

OASIS Integration HailAUS7 2019

Risk Frontiers

Email: info@riskfrontiers.com

Website: www.riskfrontiers.com

Tel: +61 2 8459 9770
Level 8, 33 Chandos Street
St Leonards NSW 2065
Australia



TABLE OF CONTENTS

<i>Executive summary</i>	<i>4</i>
<i>General Architecture.....</i>	<i>5</i>
Oasis Loss Modelling Framework.....	5
End user input/output.....	5
ModEx Interface.....	5
Risk Frontiers model.....	5
<i>Requirements</i>	<i>6</i>
<i>Installation.....</i>	<i>7</i>
Deployment	7
Running the integrated model through Oasis UI.....	8
<i>Input exposure</i>	<i>11</i>
Open Exposure Data	11
Location file	12
OED fields and Risk Frontiers location features	13
Mapping between OED occupancy code and RF Line of Business.....	13
Mapping between OED construction code and RF construction type	13
Mapping between OED and RF physical asset properties	13
<i>Implementation Details.....</i>	<i>14</i>
Technology	14
Portable engine via .Net Core 2.2	14
SQLite as backend database	14
Detailed architecture.....	14
Output generation: gulcalc output stream	16
Analytical Mean and Standard deviation	16
Sum and product of independent random variables.....	16
Uniform distribution	16
Exponential of the normal distribution	16
<i>Validation.....</i>	<i>18</i>
Methodology	18
Validation Input Exposure.....	18

Validation Accuracy	18
Validation Results	19
Exposure at latitude/longitude level.....	19
Exposure at Postcode level.....	20
<i>Appendix</i>	22
Correspondence between OED location and RF portfolio input format.....	22

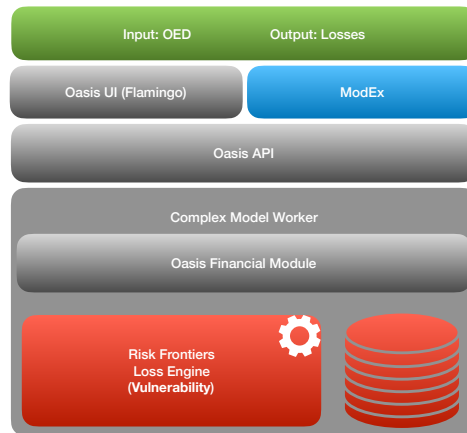
EXECUTIVE SUMMARY

This document provides detailed documentation of the integration of Risk Frontiers' Australian hail model, codename HailAUS7, into the Oasis Loss Modelling Framework. This integration is achieved using the newly released Complex Model MDK which support the integration of a custom loss engine into the Oasis framework. Here is a summary of the capabilities and features of the integrated HailAUS7 model.

- HailAUS7 is a national severe hailstorm model for Australia. It is fully integrated into OasisLMF using the Complex Model MDK.
- The event set contains simulated hail events for 50,000 years featuring around 90M individual hail storm footprints.
- It features both static and dynamic motor vulnerability models which can simulate seasonal car movements and covered car spaces in major cities in Australia.
- The integrated model supports Open Exposure Data (OED) input exposures with any combination of GNAF ID (address), Latitude/Longitude, Postcode, ICA Zone and Cresta geolocation.
- All basic features and execution modes of Risk Frontiers Multi-Peril Workbench are supported. These include the Dynamic Motor execution mode and the post amplification of losses using Demand Surge. In particular, it supports both the conventional single sample execution of Risk Frontiers models as well as the multi-sample requirement by the Oasis framework.
- The integration can only be used with Individual Risk Mode turned on, otherwise, Oasis' financial module will apply the wrong deductible to obtain the gross losses.
- The integration generates ground up losses that are validated to be practically indistinguishable from the Multi-Peril Workbench.
- Since this is an OasisLMF integration, we fully rely on the Oasis Financial Module and the validation of the integration is restricted to ground up losses.

GENERAL ARCHITECTURE

A general overview of the structure of the integration can be found in the following figure.



This is an implementation of the new Complex Model MDK from OasisLMF.

