SANDIA REPORT
SAND2007-3368
Unlimited Release
Printed June 2007

A User's Guide for INPAGN_Launcher.DLL

Patrick D. Mattie and Donald A. Kalinich

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2007-3368
Unlimited Release
Printed June 2007

A User's Guide for INPAGN_Launcher.DLL

Patrick D. Mattie
Total System Performance Assessment Department, Org. 6784

Donald A. Kalinich
Analysis and Modeling Department, Org. 6762

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS0776

Abstract

Sandia National Laboratories (Sandia), a U.S. Department of Energy National Laboratory, has over 30 years experience in the assessment of radioactive waste disposal and at the time of this publication is providing assistance internationally in a number of areas relevant to the safety assessment of radioactive waste disposal systems. In cooperation with the Republic of Taiwan's Institute of Nuclear Engineering and Research (INER), Sandia National Laboratories (SNL) has developed software that provides an interface between a deterministic mass transport code and GoldSim™ (a commercial software used to conduct Monte Carlo analyses). The SNL developed software enables INER to perform probabilistic simulations for safety analysis and performance assessment of geologic disposal of commercial spent nuclear fuel. The following report details the software design, the steps necessary to use the software, and presents an example application of the paradigm of coupling deterministic codes to a contemporary probabilistic software application.

ACKNOWLEDGMENTS

This report documents a major effort as part of a technology transfer program funded by the Taiwan ICP Phase II - Technology Transfer for Interim Dry Storage and Final Disposal of Spent Nuclear Fuel under a WFO Funds-in agreement contract between Sandia and Lockheed Martin. The Sandia team that worked on Phase II of this ICP Technology Transfer Project would like to thank Ms. Maureen Mahowald of Lockheed Martin Maritime Systems and Sensors (LM MS2) and Mr. Michael Hollmen of Lockheed Martin Industrial Participation for their funding and support, and Taiwan's Industrial Development Bureau (IDB), and Industrial Cooperation Program (ICP) Office for their support of this ICP Project. This work supports DOE/RW international leadership in geologic disposal of spent nuclear fuels and radioactive waste and working item IN-SNL-DD23, "Technology Transfer for Geologic Repository Science and Performance Assessment", under AIT/TECRO Joint Standing Committee for Civil Nuclear Cooperation.

A number of people at Sandia National Laboratories including, Hong-Nian Jow (Org. 6773) (Project Lead), F. Joseph Schelling (Org. 6772), and Cedric Sallaberry (Org. 6784) have made significant contributions to the technology transfer program which either directly or indirectly supported the development of the software documented in this report. In addition, Jerry McNeish (Org. 6784) provided an internal technical review of this report.

The Sandia team sincerely appreciates the support of the management and technical staff at the Institute of Nuclear Energy Research (INER) in Taiwan for their generous support during Sandia's visits to INER and their effort's in providing technical support to Sandia for the software development project. Specifically, Wen-Shou Chuang (Project Lead), Fu-Lin Chang, Shin-Jon Ju, and Ming-Chuan Kuo supported the development through an iterative process of software development in support of their program efforts.

Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

CONTENTS

FIGURES

TABLES

NOMENCLATURE

DOE	Department of Energy
INER	Institute of Nuclear Energy Research
INPAG-N	INER Performance Assessment Model for the Geologic Repository of Spent Fuel – Near Field Transport
INPAG-NV2	INER Performance Assessment Model for the Geologic Repository of Spent Fuel – Near Field Transport Version 2
INPAG-NH	INER Performance Assessment Model for the Geologic Repository of Spent Fuel – Near Field Transport - Horizontal Emplacement
LAN	Local Area Network
LHS	Latin Hypercube Sampling
SNF	Spent Nuclear Fuel
SNL	Sandia National Laboratories

1. INTRODUCTION

1.1. Purpose

The objective of the INPAGN_Launcher.DLL User's Guide is to provide documentation of the capabilities of the dynamically linked library (DLL) software as well as document the code used to create the application. The features of the DLL as well as required input and output information will be presented. An example model is included in Appendix A and should be used as an aide to using the INPAGN_Launcher.DLL.

1.2. Software Identification

This user's guide documents the dynamically linked library software INPAGN_Launcher_v3.DLL. INPAGN_Launcher_v3.DLL discussed in this manual has been designated as open-source and is freely available. It can be accessed on a Sandia National Laboratories website at the following URL: http://www.sandia.gov/icp/.

2. USER INFORMATION

2.1. Introduction

INPAGN_Launcher software has been designed to interface between the commercial software GoldSim.exe (GoldSim Technology Group LLC, 2006) and INER's near field radionuclide mass transport code INPAG-N.EXE (Monitor Scientific, 2000). The INPAG-N code simulates the source term and near field radionuclide mass transport for waste packages containing commercial spent nuclear fuel (SNF). The INPAG-N code was coupled with GoldSim™ to create the framework for a probabilistic model capable of evaluating uncertainties in a potential SNF repository system. This document is intended as a user's guide for the INPAGN_Launcher software and the model file GoldSim™ is needed to execute a probabilistic simulation.

The INPAGN_Launcher software is called in the form of a direct link library (DLL) by GoldSim.exe during an execution of a GoldSim™ model file specifically designed for the purpose of running INPAG-N for a probabilistic simulation. All related libraries needed by the DLL are linked with it.

The INAPGN_Launcher is programmed in the FORTRAN 90 language, and developed with Compaq Visual FORTRAN Professional Edition 6.6a and is implemented on the Windows XP operating system.

2.2. How to Use this Software

This software is called from a GoldSim™ model file. It is simply placed in a directory accessible to the GoldSim™ model file. Execution control of the DLL is provided by the GoldSim™ software (GoldSim Technology Group LLC, 2006, Appendix C). It is assumed the user has a basic proficiency with GoldSim™ and the necessary understanding of the features of the GoldSim™ code to operate this DLL. The GoldSim™ model file itself should be considered as part of the model documentation. In addition, a user must understand the features of the INPAG-N software (Monitor Scientific, 2000).

Since this software is a DLL, there is little direct interaction required between the user and the software. The GoldSim™ model file supplies the 13 data input blocks necessary for creating an INPAG-N.EXE input file to the INPAGN_Launcher DLL and receives the INPAG-N.EXE calculation results from the DLL. The user has access to the INPAGN_Launcher inputs and INPAG-N.EXE flux outputs through the GoldSim™ model. An example GoldSim™ model discussed in this manual has been designated as open-source and is freely available. The example model file is included with the report and can also be accessed on a Sandia National Laboratories website at the following URL: http://www.sandia.gov/icp/.

