histogram through the Modify main menu option

To add a seismic histogram, the following steps are required:

1. Select Modify → Histogram main menu option.

2. Right click to invoke a popup menu, and from it select Add.

3. Choose the Hazard histogram format.

4. Enter the name and description of the seismic histogram.

5. Enter the acceleration rates and frequencies. The acceleration rate is the peak ground acceleration (i.e., magnitude of the earthquake). The frequency is the probability that an earthquake that exceeds the ground acceleration will occur.

6. Press the OK button to save the new histogram.

Next, assign the histogram to the project's site hazard curve:

1. Select Modify → Project main menu option to bring up the Edit Project dialog.

2. Under the heading "Site Hazard Curves", type in the name of the seismic histogram for the "Medium" field.

Loading the Seismic Histogram Through the MAR-D Interface

The hazard curve (or histogram) may also be loaded into the SAPHIRE database using the Utility → Load and Extract main menu option (also known as the MAR-D interface). The histogram is represented in an ASCII text file and loaded into the SAPHIRE database as discussed in Appendix A. The two flat file types that are required to load the histogram using MAR-D are discussed below.

Histogram Description File (.HID)

The MAR-D flat file format for the SAPHIRE version 6 histogram description file (.HID) is shown below. (The version 7 format is the same, but can accommodate up to 24 character names, and 120 character descriptions.)

File Name:

xxxxxxxx.HID

File Format:

project =

name, type, subtype, description[, A]

where

project	16 character	Project name	
name	16 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	

An example of a histogram description file in MAR-D format is as follows:

```
SAMPLE          =
SEISMIC         , H, H, Histogram for Seismic Analysis
```

Histogram Information File (.HII)

The MAR-D data format for the SAPHIRE version 6 histogram information file (.HII) is shown below. (The version 7 format is the same, but can accommodate up to 24 character names.)

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	16 character	Project name	
NameN	16 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	

An example of a histogram information file in MAR-D format is shown below. For this example, the flat file will load seven bins with seismic hazard histogram data. For all .HII files containing seismic data, "bin1 value1" or column 1 is the earthquake frequency (per yr) and "bin1 value2" or column 2 is the mean failure acceleration of the earthquake.

```
SAMPLE, SEISMIC =
H, H
 3.680E-003, 1.000E-001
 2.980E-004, 2.000E-001
 7.200E-005, 3.000E-001
 2.620E-005, 4.000E-001
 1.170E-005, 5.000E-001
 6.000E-006, 6.000E-001
 3.360E-006, 7.000E-001
```

Event Trees

The creation of a seismic analysis model in SAPHIRE requires the development of a seismic event tree. The seismic event tree can be designed to incorporate the seismic analysis by two methods. The first method utilizes the internal basic events and fault trees assumed already present in the database. This method prioritizes and links the seismic-induced internal events and fault trees and will generate seismic sequence cut sets from the internal basic events. The second method utilizes separated seismic fault tree logic that may incorporate internal events or separate seismic events to generate the seismic cut sets. For both

methods, the seismic event tree begins with a generic seismic-initiating event set to a value of 1.0 (True Event). The actual magnitude and frequency of the earthquake of interest are identified by the user and factored into the analysis when the cut sets are generated and quantified.

The top events for the seismic event tree are those events or systems that have the potential to be induced by an earthquake. They are listed in order of severity, with the more severe-induced initiators listed first. This also addresses the potential pitfall of over-counting core damage sequences where, for example, a single earthquake induces both a large LOCA and a small LOCA at the same time. During the seismic analysis, the event tree top events are treated as seismic events with the associated seismic fragility data.

The procedure for loading or adding event trees to SAPHIRE database was discussed in Section 4.3. Identical procedures are required for the loading of the seismic event tree and any sub trees.

Fault Trees

The seismic system models (i.e., fault trees) can be created in SAPHIRE either as independent, stand-alone seismic fault trees, or they can also be integrated with the internal events analysis. To integrate seismic analysis into the internal events analysis, transformations need to be defined that convert random failures to seismic-induced failures.

Because the internal fault trees do not include several seismic related basic events, they must be added to the internal fault trees or independent seismic fault trees must be created. The procedures for loading or adding system fault trees were discussed in Section 4.5.

Basic Event Data

In most instances, seismic basic events are transformed internal basic events where the seismic considerations are implemented after the transformations. Seismic failure data are usually characterized by a median fragility and two uncertainty terms representing the random uncertainty and confidence uncertainty (Beta-R and Beta-U, respectively). See the SAPHIRE Technical Reference Manual for a more in depth discussion of seismic fragility and component failure probabilities.