Oasis Loss Modelling Framework

The grey components are produced and maintained by the Oasis team. The Oasis UI is a web-based graphical interface providing visual access to the Oasis eco-system. The Oasis API is the main bridge between the frontend and backend components of the framework. The Complex Model worker is an instance of the oasis model worker responsible for performing loss sampling (via Monte Carlo simulation) and financial calculation. These are integrated into the ktools suite.

End user input/output

The green input and output are produced and consumed by the model end user. Only OED formatted inputs are currently supported.

ModEx Interface

The blue component is provided by ModEx as an alternative visual interface to the system. Note that a ModEx deployed installation is fully managed by the ModEx team.

Risk Frontiers model

Finally, the red loss engine and databases are provided and maintained by Risk Frontiers. They correspond to Risk Frontiers' dynamic vulnerability and stochastic event catalogue with associated metadata.

REQUIREMENTS

Note that predicting an accurate resource requirement and run-time for the deployment of the complex model integration is a hard and most likely subjective science. It depends on a number of parameters including, but not limited to, the deployment setup (dedicated or share resource), the size of the input portfolio, the number of covers per risk, the number of samples, the geographical resolution of the risk (lat/lon, postcode, cresta, etc.), the number of events affecting the input locations, etc.

That being said, here is a very general rule of thumbs resulting from our experiments: **a single-batch execution of the integrated complex model consumes approximately 10-16GB of memory**. Thus, it is advised to reserve that amount of memory per batch. For instance, on a 40 cores machine with 256 GB of memory, you should set the number of batches to be between 16 and 25 processes. You should not worry spawning much less than 40 processes. The reason is that, while Oasis splits the event set into up to 25 batches in this case, Risk Frontiers engine also performs parallelization with up to 20 threads working on the risks in parallel. This will use some parts of the remaining cores while still leaving enough computing resources for other tasks.

Num. risks × cover	Num. samples	Num. of batches	Temp. storage	Run time (expected)
1	10			
100	10			
1000	10			
5000	1			
10000	1			
50000	1			
100000	1			
500000	1			
1000000	1			

INSTALLATION

Deployment

To install the complex model integration of HailAUS 7, follow the following steps. Note that the integration has been tested on Ubuntu 18.04. The commands are exactly as follows for Debian system and they should work without any issue on any other flavor of Linux with appropriate modifications.

1. Install git, docker and docker-compose

```
sudo apt update && sudo apt install tree git docker docker-compose
```

2. Add user to the docker group and switch user to obtain a shell where user is actively a member of that group

```
sudo usermod -aG docker user
su - $USER
```

3. Clone the repository

```
git clone https://github.com/risk-frontiers/OasisComplexModel.git --branch=<release tag>
```

4. **Optional:** change the user/password combination used to access the Oasis UI by changing OASIS_ADMIN_USER and OASIS_ADMIN_PASS in docker-compose.yml if required.
5. Extract the model data and copy your license.txt into the model_data root folder. You can use [WinSCP](#) to copy files from windows to linux.
6. Copy model_data inside OasisComplexModel. The folder structure should be as follows

```
user@ubuntu:/home/user$ tree -L 1 OasisComplexModel/
OasisComplexModel/
├── api_evaluation_notebook
├── complex_model <----- this contains Risk Frontiers' executables
├── conf.ini
├── docker-compose.yml
├── Dockerfile.custom_model_worker
├── install.sh
├── model_data <----- model_data contains license.txt and Risk Frontiers' data
├── model_resource.json
├── OasisUI
├── README.md
├── requirements.txt
├── rf_install.sh <----- Risk Frontiers complex model installation script
├── setup.py
└── tests
```

7. Run the deployment script

```
cd OasisComplexModel
./rf_install.sh
```

8. Open a web browser from a computer attached to the network and navigate to `http://<server>:8080/app/BFE_RShiny` to access the Oasis UI interface (<server> should be replaced with the IP address or hostname of the server where the complex integration was deployed).

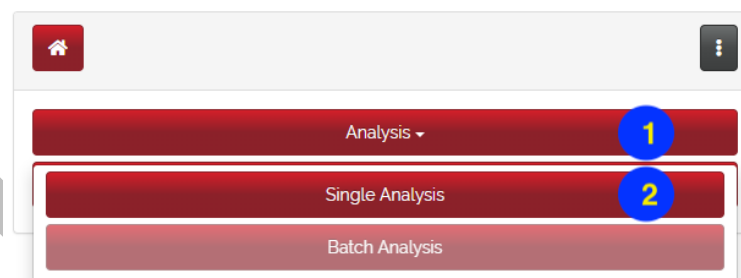
Running the integrated model through Oasis UI

To use the integrated model through the Oasis UI, please follow the following steps. To start an analysis, you need at least an input location file formatted according to the [Open Exposure Data specification](#). Our integration only accepts OED input files and a sample location file can be found in our [repository](#).