Figure 1 shows a flow chart of the operations carried out by the INPAGN_Launcher DLL. The INPAGN_Launcher requires that an array of input parameter values for INPAGN.EXE are passed from a GoldSim™ model file linked to the INPAGN DLL. The INPAGN_Launcher DLL is capable of running five different versions of the INPAG-N Code (Section 2.3.3). The input array should be defined using a GoldSim™ model as specified in the Table 1, and discussed in Section 2.3 Input Specification. When a GoldSim™ model is executed, the input array will be

passed to the INPAGN_Launcher DLL. The INPAGN_Launcher DLL will select the appropriate subroutines to execute based upon the selected version of INPAG-N which is being run. The DLL parses the input array and writes an INPAG-N input file to the local directory. The DLL will call the selected version of INPAG-N and execute the code using the generated input file. After the INPAG-N code has completed its simulation, the DLL will read selected output files and pass the output back to the GoldSim™ model file where it is loaded into table elements. These steps are repeated sequentially for each realization for a probabilistic simulation.

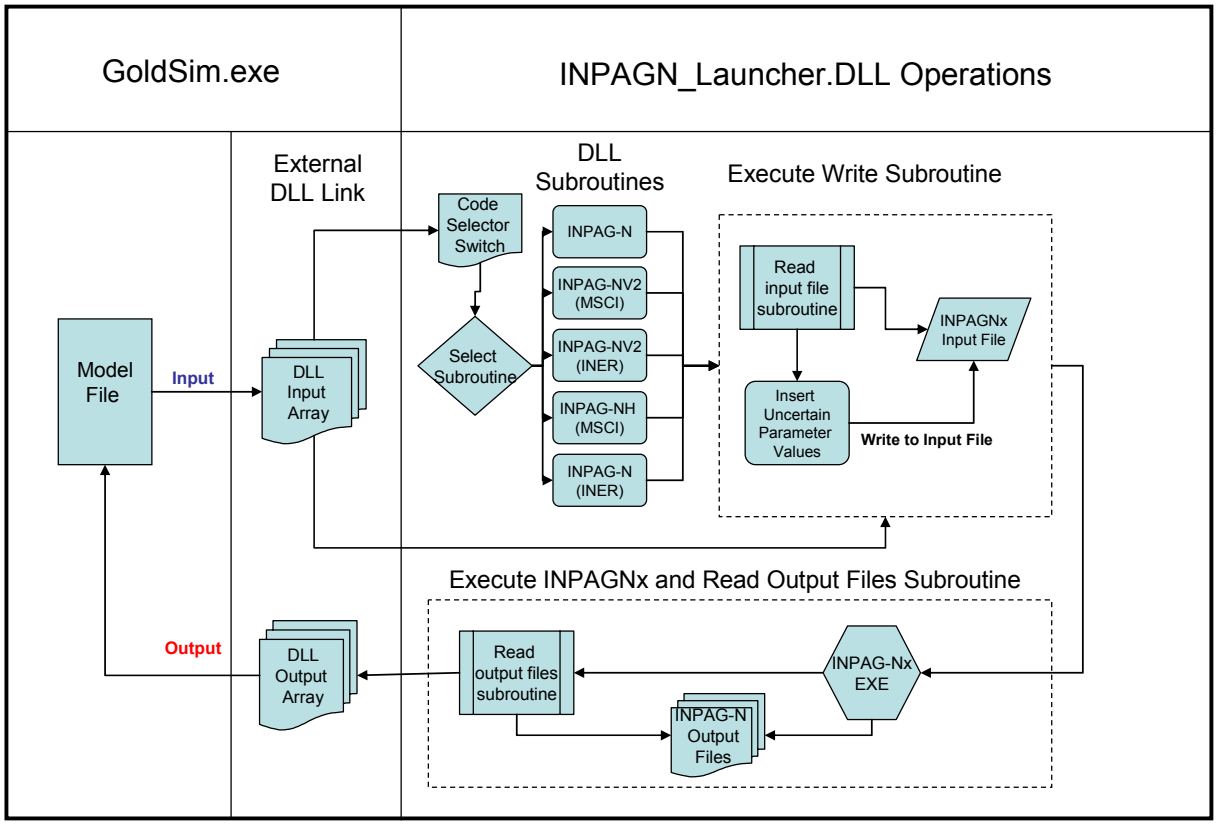

Figure 1: Flow Chart for INPAGN_Launcher

2.3. Input Specification

The inputs to the INPAGN_Launcher DLL come either from the "RN.DAT" file or from the in() array passed by a GoldSim™ model. The values passed to the DLL via the in() array can be single values or arrays of data. Table 1 lists the input array in() expected by the DLL.

The input values may be constants or stochastics that will be sampled once per realization during a probabilistic simulation.

2.3.1. INPAGN_Launcher Input Array: in()

An in() array needs to be defined in the GoldSim™ model file and passed to the INPAGN_Launcher DLL during execution of the code. Figure 2 shows an example DLL External Element (GoldSim User's Guide, GoldSim Technology Group LLC, 2006), which is used in the GoldSim™ model file to define a link between the GoldSim.exe and a DLL. The external DLL element requires the following inputs: 1) name of the DLL to be called [INPAGN_Launcher_v3.DLL], 2) the function being called [launch_inpagn], and 3) the input array [in()]. In addition, three outputs have been defined and are discussed in Section 2.4, Output Specification.

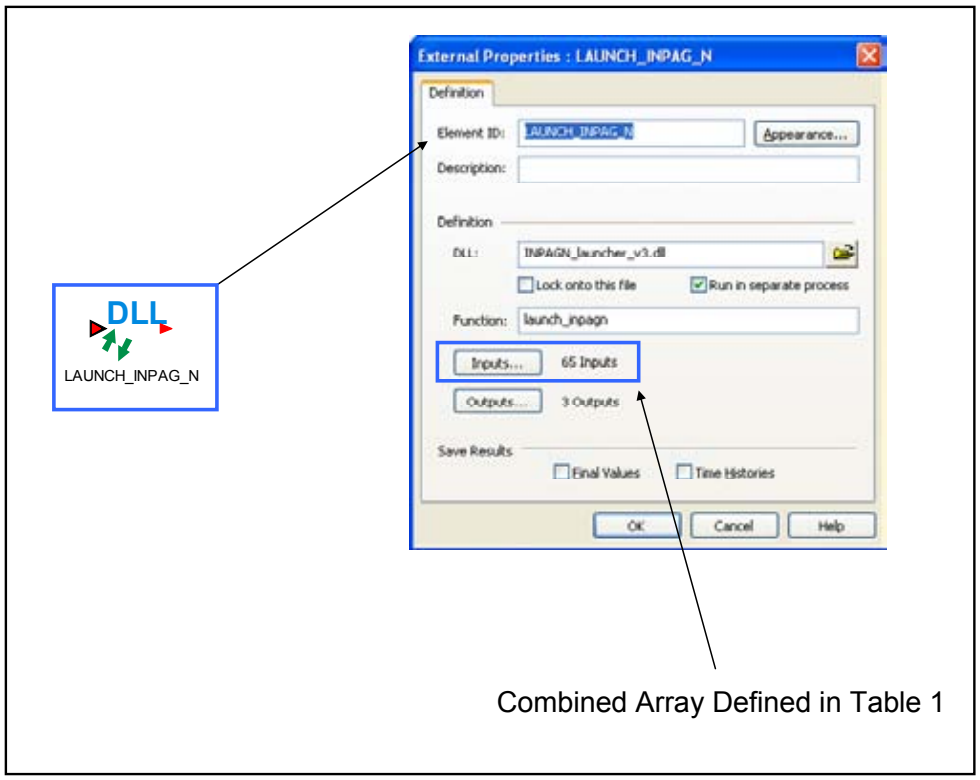

Figure 2: External DLL Element in GoldSim™ linking in() array to INPAGN_Launcher DLL

The input array [in()] is defined in GoldSim™ and must contain all of the elements listed in the "Combined Array" defined in Table 1. The combined array includes all of the input parameters required for all five versions of INPAG-N. The first three elements in the array include a parameter that identifies the version of INPAG-N which is to be run, the realization number, and the number of radionuclides in the simulation. The remaining elements in the array make up the input parameter values for each of the 13 input blocks required for an INPAG-N input file. Figure 3 illustrates the 13 input blocks as defined in the example GoldSim™ input file. Each container in the example GoldSim™ file (shown in Figure 3), contains the input parameters listed in Table 1.