The necessary steps in loading seismic basic events into the SAPHIRE program are:

1. Add the seismic event to the database including any basic event attribute data.

2. Enter the seismic failure acceleration data.

3. Enter the seismic uncertainty data.

4. Modify any internal basic events that are determined to have seismic vulnerabilities to include a seismic susceptibility "flag". This will allow for the internal basic event to be transformed into a new seismic event.

5. Enter the transformation definition to the internal basic event that is seismic susceptible.

These steps are further discussed in the following sections.

Adding Seismic Basic Events

Before the internal basic event transformation can be created, the seismic basic events must be defined. In most cases, the newly created seismic event has a different name than the internal basic event name that it is transformed from originally. For example, if the internal basic event HPI-MOV-FO-108A is determined to be seismic susceptible, then it must be transformed into a seismic event. The new seismic event could be named S-HPI-MOV-FO-108A and must be added to the database.

The procedure for adding seismic basic events and their descriptions is identical to that of internal basic events and is discussed in Section 4.6.

Loading the Seismic Failure Acceleration Data

Loading of the seismic failure data is similar to the procedures discussed for loading failure data discussed in Section 4.6. Two methods can be used to load seismic failure acceleration data. The data can be entered in the Modify → Basic Event main menu option or from basic event flat file (.BEI) and loaded through the Utility → Load and Extract main menu option as described in Appendix A. Differences between loading seismic data and the procedures discussed in Section 4.6 are outlined below.

Loading Through the Modify → Basic Event main menu option.

To enter seismic data into a seismic basic event record, go to the "Failure Data Calculation Type". Enter a "G" or an "H", which defines the basic event as a seismic basic event. Entering a "G" allows you to input an assumed g-level (earthquake strength) for use in initially generating cut sets. The "H" tells SAPHIRE to use the hazard curve identified in the “Medium” hazard curve in the Modify → Project option.

Loading Through the MAR-D Interface.

The loading of seismic failure data through the MAR-D interface is similar to the procedures described in Section 4.6 for load internal basic event failure rates. The seismic basic event flat file (.BEI) data format is similar to that in Appendix B except for the following:

1. Set the calculation type (calc) to "G" or "H" to define the basic event as a seismic event.

2. Place the Seismic Failure value in the .BEI "prob" position.

3. If a calculation type of "G" is used, specify an earthquake "G-Level". Place it in the .BEI "Lambda" position.

Loading the Seismic Uncertainty Data

Loading of the seismic uncertainty data is similar to the procedures discussed in Section 4.6. Two methods can be used to load seismic uncertainty data. The data can be entered in the Modify → Basic Events main menu option or from a basic event flat file (.BEI) and loaded through MAR-D as described in Appendix A. Differences between loading seismic data and the procedures discussed in Section 4.6 are outlined below.

Loading Through the Modify→ Basic Events main menu option

To enter seismic uncertainty data into a seismic basic event record, go to the "Uncertainty Data Calculation Type". Enter an "S", which defines the basic event as a seismic basic event. Enter the Beta-R and the Beta-U in their respected blocks.

Loading Through the MAR-D Interface.

The loading of seismic uncertainty data through the MAR-D interface is similar to the procedures described in Section 4.6 for loading internal basic event uncertainties. The seismic basic event flat file (.BEI) data format is similar to that described in Appendix B except for the following:

1. Set the uncertainty type (UdT) to "S" to allow for the implementation of seismic uncertainties.

2. Specify the seismic uncertainty term representing the random uncertainty, Beta-R. Place this value in the .BEI UdValue position. Specify the confidence uncertainty term, Beta-U, and place it in the .BEI UdValue2 position.

Defining Internal Event Susceptibility to Seismic Activity

In order to integrate the internal event analysis with a seismic analysis, the internal basic event must be transformed into the new seismic event. This process first involves defining the internal basic event as seismically susceptible. Basic event susceptibility can be entered into the SAPHIRE database through either the Modify → Basic Events main menu option or by way of a basic event attribute flat file (.BEA) loaded through the Utility → Load and Extract main menu option. Both methods are discussed below.

Defining Susceptibility Through the Modify→ Basic Events main menu option.

An internal event that is determined to be seismically vulnerable is defined in SAPHIRE as seismically susceptible. This is done under the Modify → Basic Events main menu option. Highlight the desired internal event and chose Modify from the popup menu. Select the Attributes tab and check the Seismic box in the Susceptibilities area. This will identify the basic event as susceptible to seismic initiators.

Defining Susceptibility Through MAR-D.

An internal basic event flat file (.BEA) can be generated from MAR-D as is described in Appendix A. The file format of the .BEA is described in Appendix B. To define a basic event as seismic susceptible, attribute 4 (att4) must be changed from "N" to "Y". Reloading this .BEA file with the seismic susceptible attribute is described in Appendix A.

