What Have We Done Lately?
The Current Status of the SAPHIRE Risk Analysis Software

S. Ted Wood[a*], James K. Knudsen[a], and Kellie J. Kvarfordt[a]
a - Idaho National Laboratory, Idaho Falls, United States

Abstract: The Idaho National Laboratory (INL) has been a principal developer of Probabilistic Risk Analysis (PRA) tools for over 45 years, supporting risk activities at government agencies that include the U.S. Nuclear Regulatory Commission (NRC), National Aeronautics and Space Administration (NASA), and Department of Energy (DOE). SAPHIRE software (version 8) is the latest Risk Analysis software tool capable of running on multiple processors simultaneously and is able to handle extremely complex analyses.

SAPHIRE has been designed to handle large projects, where a project may have up to 200,000 basic events and gates. These basic events and gates may be combined inside of fault trees and event trees to a maximum of 4.5 million active event tree sequences.

SAPHIRE has the capability to take a completed study and perform complex sensitivity analysis on that model while not changing the original study. These workspaces for each user interface are created and saved separately from the base model which keeps the original database intact. A new type of workspace, called a sandbox, has been developed. This paper will describe two uses of these sandboxes.

Keywords: PRA, SAPHIRE, Risk Analysis Software.

1. INTRODUCTION

SAPHIRE, or its predecessor IRRAS, has been under development since 1986 with the risk analysis software named SAPHIRE for Windows first released in 1997 as SAPHIRE 6.0. The current version of SAPHIRE is SAPHIRE 8.1.8. SAPHIRE's user interface and solving engine have not changed dramatically in recent years, but some capabilities that take advantage of more powerful computer workstations with arrays of processors and large storage capacities have been integrated into this current release. These capabilities allow a SAPHIRE user to gain insights into and understanding of their PRA models without manually attempting repetitive tasks that can be quite time consuming.

These capabilities are based upon the idea of what the SAPHIRE development team call sandboxes. A sandbox is very similar to the concept of a workspace. A workspace is a snapshot copy of the current working PRA model that is modified through basic event failure changes and logic changes to provide insights and reports. These workspaces can be saved and modified many times with no impact to the current working PRA model.

A sandbox is a new type of workspace that is a snapshot copy of the current working PRA model that is then modified in any number of ways. A sandbox is not saved like a workspace, but selected results are sent back to the current working PRA model to be stored in files that can then produce reports and provide insights. The current working PRA model's logic and basic event data is not changed, but any number of files in the publish directory are updated with new analysis results.

The power of the sandboxes is that each of them is assigned an instance of the SAPHIRE executable that performs a pre-defined solving process to obtain the requested results. These SAPHIRE sub-instances communicate with the main SAPHIRE instance via Windows messaging to transfer essential results and parameters. Two examples of these sandboxes are presented.

---

* Ted.Wood@inl.gov

2. SANDBOXES

SAPHIRE now allows users to take advantage of the power of modern computer workstations with multiple processors to perform repetitive sub-tasks on temporary versions of the working PRA model. SAPHIRE copies the necessary project files to one or more temporary sub-directories that are referred to as sandboxes. The source or working PRA model is unchanged except for the report from the task being performed. The Convergence Report process and the Plant Information e-Book (PRIB) process are examples of the use of sandboxes.