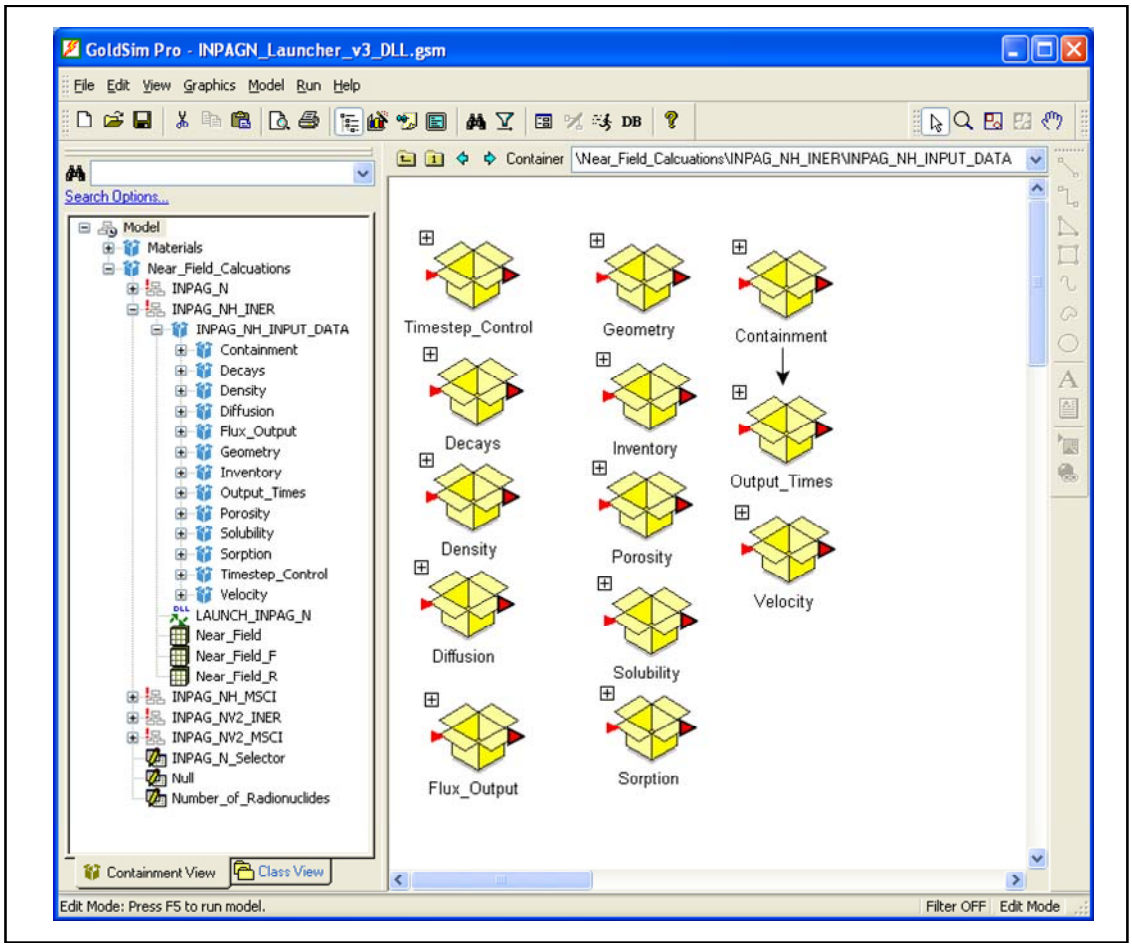

Figure 3: Example GoldSim™ model file with 13 Input Blocks needed for INPAG-N input.

The input blocks and parameters are listed in Table 1 in the order that the DLL expects to receive them. The DLL reads the entire array and selects the values needed for the selected version of INPAG-N. The DLL accepts a value between 1 and 5 for the version of INPAG-N (1=INPAGN.EXE, 2=INPAGNV2.EXE, 3=INPAGNH.EXE, 4=INPAGNV2I.EXE (INER version), and 5=INPAGNH.EXE (INER version)). The DLL ignores the values in the array that are not used for a given version of the INPAG-N code. The elements in the in() array that are ignored by the DLL vary with each INPAG-N version. Table 1 lists how the in() array is read by

ignored by the DLL vary with each INPAG-N version. Table 1 lists how the in() array is read by the DLL for each INPAG-N version, elements of the array that are ignored are listed as NULL. It is important to note that several of the input parameters shown in Table 1 are vectors and include a number of elements equal to the number of radionuclide species. Thus, the total size of the input array will change with the number of radionuclides defined in the simulation.