1. Navigate to `http://<server>:8080/app/BFE_RShiny` and enter `admin:password` as the default admin and password (or use the combination set during deployment).



2. Create an analysis by selecting **Analysis** → **Single Analysis** on the left-hand side panel.



3. We now need to create a portfolio.

Choose Portfolio | Choose Analysis | Configure Output & Run

Portfolios table

Show 10 entries

No portfolio available

Showing 1 to 1 of 1 entries

Previous 1 Next

Create Portfolio **1** Amend Portfolio Delete Portfolio Upload Source Files Show Source Files

4 Proceed to Choose Analysis

Create portfolio

Portfolio metadata

Portfolio Name

test **2**

3 Submit

4. Upload the location, account and reinsurance files.

Choose Portfolio | Choose Analysis | Configure Output & Run

Portfolios table

Show 5 entries

id	name	created	modified	status
1	test	8-04-19 02:45:24	8-04-19 02:45:35	

Showing 1 to 1 of 1 entries

Previous 1 Next

Create Portfolio Amend Portfolio Delete Portfolio Upload Source Files Show Source Files

3 Proceed to Choose Analysis

Link input files to portfolio id 1 "test"

Location file

Browse RandomGNAF_1.csv **1**

Link File **2**

Upload complete

Account file

Browse No file selected

Link File

RI info file

Browse No file selected

Link File

RI scope file

Browse No file selected

Link File

Clear

5. Create an analysis. This will generate input files by calling Risk Frontiers keys server. It might take a while especially if the locations are given at lat/lon level and without postcodes.

Choose Portfolio | **Choose Analysis** | Configure Output & Run

Portfolio ID 1

Analyses associated with portfolio "test", id 1

Show 10 entries

no analysis available

Showing 1 to 1 of 1 entries

Previous 1 Next

Start Input Generation Cancel Input Generation Show Generated Inputs Show Log Show Details

1 Create Analysis

4 Proceed to Configure Output & Run

Pick a model and choose an analysis name

Show 5 entries

id	supplier_id	model_id	version_id	created	modified	resource_file
1	RiskFrontiers	HAI4US	7.0.0.0	8-04-19 02:43:27	8-04-19 02:43:27	http://127.0.0.1:8000/v1/models/1/resource_file/

Showing 1 to 1 of 1 entries

Show Model Details

Analysis Name

analysis1 **2**

3 Submit

6. Configure the model run and then click **Execute**. Depending on the size of the portfolio and the model, it may take a while for the calculation to be performed.

Choose Portfolio Choose Analysis **Configure Output & Run**

Portfolio ID: 1

Analyses associated with portfolio id 1 Refresh

Processes' Status
☒ All ☐ In Progress
 Show 5 of 1 entries

id	name	portfolio	model	modified	created	status
1	analysis	1	1	8-04-19 02:47:31	8-04-19 02:47:32	✓

Showing 1 to 1 of 1 entries

Cancel Analysis Run Show Log Output Configuration Proceed to Dashboard

Define output configuration for analysis id 1 "analysis"

Configuration details

Select Custom Configuration:

Model parameters

Number of Samples:

1

Output configuration

☒ Ground Up Loss

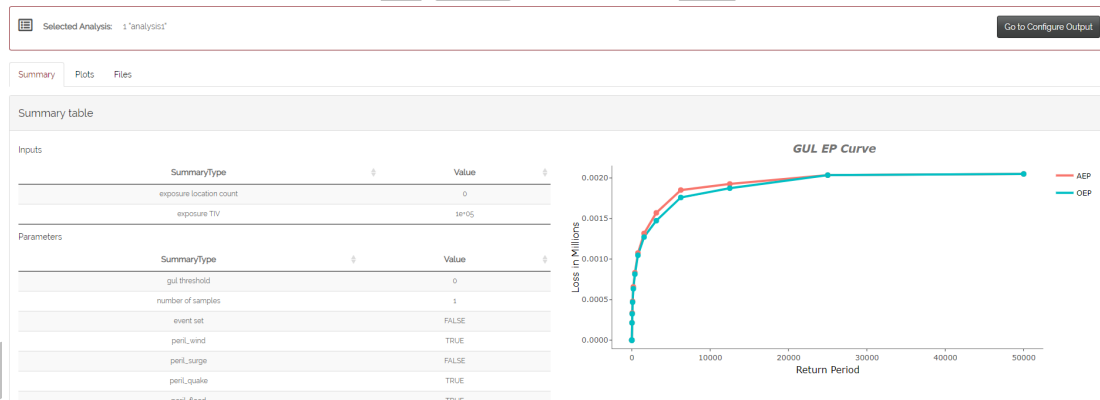
☐ Insured Loss

☐ Net RI Loss

Advanced

Execute Run

7. Once the calculation has been performed, the button **Proceed to Dashboard** will be enable. Click on it to download, analyse or visualize the output.



INPUT EXPOSURE

Open Exposure Data

Open Exposure Data, or OED, is an open-source exposure-data format that is specific to the Oasis framework. It provides a unified format for expressing exposure thus removing possible ambiguity from supporting multiple input formats. Moreover, OED was designed to align with the Oasis financial module.

OED has four main components:

- **Location:** This table contains details relating to each location. This includes details relating to the value and type of asset (including primary and secondary modifiers), geographical information, the perils covered and the financial structures within the insurance contract relating to the location. Note that **one location can be represented by multiple rows**.
- **Account:** The account table contains details of the policies and accounts that exist within the import portfolios. Most of the fields in this table relate to financial structures, including special conditions. Note that **one policy may have multiple rows** in the account file.
- **Reinsurance info:** The reinsurance info table contains details of the reinsurance contracts that relate to the underlying portfolios, accounts and locations. There must be exactly one entry per reinsurance contract in this table.
- **Reinsurance scope:** The reinsurance scope file contains details of three different but related pieces of information: (a) the scope of the reinsurance contract: i.e. which portfolios, accounts, locations are covered by a particular reinsurance contract, (b) the risk level of the reinsurance contract: i.e. for reinsurance contracts with financial structures relating to a 'risk', the definition of what risk means, (c) the CededPercent for a surplus treaty: which can vary for each risk covered by the treaty.

Since the main role of Risk Frontiers is to provide the fundamental model to generate ground up losses, we will only discuss about the Location file and the interpretation of the data contained within that input when executed by Risk Frontiers' models. More information about the full OED specification can be found [here](#). Please ensure that your input files are formatted according to that specification, in particular, regarding the presence and values of required and conditionally required fields.

Location file

The location file must be provided by user when running an analysis in the Oasis framework. It records the geographical, financial and physical features of the insured location. These features are recoded as columns within the OED format, which is stored as a standard Comma Separated Value (CSV) file.

The following fields are the minimal required and conditionally required columns as per the OED specification. In fact, all that is required to generate ground up losses are the respective total insured values.

Field name	Description	Datatype
LocNumber	Location number	nvarchar(20)
AccNumber*	Account number	nvarchar(40)
PortNumber*	Portfolio number	varchar(20)
BuildingTIV	Building total insured value	float
OtherTIV	Other total insured value (including motor)	float
ContentsTIV	Contents total insured value	float
BITIV	Business interruption total insured value	float
CountryCode	Country code (based on ISO3166 alpha-2 codes)	char(2)
LocPerilsCovered	Location perils covered	nvarchar(250)
LocCurrency	Location currency	char(3)

* These two fields are not specified as required in the OED specification but should be part of any location file as per the OED documentation pdf.

Note that location must also have at least one specified geo-location, which can be an address id (GNAF), latitude/longitude pair, postcode, ICA zone or cresta. The following two tables specify these fields.