2.1 Convergence Report

The Convergence Report calculates and reports the overall Core Damage Frequency (CDF) at various truncation levels so the user may determine an appropriate truncation level for further detailed analysis. An overview of this process is given below.
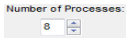
Figure 1 - Project Reports - Convergence Highlighted
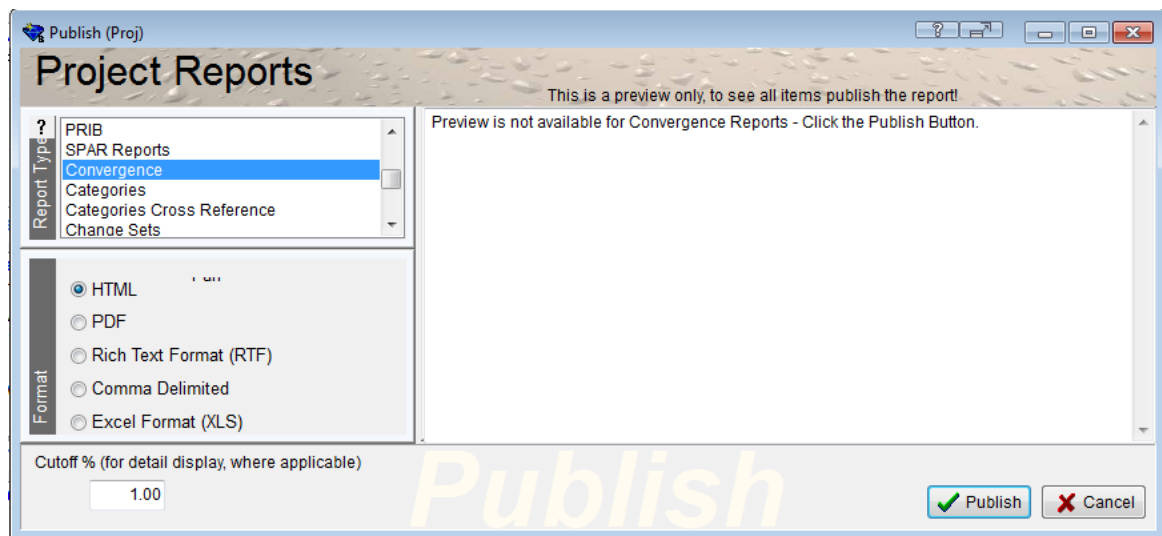
The user selects Project Reports from the main menu and highlights Convergence (see Figure 1). The user may also select the output format of the Convergence Report from the five choices: Hypertext Markup Language (HTML), Portable Document Format (PDF), Rich Text Format (RTF), Comma Separated Value (CSV), and Microsoft Excel Spreadsheet (XLS). Clicking the Publish button displays the Convergence Report form as shown in Figure 2.

Figure 2 - Convergence Report Setup
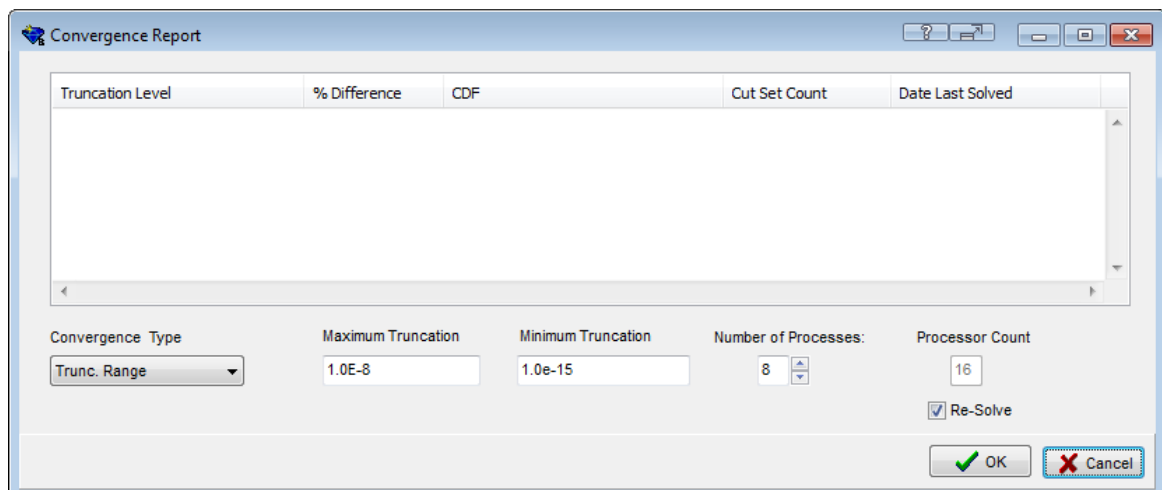
The user can select the number of processes (sandboxes) desired by entering a number in the edit box or clicking on the up/down arrows in the [Number of Processes: 8] area. The Processor Count [16] shows the number of processors available on the users system. It is not recommended that the user exceed the number of processors available and no more than 32 processes are allowed. The Re-Solve check box [✓ Re-Solve] is automatically checked.

The Convergence Type combination box (see Figure 3) allows the user to select from a number of options.

Figure 3 - Convergence Type
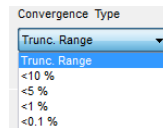
The default option is "Trunc. Range", or truncation range, which allows the user to select the maximum and minimum truncation values. The convergence process will then calculate the overall CDF for each decade of truncation including the maximum and minimum. In the example above, the truncation values are 1.0E-8, 1.0E-9, 1.0E-10, 1.0E-11, 1.0E-12, 1.0E-13, 1.0E-14, and 1.0E-15. If the maximum or minimum is not a 1.0E-x (decade) value, the software increments the value to the next highest decade that exceeds the maximum or minimum. If, for example, the maximum is set to 5.0E-8, the high decade becomes 1.0E-7. If the minimum is set to 5.0E-11, the minimum decade is 1.0E-10.