Table 1: INPAGN_Launcher.DLL Input Array

BLOCK	Combined Array	INPAG_N(MSCI)	INPAG_NV2(MSCI)	INPAG_NH(MSCI)	INPAG_NV2(INER)	INPAG_NH(INER)	Index	Type
N/A	CONTROL VALUES	INPAG_N_Selector	INPAG_N_Selector	INPAG_N_Selector	INPAG_N_Selector	INPAG_N_Selector	1	Value
		Realization	Realization	Realization	Realization	Realization	2	Value
		Number_of_RN	Number_of_RN	Number_of_RN	Number_of_RN	Number_of_RN	3	Value
CONTAINMENT	FAIL-TIME	Fail_Time	Fail_Time	Fail_Time	Fail_Time	Fail_Time	4	Value
	INV-START-TIME	Inv_Start_Time	Inv_Start_Time	Inv_Start_Time	Inv_Start_Time	Inv_Start_Time	5	Value
	ALTER-TIME	Alter_Time	Alter_Time	Alter_Time	Alter_Time	Alter_Time	6	Value
TIMESTEP-CONTROL	TOLERANCE	Tolerance	Tolerance	Tolerance	Tolerance	Tolerance	7	Value
	MIN-TIMESTEP	Min_Timestep	Min_Timestep	Min_Timestep	Min_Timestep	Min_Timestep	8	Value
DECAYS	DECAY RATES	Decay_Rate	Decay_Rate	Decay_Rate	Decay_Rate	Decay_Rate	9	Vector
DENSITY	CANISTER	Canister	Canister	Canister	Canister	Canister	13	Value
	BENTONITE	Bentonite	Bentonite	Bentonite	Bentonite	Bentonite	14	Value
	ROCK	Rock	Rock	Rock	Rock	Rock	15	Value
	EDZ	Null	Null	Null	EDZ_Density	Null	16	Value
	TUNNEL	Null	Null	Null	Tunnel Density	Null	17	Value
DIFFUSION	CANISTER	Canister_Diffusion	Canister_Diffusion	Canister_Diffusion	Canister_Diffusion	Canister_Diffusion	18	Value
	BENTONITE	Bentonite_Diffusion	Bentonite_Diffusion	Bentonite_Diffusion	Bentonite_Diffusion	Bentonite_Diffusion	19	Value
	EDZ	EDZ_Diffusion	EDZ_Diffusion	EDZ_Diffusion	EDZ_Diffusion	EDZ_Diffusion	20	Value
	ROCK-FRACTURE	Rock_Fracture	Rock_Fracture	Rock_Fracture	Rock_Fracture	Rock_Fracture	21	Value
	ROCK-MATRIX	Null	Rock_Matrix_Diff	Rock_Matrix_Diff	Rock_Matrix_Diff	Rock_Matrix_Diff	22	Value
	TUNNEL	Null	Tunnel_Diff	Null	Tunnel Ddiff	Null	23	Value
FLUX-OUTPUT	UNITS MOLPYR	Units	Units	Units	Units	Units	24	Value
GEOMETRY	INNER-CAN-RAD	Inner_Can_Rad	Inner_Can_Rad	Inner_Can_Rad	Inner_Can_Rad	Inner_Can_Rad	25	Value
	OUTER-CAN-RAD	Outer_Can_Rad	Outer_Can_Rad	Outer_Can_Rad	Outer_Can_Rad	Outer_Can_Rad	26	Value
	BENT-RAD	Bent_Rad	Bent_Rad	Bent_Rad	Bent_Rad	Bent_Rad	27	Value
	EDZ-RAD	EDZ_Rad	EDZ_Rad	Null	EDZ_RAD	EDZ_RAD	28	Value
	LENGTH	Length	Length	Null	Length	Null	29	Value
	EDZ-THICK	Null	Null	EDZ_Thick	Null	Null	30	Value
	CAN-LENGTH	Null	Null	Can_Length	Null	Can_Length	31	Value
	FRACTURE-SPACING	Fracture_Spacing	Fracture_Spacing	Fracture_Spacing	Fracture_Spacing	Fracture_Spacing	32	Value
	FRACTURE-APERTURE	Fracture_Aperture	Fracture_Aperture	Fracture_Aperture	Fracture_Aperture	Fracture_Aperture	33	Value
	NO-BENT-BOX	No_Bent_Box	Null	Null	No_Bent_Box	No_Bent_Box	34	Value
	BENT-TOP-THICK	Null	Bent_Top_Thick	Null	Bent_Top_Thick	Null	35	Value
	TUNNEL-HEIGHT	Null	Tunnel_Height	Tunnel_Height	Tunnel_Height	Null	36	Value
	ROCK-SIDE-THICK	Null	Rock_Side_Thick	Null	Rock_Side_Thick	Null	37	Value
	ZERO-CONC-LENGTH	Null	Zero_Conc_Length	Null	Zero_Conc_Length	Zero_Conc_Length	38	Value
	FRACTURE-SORB-DEPTH	Null	Fracture_Sorb_Depth	Fracture_Sorb_Depth	Fracture_Sorb_Depth	Fracture_Sorb_Depth	39	Value
	BENT-BOT-THICK	Null	Bent_Box_Thick	Null	Bent_Box_Thick	Null	40	Value
	ROCK-BOT-THICK	Null	Rock_Bot_Thick	Null	Rock_Bot_Thick	Null	41	Value
	BOREHOLE-PITCH	Null	Null	Borehole_Pitch	Null	Borehole_Pitch	42	Value
	CAN-DIS	Null	Null	Can_Dis	Null	Can_Dis	43	Value
	NO-OF-CAN	Null	Null	No_of_Can	Null	No_of_Can	44	Value
	TUNNEL-FLOW-THICK	Null	Null	Tunnel_Flow_Thick	Null	Tunnel_Flow_Thick	45	Value
	NF-ROCK-THICK	Null	Null	Null	Null	NF_Rock_Thick	46	Value
INVENTORY	INVENTORY	Species_Inventory	Species_Inventory	Species_Inventory	Species_Inventory	Species_Inventory	47	Vector
	GAP FRACTION	Gap_Fraction	Gap_Fraction	Gap_Fraction	Gap_Fraction	Gap_Fraction	51	Vector
POROSITY	CANISTER	Canister_Porosity	Canister_Porosity	Canister_Porosity	Canister_Porosity	Canister_Porosity	55	Value
	BENTONITE	Bentonite_Porosity	Bentonite_Porosity	Bentonite_Porosity	Bentonite_Porosity	Bentonite_Porosity	56	Value
	EDZ	EDZ_Porosity	EDZ_Porosity	EDZ_Porosity	EDZ_Porosity	EDZ_Porosity	57	Value
	ROCK-MATRIX	Null	Rock_Matrix_Porosity	Rock_Matrix_Porosity	Null	Rock_Matrix_Porosity	58	Value
	ROCK	Null	Null	Null	Rock_Porosity	Null	59	Value
	TUNNEL	Null	Tunnel_Porosity	Null	Tunnel_Porosity	Null	60	Value
	ROCK-FRACTURE	Null	Null	Rock_Fracture_Poros	Null	Rock_Fracture_Porosity	61	Value
SOLUBILITY	SPECIES SOLUBILITY	Species_Solubility	Species_Solubility	Species_Solubility	Species_Solubility	Species_Solubility	62	Vector
SORPTION	CANISTER	Sorption_Canister	Sorption_Canister	Sorption_Canister	Sorption_Canister	Sorption_Canister	66	Vector
	BENTONITE	Sorption_Bentonite	Sorption_Bentonite	Sorption_Bentonite	Sorption_Bentonite	Sorption_Bentonite	70	Vector
	ROCK	Sorption_Rock	Sorption_Rock	Sorption_Rock	Sorption_Rock	Sorption_Rock	74	Vector
OUTPUT-TIMES	START-TIME	Start_Time	Start_Time	Start_Time	Start_Time	Start_Time	78	Value
	END-TIME	End_Time	End_Time	End_Time	End_Time	End_Time	79	Value
	# INTERVAL	Interval	Interval	Interval	Interval	Interval	80	Value
	NO-PER-DECADE	No_Per_Decade	No_Per_Decade	No_Per_Decade	No_Per_Decade	No_Per_Decade	81	Value
VELOCITY	DARCY-VELOCITY	Darcy_Velocity	Darcy_Velocity	Null	Darcy_Velocity	Darcy_Velocity	82	Value
	DIVISION-FRACTION	Null	Division_Fraction	Null	Division_Fraction	Division_Fraction	83	Value
	EDZ-VELOCITY	Null	Null	EDZ_Velocity	Null	Null	84	Value
	ROCK-VELOCITY	Null	Null	Rock_Velocity	Null	Null	85	Value
	FRACTURE-LOCATION	Null	Null	Fracture_Location	Null	Fracture_Location	86	Value

Note: The in() array index number is scaled for four radionuclides for the example presented in Table 1.