Field name	Description	Datatype
Latitude*	Latitude in degrees (-90.0 to +90.0)	float
Longitude*	Longitude in degrees (-180.0 to +180.0)	float
PostalCode	A 4 digits number representing an Australian postcode	int
AreaCode	At most 3 characters Australian state code	char(3)

* Please make sure that these are either empty or missing

GeogScheme	Interpretation	Example GeogName value	GeogName Datatype
XGNAF	Australiana GNAF ID	GATAS702557490	char(14)
CRL, CRH	ICA Zone code from 1 to 49	1	int
CRO	Cresta code from 1 to 49	5	int

OED fields and Risk Frontiers location features

Care should be taken regarding the following convention. The detailed correspondence between OED location fields and Risk Frontiers portfolio columns can be found in the appendix.

Field name	Description	Data Type
LocUserDef1	Assumed to hold Proportion of building built before 1981	float
YearBuilt	Assumed to hold the year when the location was built	int

Mapping between OED occupancy code and RF Line of Business

The following table records the correspondence between OED occupancy codes and Risk Frontiers Line of Businesses.

OED Occupancy Code Range	Broad Category of Occupancy	Risk Frontiers Line of Business
1000	Unknown	Residential
1050 – 1099	Residential	Residential
1100 – 1149	Commercial	Commercial
1150 – 1199	Industrial	Industrial
1200 – 3999	Unsupported	Unsupported

Mapping between OED construction code and RF construction type

This is currently not required for HailAUS 7.

Mapping between OED and RF physical asset properties

This is currently not required for HailAUS 7.

IMPLEMENTATION DETAILS

Technology

Portable engine via .Net Core 2.2

The main technology that enables the embedding of Risk Frontiers loss engine into the oasis platform is **.Net Core 2.2**. This permits us to maintain the exact same code base for Risk Frontiers Multi-Peril Workbench and its Oasis integration counterpart. It also allows the loss engine to run on multiple platform as the .Net Core technology is platform agnostic.

SQLite as backend database

One main technical difference between the Multi-Peril Workbench and the Oasis complex model is the backend database used by the loss engine. In this integration, we are using **SQLite** databases to store hazard, exposure, loss and auxiliary data.

The main implication is that there are limitation to using embedded relational databases such as SQLite compared to a full blown SQL server. In particular, this is particularly acute in terms of performance where the integrated Oasis complex model is much slower than the Multi-Peril Workbench.

There are also some subtle differences in accuracy as SQLite does not support 4-bytes integer. Therefore, the loss output from the complex model and the Multi-Peril Workbench are not exactly the same in terms of decimal number representation. For more information regarding this challenge, please read the Validation section.