Other possible options in the Convergence Type combination box allow the user to designate a floor percent increase that will stop the processing if that floor is achieved. These hard coded floors are "< 10 %", "< 5%", "< 1%", and "< 0.1%". The "% Difference" is defined as:

$$\% \text{ Difference} = ((CDF_n - CDF_{n-1}) / CDF_{n-1}) * 100.0 \qquad (1)$$

where CDF_n is the overall CDF for the current truncation level and
            CDF_{n-1} is the overall CDF for the previous truncation level.

The maximum and minimum truncation are used to start each of the processes up, but if the "% Difference" between truncation levels is less than the selected floor, the processes are stopped, and the Convergence Report will only show valid values for the completed truncation levels.

Once the user has entered the desired parameters, she presses the [✓ OK] button and the task begins. In this example, eight processes where chosen with a truncation range of 1.0E-8 through 1.0E-15 or one truncation level per sandbox process.

The Windows Task Manager screen (see Figure 4) shows a snap shot of the state of the workstation when eight processors (one per sandbox) are selected. The current PRA model is copied to eight sub-directories and eight sandbox instances of SAPHIRE are started.

Figure 4 - Task Manager
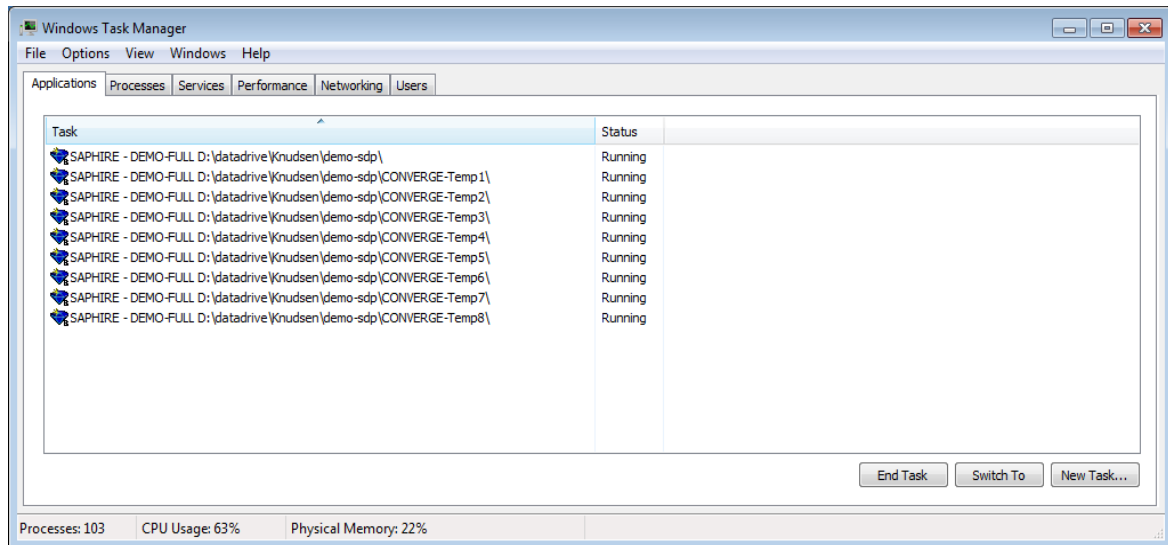
Each of the SAPHIRE sandbox sub-instances is started with an assigned project sub-directory and is running a macro (SAPHIRE defined command language) that waits for a Windows message from the main SAPHIRE instance to begin the solve process. The solve process is defined as: (1) Generate cut sets, (2) Apply post-processing rules, and (3) Perform a cut set update on all the main level linked event trees. Since a given sandbox may perform this run at several truncation levels, the truncation level is passed to each sandbox SAPHIRE instance via a Windows message. Each sandbox SAPHIRE instance performs the solve process at the designated truncation level, sends the results (CDF, cut set count, and truncation level) back to the main SAPHIRE instance via a Windows message, and waits for instructions from the master SAPHIRE instance. The master SAPHIRE instance retains the results and checks if all solve processes for all the truncation levels have been requested or sent to a sandbox. If all solve processes for each desired truncation levels have been requested, the master SAPHIRE instance instructs each waiting sandbox SAPHIRE instance to close itself and deletes the temporary data files contained in the sandbox sub-directory and that sub-directory. When all solve runs for each requested truncation levels have been completed, the retained results are then compiled and placed into the Convergence Report (see Figure 5). The Converge_Insights.txt file contains the results of the last convergence process and is stored in the publish sub-directory of the master SAPHIRE model.