2.3.2. INPAGN_Launcher RN.DAT Input File

The INPAGN_Launcher DLL needs to read a file called "RN.DAT". The RN.DAT file is a simple text file that contains a list of the radionuclides included in the simulation. GoldSim™ cannot pass text STRINGS and therefore the radionuclide labels are needed for the DLL to write the INPAG-N input file. Figure 4 shows an example of a RN.DAT file that contains 40 radionuclide species and 6 stable isotope species. The RN.DAT file should list the radionuclides in the same order and format (excluding the half life) as the DECAYS input block required by INPAG-N. The RN.DAT input file must be defined in the order shown in Figure 4. The first value is the total number of radionuclides, the DLL uses this value to check that the in() array passed by GoldSim™ is the correct size. The second number lists the number of stable isotope species to be included in an INPAG-N input file. This feature is only available in the INER versions for INPAGNV2.exe and INPAGNH.exe. This value should be 0 for simulations without any stable isotopes defined.

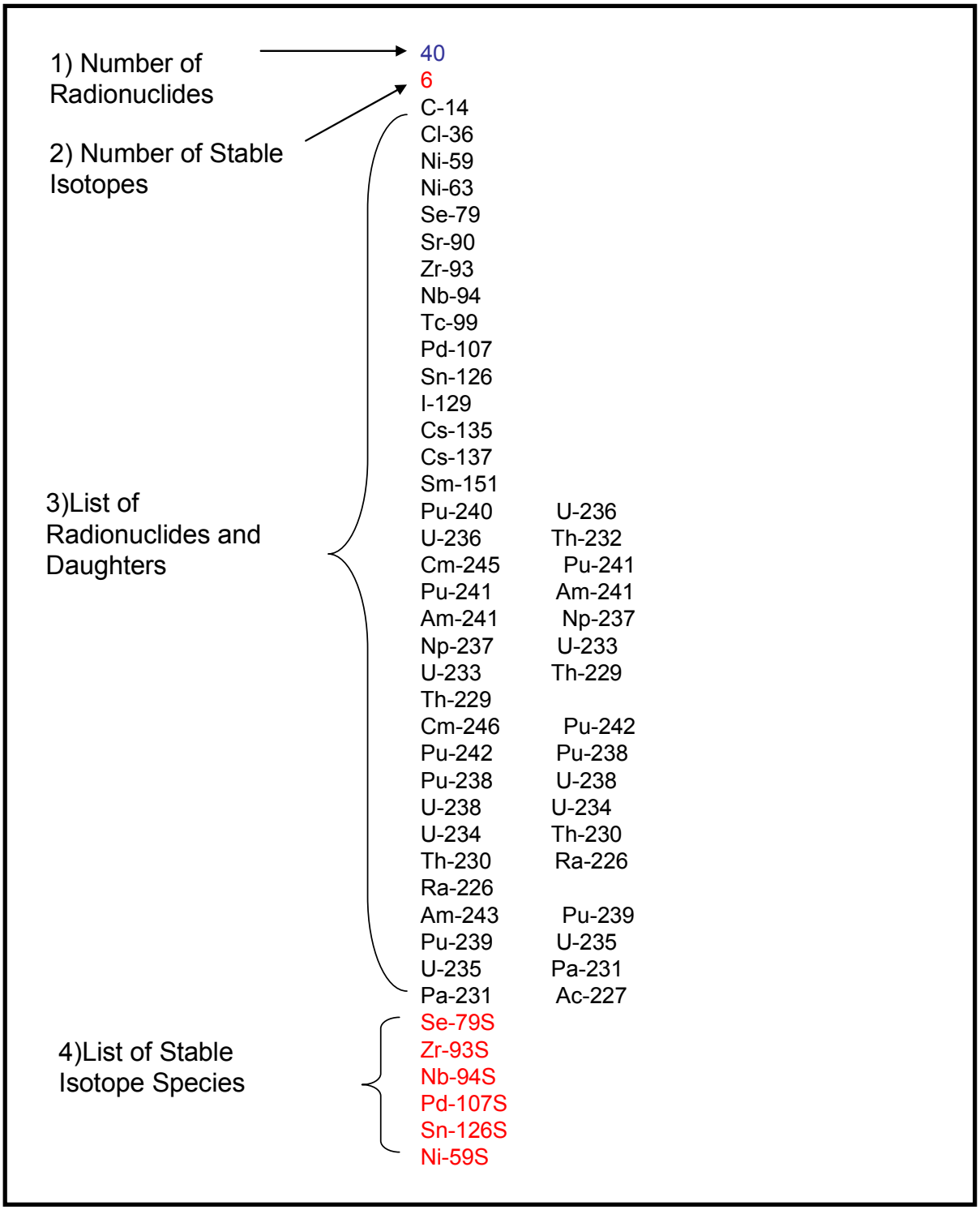

Figure 4: Example RN.DAT input file with 40 radionuclides and 6 stable isotope species.

2.3.3. Version of INPAG-N compatible with INPAGN_Launcher

The INPAGN_Launcher DLL is capable of executing with five versions of INPAG-N. The five versions are listed in Table 2.

Table 2: Versions of INPAG-N that can be run with INPAGN_Launcher

INPAG-N Version	Date	Description	Reference
INPAGN.EXE	04/19/2002	INPAG-N is developed for calculating radionuclide release rates from the EBS for the total system performance assessment of Taiwanese geological repository of spent nuclear fuel.	(Zhou, 2000)
INPAGNV2.EXE	08/20/2004	Version 2 of INPAG-N with vertical release into the disposal tunnel, or axial transport in the cylindrical system.	(Zhou, 2004)
INPAGNH.EXE	09/13/2005	INPAG-NH is a model and computer program that was developed to quantitatively assess the performance of horizontal emplacement repository for disposal of nuclear spent fuel in geological media.	(Zhou, 2005)
INPAGNV2I.EXE	12/11/2006	INPAGNV2.EXE Modified by INER	(INER, 2007a)
INPAGNHI.EXE	05/15/2007	INPAGNH.EXE Modified by INER	(INER, 2007b)

2.3.4. INPAGN_Launcher Input Subroutines

The INPAGN_Launcher DLL is executed using a series of calls to subroutines. The order of subroutine calls are listed below:

1. Containment
2. Timestep_Control
3. Decays
4. Density
5. Diffusion
6. Flux_Output
7. Geometry
8. Inventory
9. Porosity
10. Solubility
11. Sorption
12. Output_Times
13. Velocity

The subroutines read the input values contained in the in() array passed by GoldSim™. Each subroutine writes the data to an INPAG-N output file. Because each version of INPAG-N utilizes different elements in the input array, in(), and requires a unique input file, several subroutines were constructed that contain functionality unique to the INPAG-N version being used. The name and usage of each subroutine for each version of INPAG-N are listed in Table 3. The source code is included in the attached CD ROM, as listed in Appendix A.