Defining the Internal Basic Event Transformations

A transformation is a replacement or addition inside the fault tree logic. An internal event that is determined to be seismically vulnerable needs to be transformed into a new seismic event in SAPHIRE. During the transformation process, the internal basic event is replaced with a seismic basic event or a series of seismic events.

SAPHIRE utilizes three types of transformations: (1) AND, (2) OR, and (3) ZOR. An "AND" type transformation replaces the event being transformed with an AND gate having any transformed events as inputs. An "OR" type transformation replaces the event being transformed with an OR gate having any transformed events as inputs. A "ZOR" type transformation implies that if any transformed events from the original transformed event fail, then all events fail. Since for seismic analysis, an internal random basic event is transformed into one new seismic basic event, the transformation type should be "OR". This will prevent the random event and the seismic event from being "ANDed" together during the seismic analysis.

Basic event transformation also requires a "transformation level" that indicates the level of substitution for the transformation. The transformation is an integer between 0 and 255. For seismic analysis, the transformation level is generally either 0 or 1.

Transformation data can be entered into the SAPHIRE database using either the Modify → Basic Events main menu option or from a basic event transformation flat file (.BET) loaded through the Utility → Load and Extract main menu option. Both methods are discussed below.

Loading Seismic Transformations Using the Modify → Basic Event main menu

Basic event transformation is accomplished in SAPHIRE through the "Modify → Basic Events main menu option. This is done by with the following steps:

1. Highlight the desired internal event and choose “Modify” from the popup menu.
2. Select the Transformations tab.
3. Choose the transformation type (usually “OR”) and enter the transformation level (usually 0 or 1).
4. From the “All Events” list located on the left side of the dialog, highlight one or more seismic events you wish to transform the original event, and click the Add button. The selected transformation events will appear on the right side of the dialog in the “Selected Event” area. Repeat this process until all desired seismic events have been included.
5. Choose the OK button to save the changes

Loading Seismic Transformations with the MAR-D Utility

Basic event transformation may also be loaded into SAPHIRE through a MAR-D file (.BET). Below is the MAR-D file format for the SAPHIRE version 6 basic event transformation file (.BET). (The version 7 format is the same, but can accommodate up to 24 character names.)

File Name:

xxxxxx.BET

File Format:

```
project =
name1,level,type
bename1, bename2, . . . ,
. . . , benameN
^EOS
name2,level,type
bename1, bename2, . . . ,
. . . , benameN
^EOS
```

Where

Project	16 character	Project name
Name	16 character	Event name
Type	4 character	Transformation type
Level	3 character	Transformation level
bename1..N	16 character	Event name

The loading of a MAR-D flat file into SAPHIRE is described in detail in Appendix A.

GENERATING AND QUANTIFYING SEISMIC CUT SETS

Generating and quantifying seismic cut sets at both the fault tree level and the sequence level is similar to that for internal (random) analysis described in Sections 4.5.4 and 4.5.7, respectively. The few minor differences are noted below.

Generating Seismic Cut sets

When generating seismic cut sets during both fault tree and sequence analysis, you must specify that seismic analysis is desired. This is accomplished in both the "Fault Trees" and "Sequences" main menu options of SAPHIRE. To change from "Random" analysis to "Seismic" analysis, you have two options:

1. Open the Define Constants dialog found under the Utility → Define Constants main menu option. On the "General" tab, select "Seismic" from the analysis type combo box.

2. Open the "Fault Trees" or "Sequences" dialogs found under the corresponding main menu options. Select "Seismic" from the analysis type combo box located in the dialog.

Quantifying Seismic Cut sets

When quantifying seismic cut sets during both fault tree and sequence analysis, you should confirm that the "Analysis type" is set to "Seismic". In addition, after selecting "Quantify" from the popup menu option, you must choose the "G-Level" for which quantification is to be performed. The options available for "G-level" quantification include:

1. Selecting one of the g-level bins that contain a non-zero value obtained from the hazard histogram identified for use with the current project.

2. Selecting "ALL COMBINED". This gives an overall value obtained by adding the data using all bins in the histogram.

3. Selecting "ALL SEPARATE". This quantifies the cut sets at each g-level bin that contains a non-zero value obtained from the hazard histogram used with the current Family. It should be noted that after quantification using the "ALL SEPARATE" option, the cut set list for each g-level is not maintained. When quantification is completed, only the last quantification performed (at that specific g-level) is available. However, numerical results are stored and are available for each individual g-level that was calculated. These individual results are generally used during uncertainty analysis.