If a Convergence Report has already been created, a Converge_Insights.txt file exists. This file is used to store the last performed convergence information. If the convergence process has been recently performed and no changes to the model have been made, the user can select different output formats, un-check the Re-Solve check box, and click the OK button and the Convergence Report will be output into the desired output format.

Figure 5 - Convergence Report
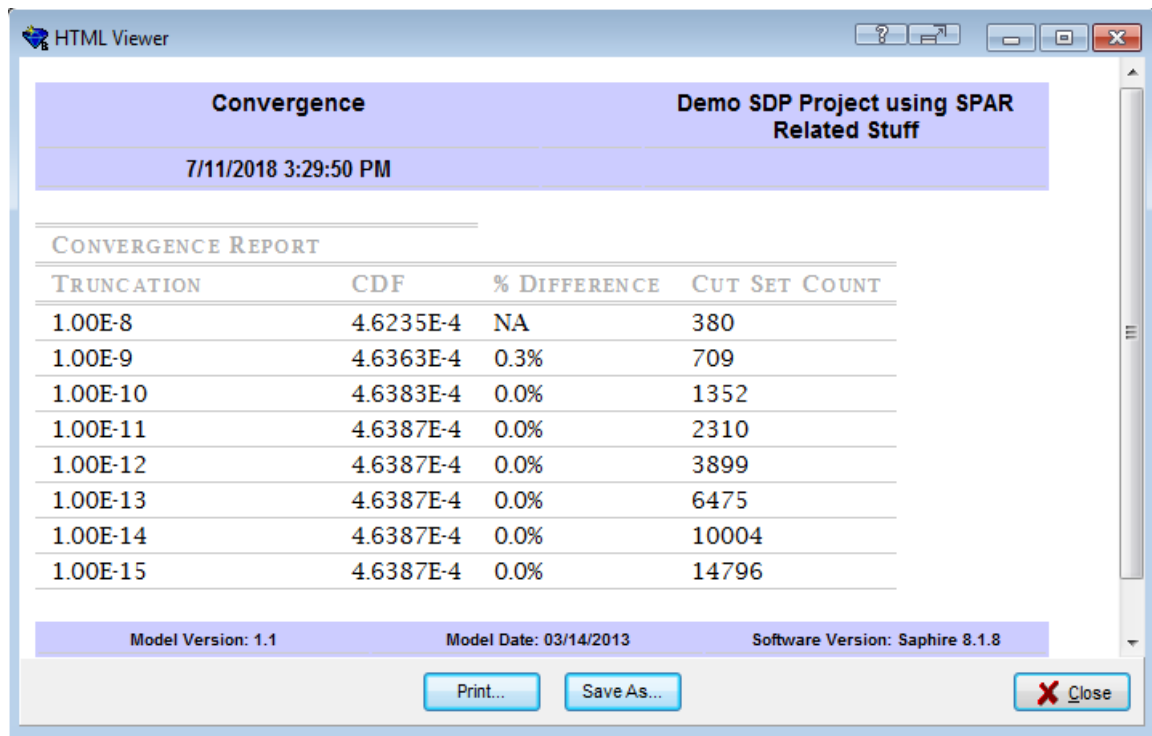
To demonstrate the power of sandboxes and show the impact of the sandbox setup compared to the time required to complete the overall process, ten convergence tests were performed. These tests were performed on a Dell Precision Tower 5810 with 8 hyper-threaded Intel® Xeon® E5-1680 v3 @ 3.2GHz processors and 32 GB of memory with a 4 TB Western Digital Hard Drive having a model type of WD40EZRX-75S. The Convergence Reports timing tests were performed for a truncation range of 1.0E-4 to 1.0E-21 and those results are shown in Table 1with a representative report shown in Figure 6. It is interesting to note that the odd number of sandboxes appears to be faster than even number of sandboxes. The two truncation levels that require the most time are 1.0E-20 and 1.0E-21 and the odd number of sandboxes start processing them sooner and they complete in a timely manner. The four sandbox group appears to start them later and thus increases the total processing time.