Table 3: Input Subroutine Calls used in INPAGN_Launcher DLL

INPUT SUBROUTINE	INPAGN	INPAGNV2	INPAGNH	INPAGNV2 (INER)	INPAGNH (INER)
Containment	X	X	X	X	X
Timestep_Control	X	X	X	X	X
Decays	X	X	X	X	X
Density_NH_NV2M_NH	X	X	X		X
Density_NV2I				X	
Diffusion_NM	X				
Diffusion_NV2		X		X	
Diffusion_NH			X		X
Flux_Output	X	X	X	X	X
Geometry	X				
Geometry_NV2M		X			
Geometry_NV2I				X	
Geometry_NHM			X		
Geometry_NHI					X
Inventory	X	X	X	X	X
Porosity_NM	X				
Porosity_NV2M		X			
Porosity_NV2I				X	
Porosity_NH			X		X
Solubility	X	X	X	X	X
Sorption	X	X	X	X	X
Output_Times	X	X	X	X	X
Velocity_NM	X				
Velocity_NV2		X		X	
Velocity_NHM			X		
Velocity_NHI					X

2.4. Output Specification

The outputs from the INPAGN_Launcher DLL come from the output flux files created from INPAG-N. The DLL reads the values from the INPAG-N output file and loads them into the out() array which comprises a single array of values that is passed to GoldSim.exe at the end of each INPAG-N model run. Table 1 lists the output array out() passed by the DLL to the GoldSim.exe. The output values will be passed to GoldSim.exe once per realization during a probabilistic simulation.

2.4.1. INPAGN_Launcher Output Array: out()

An out() array needs to be defined in the GoldSim™ model file to accept the out() array passed to the INPAGN_Launcher DLL during execution of the code. Figure 5 shows an example DLL External Element (GoldSim User's Guide, GoldSim Technology Group LLC, 2006), which is used in the GoldSim™ model file to define a link between the GoldSim.exe and a DLL. The external DLL element requires three 2-D table elements to be defined. The INPAGN_Launcher DLL passes the following outputs: 1) total radionuclide flux from INPAG-N, 2) the tunnel or fracture radionuclide flux concentration for INER versions of INPAG-NV2 INPAG-NH respectively, and 3) the radionuclide rock flux concentrations for INER versions of INPAG-NV2 INPAG-NH.

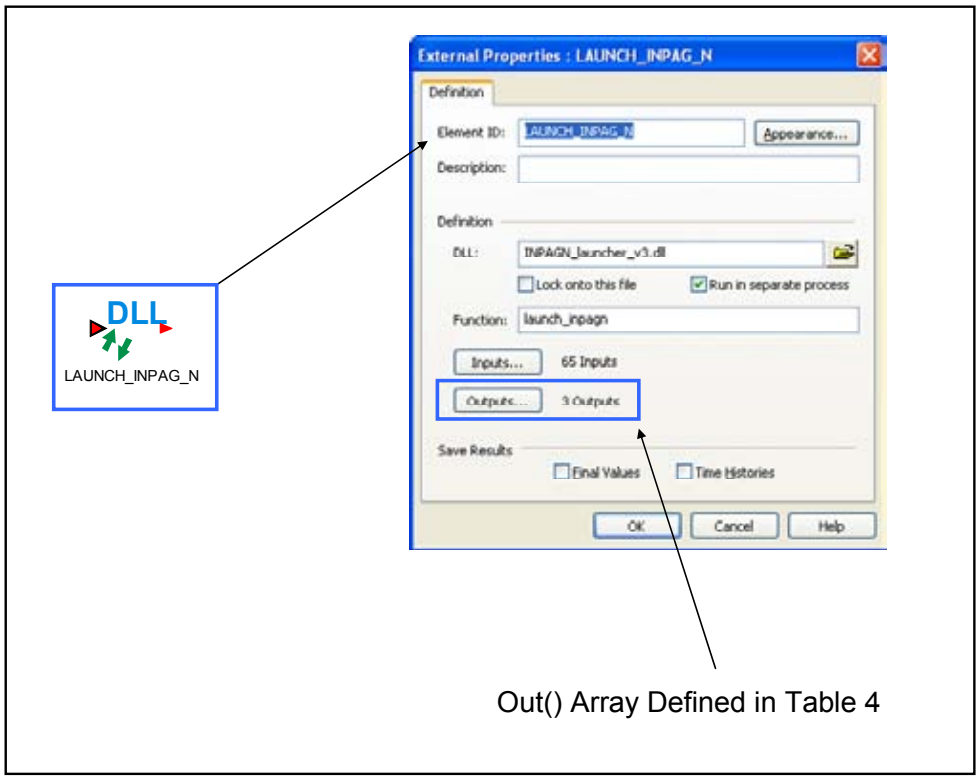

Figure 5: External DLL Element in GoldSim™ Linking out() Array to INPAGN_Launcher DLL

The output blocks and values are listed in Table 4 in the order that the GoldSim.exe and External DLL element expects to receive them. The GoldSim.exe calls the METHOD and reads the entire out() array at each call. The out() array is overwritten for each METHOD call. Refer to the GoldSim User's Manual Appendix C (p. 534, GoldSim Technology Group LLC, 2006) for further details on linking DLLs with the GoldSim.exe.

Table 4: INPAGN_Launcher DLL Output Array

out ( ) Array	Value	Description
		Method = 2
out(1)	1.0	DLL Version
		Method = 2
out(1)	57+(7*number of RN's)	Expected values in the in() array
out(2)	(number of RN's*200*3)+200	Expected out() array size
		Method = 2
out(1)	2.0	Start RN flux (total) output (value of 2 = 2-D table)
out(2)	T	number of timesteps
out(3)	R	number of columns
out(4) - out(3+T)	Timestep values	row labels
out(*) - out(*+R)	RN number	column labels (integer values between 1 and the total number of RNs)
out(*) - out(*+T*R)	RN Concentrations	flux values passed by row
out(1)	2.0	Start RN flux (tunnel or fracture) output (value of 2 = 2-D table)
out(2)	T	number of timesteps
out(3)	R	number of columns
out(4) - out(3+T)	Timestep values	row labels
out(*) - out(*+R)	RN number	column labels (integer values between 1 and the total number of RNs)
out(*) - out(*+T*R)	RN Concentrations	flux values passed by row
out(1)	2.0	Start RN flux (tunnel or fracture) output (value of 2 = 2-D table)
out(2)	T	number of timesteps
out(3)	R	number of columns
out(4) - out(3+T)	Timestep values	row labels
out(*) - out(*+R)	RN number	column labels (integer values between 1 and the total number of RNs)
out(*) - out(*+T*R)	RN Concentrations	flux values passed by row

2.4.2. Output Files Read by INPAGN_Launcher DLL

The INPAGN_Launcher DLL reads up to three files output by INPAG-N. Table 5 lists the INPAG-N output files by INPAG-N version. The INPAGN_Launcher DLL will read one or three INPAG-N output files, dependent upon the version selected.