Table 1 - Convergence Timing Tests

Sandbox Count	Setup Time (mm:ss)	Total Time (mm:ss)
1	0:03	4:01
2	0:05	1:25
3	0:08	1:09
4	0:10	1:30
5	0:12	0:50
6	0:15	1:03
7	0:17	0:51
8	0:20	0:54
12	0:30	1:05
16	0:39	1:20

Figure 6 - Timing Convergence Report

Convergence		Demo SDP Project using SPAR Related Stuff
7/13/2018 10:45:44 AM		

CONVERGENCE REPORT

TRUNCATION	CDF	% DIFFERENCE	CUT SET COUNT
1.00E-4	1.5010E-4	NA	1
1.00E-5	2.8474E-4	47.3%	8
1.00E-6	4.1814E-4	31.9%	39
1.00E-7	4.5467E-4	8.0%	137
1.00E-8	4.6235E-4	1.7%	380
1.00E-9	4.6363E-4	0.3%	709
1.00E-10	4.6383E-4	0.0%	1352
1.00E-11	4.6387E-4	0.0%	2310
1.00E-12	4.6387E-4	0.0%	3899
1.00E-13	4.6387E-4	0.0%	6475
1.00E-14	4.6387E-4	0.0%	10004
1.00E-15	4.6387E-4	0.0%	14796
1.00E-16	4.6387E-4	0.0%	20469
1.00E-17	4.6387E-4	0.0%	27158
1.00E-18	4.6387E-4	0.0%	35561
1.00E-19	4.6387E-4	0.0%	45014
1.00E-20	4.6387E-4	0.0%	55265
1.00E-21	4.6387E-4	0.0%	65629

2.2 PRIB Report

The PRIB report on Standardized Plant Analysis Risk (SPAR) models works in a similar fashion with multiple sandboxes of SAPHIRE calculating system and component type importance measures with the copied sandbox models.

The SPAR Models have been set up with System and Component Type Categories as shown in the Categories Report (see Figure 7). Each basic event in the project can be assigned one of the System and one of the Component Type categories as shown in the report example in Figure 8. Figure 9 shows the basic event editing form where these assignments can be made on an individual basis. Through the Load/Extract (MAR-D) functionality, wholesale updates to the categories and basic event category assignments may be performed.

This assignment then allows SAPHIRE to calculate the "importance" of selected systems and selected system/component types pairs by creating temporary change sets that turn on (set to "TRUE") or off (set to "FALSE") all basic events with the desired systems or system/component type pairs. This process then calculates an overall CDF with all matching events for the System/Component Type set to "TRUE" and repeats the process with them set to "FALSE". This difference in the overall CDF is treated like a Birnbaum Importance Measure, which indicates the sensitivity of the minimal cut set upper bound with respect to a change in the probability of a group of basic events.

$$Birnbaum = F(1) - F(0) \qquad (2)$$

where   F(1) is the minimal cut set upper bound with the group probability set equal to 1.0 and
        F(0) is the minimal cut set upper bound with the group probability set equal to 0.0.

Each system and system/component type pair overall CDF values and differences are stored in the
PRIB Insights Reports.txt in the publish sub-directory of the master PRA model.
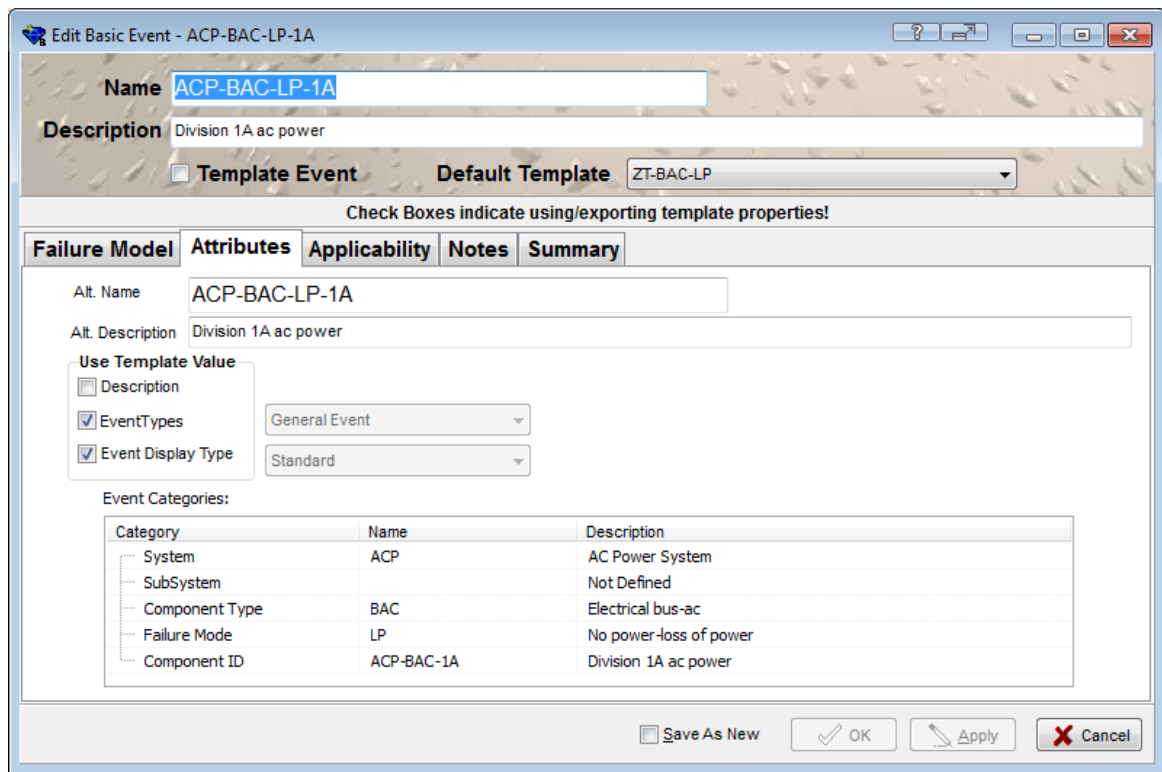
Figure 7 – Categories Report

Figure 8 - Category Cross-Reference
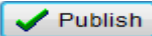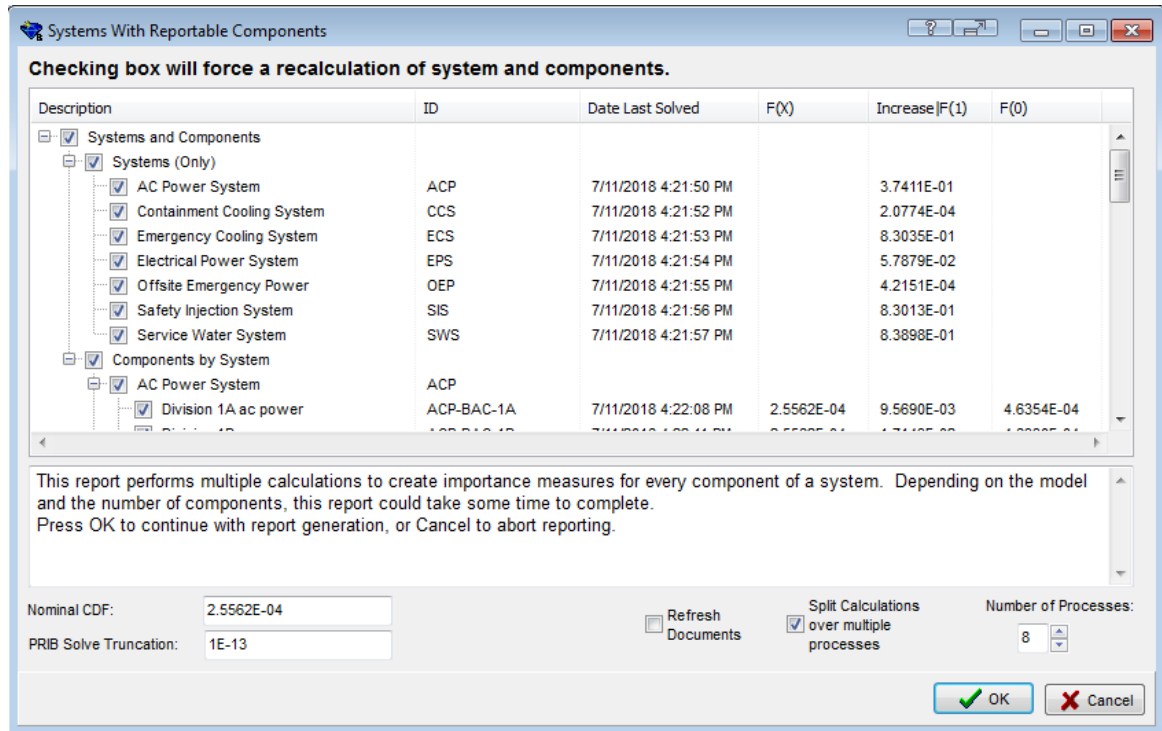
Figure 9 - Basic Event Category Editor