Table 5: Versions of INPAG-N output files that can be read by INPAGN_Launcher

INPAG-N Version	Output File	Description
INPAGN.EXE	BASENF.FLX	INPAG-N output that contains the radionuclide concentration for selected output times.
INPAGNV2.EXE	*.FLX	Version 2 of INPAG-N output file that contains the radionuclide concentration for selected output times. * = input file name
INPAGNH.EXE	*.FLX	INPAG-NH output file that contains the radionuclide concentration for selected output times. * = input file
INPAGNV2I.EXE	*.FXN	Output file that contains the total radionuclide concentration for selected output times. * = input file
	*N.FXN	Output file that contains the tunnel radionuclide concentration for selected output times. * = input file
	*R.FXN	Output file that contains the rock radionuclide concentration for selected output times. * = input file
INPAGNHI.EXE	*.FXN	Output file that contains the total radionuclide concentration for selected output times. * = input file
	*F.FXN	Output file that contains the fracture radionuclide concentration for selected output times. * = input file
	*R.FXN	Output file that contains the rock radionuclide concentration for selected output times. * = input file

2.4.3. INPAGN_Launcher Output Subroutines

The INPAGN_Launcher DLL reads the INPAG-N output files using a series of calls to subroutines. The first call is to execute the specified version of the INPAG-N exe for the input file generated by the DLL (see Section 2.3 for Input Specifications). Next the output files are read and the data is passed back to the GoldSim.exe. The order of subroutine calls are listed below:

1. SYSTEM
2. Read INPAG-N Output Flux files
3. Load out() array
4. Pass out() array to GoldSim.exe
5. Clean-up and exit

The subroutines read the values contained in the output files created by INPAG-N. Each subroutine writes the data to an array. Because each version of INPAG-N utilizes different output files, several subroutines were constructed that contain functionality unique to the INPAG-N version being used. The subroutines are listed by name and an 'X' indicates when they are called for each version of INPAG-N in Table 6. The source code is included in the attached CD ROM, as listed in Appendix A.

Table 6: Output Subroutine Calls used in INPAGN_Launcher DLL

OUTPUT SUBROUTINE	INPAGN	INPAGNV2	INPAGNH	INPAGNV2 (INER)	INPAGNH (INER)
Execute INPAG-N					
SYSTEM	X	X	X	X	X
Read Flux Files					
Read_FLX_File	X	X	X		
Read_FLX_File_NV2I				X	X
Read_FLX_File_NV2IN				X	X
Read_FLX_File_NV2IR				X	X
Load data into out() array and pass back to GoldSim.exe					
BASENF_FLX_FILE	X	X	X	X	X
N_FLX_FILE				X	X
R_FLX_FILE				X	X
Clean-up and Exit DLL (METHOD = 99)					
DEALLOCATE	X	X	X	X	X
CLOSE (*)	X	X	X	X	X

2.5. Data Files

Sections 2.3 and 2.4 describe the format of the RN.dat input file and the required output data files as dictated by the versions of INPAG-N. The INPAGN_Launcher DLL writes one debug data file, scratch.txt. This is an ASCI text file that echo's the input and output arrays and prints out check points during the DLL execution that can be used to debug the input/output operations.

2.6. Defaults

There are no defaults. The INPAGN_Launcher DLL must receive the entire in() array or it will not run.

2.7. Errors

General error detection is not provided with the DLL. Table 7 lists common error messages generated by the DLL and the GoldSim.exe and some possible causes. The GoldSim™ program will be halted and the message displayed by GoldSim™ program to the user.

Table 7: Common DLL Error Messages

Error Message	Possible Cause(s)
The model cannot be run. LAUNCH_INPAG_N: \Near_Field_Calcuations\INPAG_N\LAUNCH_INPAG_N Could not load the DLL INPAGN_launcher_v3.dll	DLL is not in the Local Directory
	RN.DAT specified incorrectly
	in() array specified incorrectly
The model cannot be run. External element \Near_Field_Calcuations\INPAG_N\LAUNCH_INPAG_N must have at least 1 output	No output defined in GoldSim.exe model file for the DLL external element.
The model cannot be run. External element \Near_Field_Calcuations\INPAG_N\LAUNCH_INPAG_N input arguments do not match; External function is expecting 85 inputs, 86 arguments are defined in GoldSim™	in() array specified with incorrect number of elements.
The model cannot be run. Dimensions do not match for output of external element \Near_Field_Calcuations\INPAG_N\LAUNCH_INPAG_N	The DLL passes a 2-D table to the GoldSim.exe and the external element must have the correct type selected.
\Near_Field_Calcuations\INPAG_NH_INER\LAUNCH_INPAG_N Win32Exception caught: at address 0x	DLL Error
Visual Fortran run-time error: Forrtl:severe(2): file not found, unit 95,	Rn.dat file is not in local directory
Visual Fortran run-time error: Forrtl:severe(2): file not found, unit 9,	Rn.dat file is not in local directory
Visual Fortran run-time error: Forrtl:severe(2): file not found, unit 12,	*.FLX file not in local directory.
	selected version of INPAGN.EXE did not execute successfully
	selected version of INPAGN is not found in the local directory
	Input file did not execute; errors in input file data
Visual Fortran run-time error: Forrtl:severe(2): file not found, unit 13,	*.FXN file not in local directory.
	selected version of INPAGN.EXE did not execute successfully
	selected version of INPAGN is not found in the local directory
	Input file did not execute; errors in input file data
Visual Fortran run-time error: Forrtl:severe(2): file not found, unit 14,	*.FXN file not in local directory.
	selected version of INPAGN.EXE did not execute successfully
	selected version of INPAG is not found in the local directory
	Input file did not execute; errors in input file data

2.8. Hardware/Software Environments

The minimal hardware configuration to run INPAGN_Launcher DLL is a computer equipped with a Windows 32-bit or 64-bit operating system (e.g., Windows XP). INPAGN_Launcher DLL is developed as a dynamic link library (DLL) that must be called by another application. The software, GoldSim.exe (GoldSim Technology Group LLC, 2006), is required to operate the DLL. In addition, at least one version of INPAG-N (listed in Table 2) is required for the successful execution of the DLL.

3. INSTALLATION AND TEST CASES

3.1. Installation

Before installing the INPAGN_Launcher software, the user must ensure that a Windows operating system is installed on the target platform (see Section 2.8 Hardware/Software Specifications). The INPAGN_Launcher software is installed by placing a copy of the DLL on the hard-drive of a computer. The installation and test cases described herein assume that GoldSim™ (e.g. GoldSim™ Version 9.21 or later) is installed on the target machine. In addition, at least one version of INPAG-N (listed in Table 2) must be installed in the local directory to which the DLL and test cases are executed.

3.2. Test Cases

A test case is provided in Appendix A. Following the steps listed in Table 8 will provide evidence that the DLL is operating correctly once installed and that the functional requirements are satisfied. The test looks at the normal or expected operation of the DLL and has been developed to show a comparison of the resulting tables.

Table 8: Steps for Installation and Testing

Step	Description	Verification Results
1	Copy the INPAGN_Launcher_v3.dll, RN_Daughter.DAT and INPAGN_Launcher_DLLv3_Test_Case.gsm from the Appendix A directories on the CD media to a test directory on a workstation or desktop computer	Files appears in directory
2	Copy at least one executable version of INPAG-N (listed in Table 2) on the local computer that contains INPAGN_Launcher_v3.dll and INPAGN_Launcher_ DLLv3_Test_Case.gsm	File(s) appears in directory on the local computer.
3	Rename RN_Daughter.DAT to RN.DAT	File appears in directory.
4	Launch GoldSim™ v9.21 (or later)	Goldsim™ is launched
5	Open the file INPAGN_Launcher_ DLLv3_Test_Case1.gsm in GoldSim™.	File opens.

Table 8: Continued.

Step	Description	Verification Results
6	Select the version of INPAG-N by selecting 1-5 for the element "INPAG_N_Selector" in the INPAGN_Launcher_ DLLv3_Test_Case.gsm file.	N/A
7	Run the file by selecting /Model/Run Model from the pull-down menus.	GoldSim™ Run Controller appears. Dialog box appears with query, "Starting a new simulation will destroy your existing results. Are you sure you want to run again?"
8	Select "yes" to dialog box query.	Dialog box appears stating "Simulation Complete!"
9	Click "OK" in dialog box.	Dialog box disappears. Run Controller may also disappear depending upon GoldSim™ option settings.
10	Verify that the contents of the results tables match the output for test files included in Appendix A.3.	Results match.
11	Repeat Steps 6-10 for each version of INPAG-N	N/A

Please note that the values contained in the test case, INPAGN_Launcher_DLLv3_Test_Case.gsm, should not be used for any other purpose than as an example. The values do not necessarily represent a real repository system and are only default values used in an example model. In addition, the test case is set up for multiple realizations. Single realizations should be run to compare the results to the results from stand alone INPAG-N .exe runs.

4. SUMMARY

The objective of this document has been to provide a guide for using the INPAGN_Launcher DLL software package with GoldSim.exe. The test case included in this manual was based upon a hypothetical performance assessment model constructed by Sandia National Laboratories as part of a technology transfer program funded by the Taiwan ICP Phase II - Technology Transfer for Interim Dry Storage and Final Disposal of Spent Nuclear Fuel under a WFO Funds-in agreement contract between Sandia and Lockheed Martin. This work supports DOE/RW international leadership in geologic disposal of spent nuclear fuels and radioactive waste and working item IN-SNL-DD23, "Technology Transfer for Geologic Repository Science and Performance Assessment", under the AIT/TECRO Joint Standing Committee for Civil Nuclear Cooperation. Sandia National has over 30 years experience in assessing radioactive waste disposal and at the time of this publication is providing assistance internationally in a number of areas relevant to the safety assessment of radioactive waste disposal systems.

The SNL developed software enables INER to perform probabilistic simulations for safety analysis and performance assessment of geologic disposal of commercial spent nuclear fuel. The report guides a user through the steps necessary to use the software and presents a successful test case demonstrating the paradigm of coupling deterministic codes to a contemporary probabilistic software application. The effort was the result of a cooperative technology transfer project between Sandia National Laboratories and the Institute of Nuclear Energy Research (INER) in Taiwan for the preliminary assessment of a potential SNF waste repository in Taiwan. However, this software package can be easily modified to model site specific conditions for assessments of other potential radioactive waste repositories.

5. REFERENCES

Digital Equipment Corporation, 1998, Digital Visual FORTRAN, Professional Edition, Version 6.0.A, Software

GoldSim Technology Group LLC, 2006, Goldsim User's Guide Version 9.20, GoldSim Technology Group LLC, www.goldsim.com

INER, 2007a, INPAGNV2I.EXE, www.iner.tw.gov

INER, 2007b, INPAGNHI.EXE, www.iner.tw.gov

Zhou, W, 2000, INPAG-N User's Guide, Report MSCI-2020-2 Rev. 0, October 20, 2000, Monitor Scientific LLC, www.monitorsci.com.

Zhou, W, 2004, Development of INPAG-N Version 2: The Two-Dimensional Compartment Near-Field Model for Total System Performance Assessement of Taiwanese Geological Disposal Spent Nuclear Fuel, Report MSCI-2409-1 Rev. 0, August 17, 2004, Monitor Scientific LLC, www.monitorsci.com

Zhou, W., 2005, Performance Assessment of Horizontal Emplacement Repository, Report MSCI-2507-1 Rev. 0, July 21, 2006. Monitor Scientific LLC, www.monitorsci.com

APPENDIX A: SOURCE CODE AND TEST CASES

The source code and example GoldSim™ model file discussed in this document are stored on the Compact Disc (CD) distributed with this SAND report, and are also available for download at http://www.sandia.gov/icp/.

A.0 Test Case

INPAGN_Launcher_v3_DLL_GSv921_Test_Case.gsm

A.1 INPAGN_Launcher_v3.DLL and Source Code

Launch_inpagn_v003.f90
INPAGN_Launcher_v3.DLL

A.2 RN.DAT Examples

RN_No_Daughter.DAT
RN_Daughter.DAT

A.3 INPAG-N Test Case Results

BASENF.FLX
inpag_nv2.flx
inpag_nh.flx
inpagn_v2i.fxn
inpag_nv2iN.fxn
inpag_nv2iR.fxn
inpag_nhi.fxn
inpag_nhiR.fxn
inpag_nhiF.fxn

DISTRIBUTION

External Distribution

10 Institute of Nuclear Energy Research
No. 1000, Wenhua Rd. Chiaan Village
Lungtan Taoyuan. 325
TAIWAN. R.O.C.
Attn: Wen-Shou Chuang (7)
Shin-Jon Ju
Fu-Lin Chang
Ming-Chuan Kuo

1 Mr. Michael Hollmen
Director, Industrial Participation
Lockheed Martin MS2
P.O. Box 64525, MS U2E25
St. Paul, MN 55164-0525

1 Ms. Maureen Mahowald
Manager, International Cooperation and Offsets
Lockheed Martin MS2
P.O. Box 64525, MS U2E25
St. Paul, MN 55164-0525

1 Mr. Ralph Cady
Office of Nuclear Regulatory Research
Mail Stop T-9C34
U.S. Nuclear Regulatory Commission
Washington, DC 20555

2 Mr. Ian Miller
GoldSim Technology Group
22516 SE 64th Place, Suite 110,
Issaquah, WA 98027

Sandia National Laboratories Internal Distribution

5	MS0748	Donald A. Kalinich	6762
10	MS0776	Patrick D. Mattie	6784
1	MS0776	Jerry McNeish	6784
1	MS0776	Cedric Sallaberry	6784
5	MS1363	Hong-Nian Jow	6725
1	MS1399	M. Kathryn Knowles	6781
2	MS9018	Central Technical Files	8944
2	MS0899	Technical Library	4536