Figure 10 - Project Reports
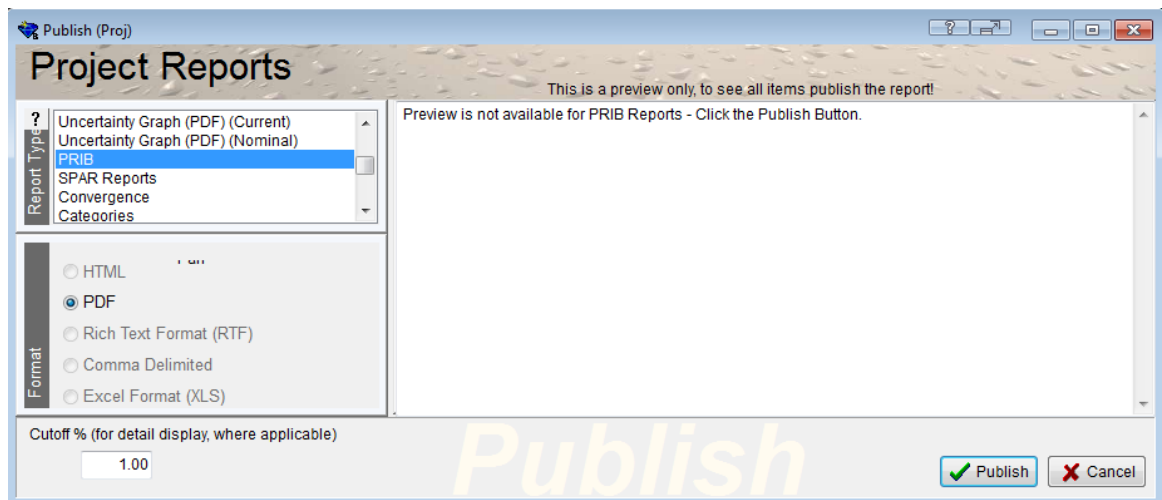
The user selects Project Reports from the main menu and highlights PRIB (see Figure 10). The PRIB Report is only available as a Portable Document Format (PDF). Clicking the ✔ Publish button displays the PRIB Report form as shown in Figure 11.

**Figure 11 - Systems with Reportable Components**

The user can select the number of processes (sandboxes) desired by clicking

 the Split Calculations check box and entering a number or clicking on the up/down arrows in the Number of Processes field. The user can also adjust the PRIB Solve Truncation value to a different value than the default is desired.

Once the user has entered the parameters desired, she presses the button and the PRIB task begins. In this example, eight processes where chosen.

The Windows Task Manager screen (see Figure 12) shows a snap shot of the state of the workstation when eight processors (one per sandbox) are selected. The current PRA model is copied to eight sub-directories and eight sandbox instances of SAPHIRE are started.

Figure 12 – Windows Task Manager
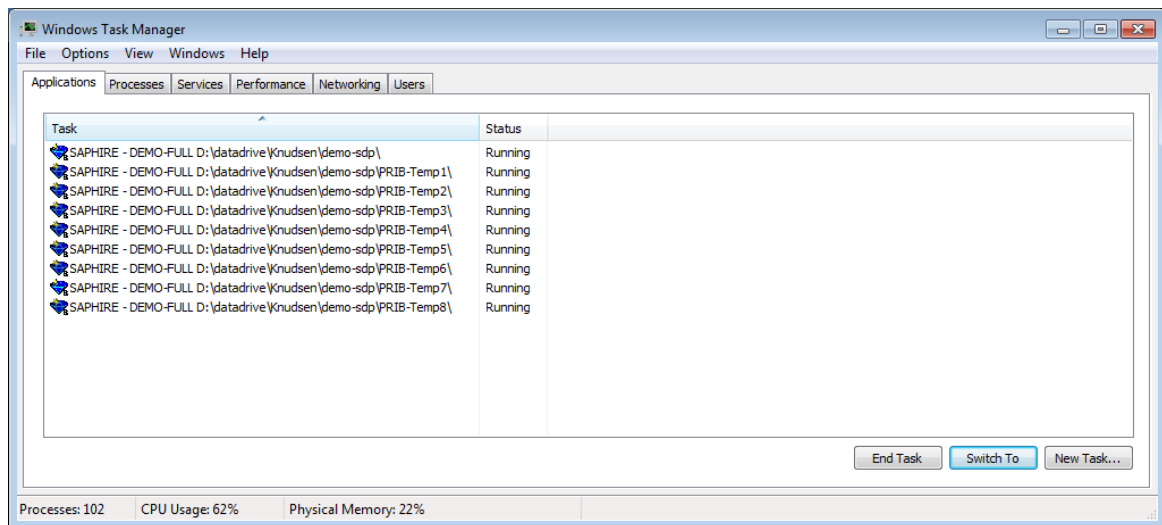
Each of the SAPHIRE sandbox instances is started with an assigned project sub-directory and is running a macro (SAPHIRE defined command language) that waits for a Windows message from the main SAPHIRE instance to begin the solve process. The solve process is different depending if it is a system level or system/component type level importance.

The solve process for system level importance is defined as: (1) Set all basic events that belong to the System to "TRUE", (2) Generate cut sets, (3) Apply post-processing rules, (4) Perform a cut set update on all the main level linked event trees, and (5) Save the overall CDF.

The solve process for system/component type level is defined as: (1) Set all basic events that belong to the System/Component Type pair to "TRUE", (2) Generate cut sets, (3) Apply post-processing rules, (4) Perform a cut set update on all the main level linked event trees, (5) Save the overall CDF, (6) Set all basic events that belong to the System/Component Type pair to "FALSE", (7) Generate cut sets, (8) Apply post-processing rules, (9) Perform a cut set update on all the main level linked event trees, and (10) Save the overall CDF.

Since a given sandbox may perform this run for any number of systems or system/component type pairs, the desired system or system/component type pair is passed to each sandbox SAPHIRE instance via a Windows message. Each sandbox SAPHIRE instance performs the solve process for the desired system or system/component type pair, sends the results (CDF, cut set count, and truncation level) back to the main SAPHIRE instance via a Windows message, and waits for instructions from the master SAPHIRE instance. The master SAPHIRE instance retains the results and checks if all solve processes for all desired/marked entries have been requested or sent to a sandbox. If all solve processes for each marked entries have been requested, the master SAPHIRE instance instructs each waiting sandbox SAPHIRE instance to close itself. The master SAPHIRE instance then deletes the temporary data files contained in the sandbox sub-directory and that sub-directory. When all solve processes have completed, the retained results are then compiled and placed into the PRIB Report (not shown at the request of the NRC). The PRIB_Insights.txt file contains the results of the last PRIB process and is stored in the publish sub-directory of the master SAPHIRE model.

If a PRIB Report has already been created, a PRIB_Insights.txt file exists. This file is used to store the last performed importance information.

Figure 13 shows an example of a SAPHIRE sandbox instance display. The caption displays the sandbox directory and the text to the right of the clock shows which system/component type pair is being processed along with the counter of which component it is of the total components for the

system. The text to the right of the system/component pair name shows where in the solve process this instance is. The text just beneath the caption indicates the overall solve process and how many times this sandbox has been used. In this case, this system/component type pair is the 12th one to be solved by this sandbox and the process is in the midst of applying the cut set post-processing rules to the cut sets of the TRANS event tree.

To demonstrate the power of sandboxes and show the impact of the sandbox setup compared to the time required to complete the overall process, ten PRIB tests were performed. These tests were performed on a Dell Precision Tower 5810 with 8 hyper-threaded Intel® Xeon® E5-1680 v3 @ 3.2GHz processors and 32 GB of memory with a 4 TB Western Digital Hard Drive having a model type of WD40EZRX-75S. The PRIB Reports timing tests were performed for all systems and all system/component pairs and those results are shown in Table 2. It is interesting to note that at a certain point, the setup time starts to have a negative impact on the overall time.

Figure 13 - Sample of a Sandbox SAPHIRE Instance Display
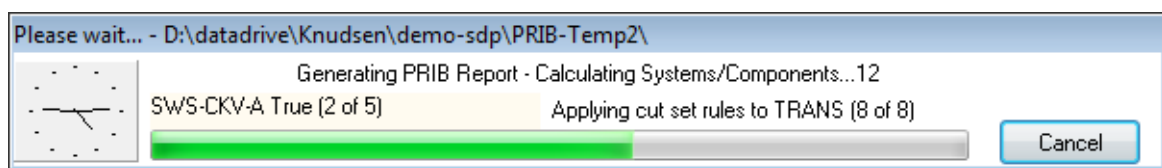
Table 2: PRIB Importance Timing Tests

Sandbox Count	Setup Time (mm:ss)	Total Time (mm:ss)
1	0:02	14:00
2	0:08	7:20
3	0:11	4:50
4	0:14	3:30
5	0:18	3:02
6	0:21	2:51
7	0:24	2:53
8	0:29	3:13
12	0:43	3:42
16	1:00	4:10

3. CONCLUSION

SAPHIRE's user interface and solving engine have not changed dramatically in recent years, but some capabilities that take advantage of more powerful computer workstations with arrays of processors and large storage capacities have been integrated into this current release, SAPHIRE 8.1.8. These capabilities have been named sandboxes by the software development team and allow a SAPHIRE user to gain insights and understanding of their PRA models without manually attempting repetitive tasks that are time consuming. Two examples of the sandbox capability have been presented. The sandbox concept will be integrated into other areas of SAPHIRE to speed up other tasks and assist users in their use of SAPHIRE as they analyze their PRA models.

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 of their employees, 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 report, or represents that its use by such third party would not infringe privately owned rights. The views expressed in this paper are not necessarily those of the U.S. Nuclear Regulatory Commission.