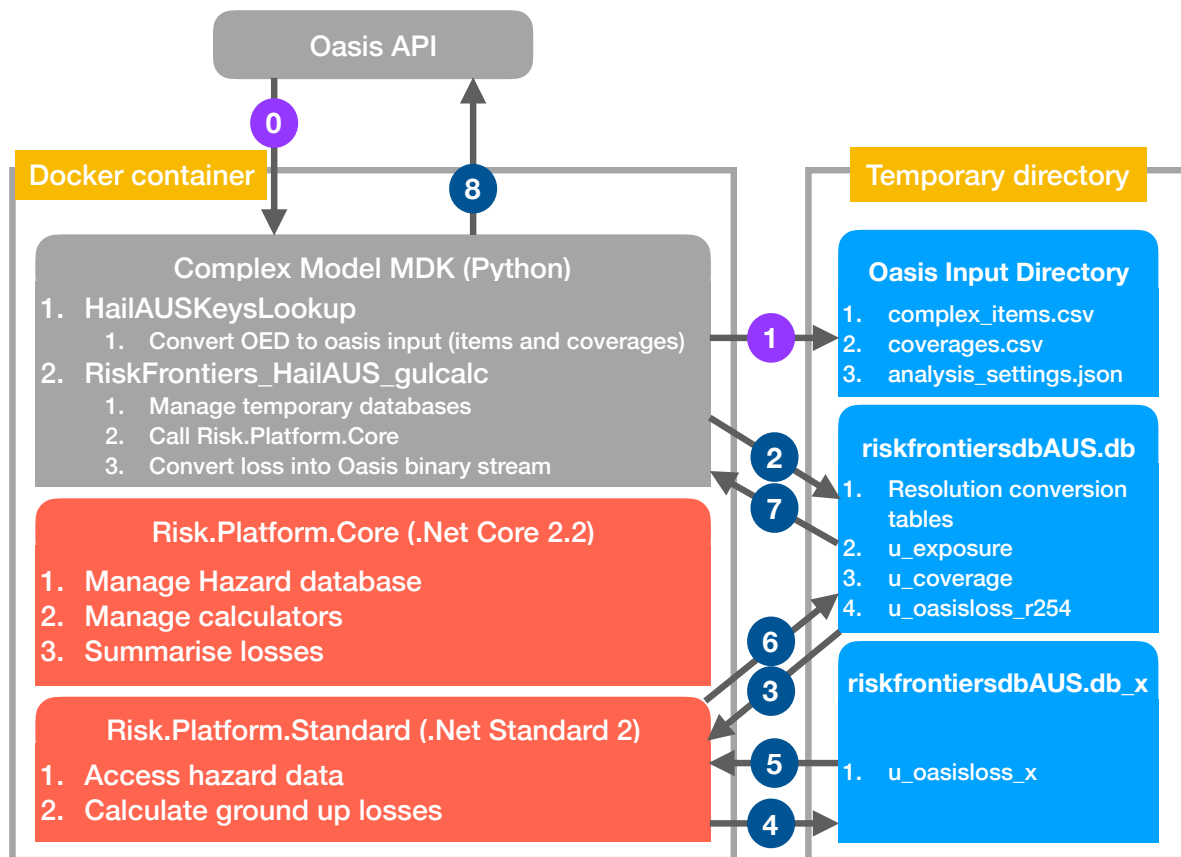
Detailed architecture

The following diagram shows the work flow as well as data flow for the Oasis complex model integration. Step 0 and 1 is about input preparation while Step 2 to 8 generate losses and formats them according to the Oasis stream specification. The descriptions of these steps are as follows.

- **Step 0:** The Oasis API calls the HailAUSKeysLookup to generate the necessary oasis input files. It accepts OED directly from the oasis framework. This lookup service provides basic features such as postcode

lookup for lat/lon based input using a QuadTree search. Address based input postcode lookup are dealt with at a later stage.

- **Step 1:** the input are store as csv files in a temporary directory (copies are also stored persistently somewhere else for cases such as re-running an analysis).



- **Step 2:** At this stage, RiskFrontiers_HailAUS_gulcalc has been called and the necessary input files have been generated. This module transforms these input files into uni_exposure rows and stores them into the riskfrontiersdbAUS.db (which is copied from a template in model_data). It then calls the Risk.Platform.Core executable which will create the required calculator and runs it.
- **Step 3:** The calculator instance is now running. It reads each row exposure stored in u_exposure and u_coverage and sample losses. Processing of the rows are done in batches and in parallel.
- **Step 4:** The losses are stored in independent temporary databases to maximise parallelism.
- **Step 5-6:** Once every row has been processed, the partial loss databases are combined and stored into u_oasisloss_r254. The losses are order by event_id, region_id and sample_id. Two special samples have index -1 and -2 and they correspond to Oasis' analytical mean and standard deviation.
- **Step 7-8:** The combined table is not transformed into oasis gulcalc output binary stream. This stream is sent through a pipe and consumed by the next operation in the ktools suite. The construction of this stream is discussed in detail in the following section.

Output generation: gulcalc output stream

A gulcalc stream is one of the many streams in the Oasis framework. It represents an optimized stream of losses than can be consumed directly by other processes in the ktools suite. The following datagram shows how gulcalc streams are built.

1	2	3	4	5	6	7	8
0x01	stream_type:0x02			num_samples:int32			
event_id:int32				item_id:int32			
sidx:int32				loss:float32			
sidx:int32				loss:float32			
0x0000000000000000							
event_id:int32				item_id:int32			
sidx:int32				loss:float32			

Analytical Mean and Standard deviation

We use the following formulae to compute analytical mean and variances

Sum and product of independent random variables

All random variables used in HailAUS 7 are independent and the analytical mean and variances of their linear combinations and products are computed using the following identities

$$E[aX + bY] = aE[X] + bE[Y]$$

$$E[XY] = E[X]E[Y]$$

$$Var(X) = E[X^2] - E[X]^2$$

$$Var(aX + bY) = a^2Var(X) + b^2Var(Y)$$

$$Var(XY) = Var(X)Var(Y) + E[X]^2Var(Y) + Var(X)E[Y]^2$$

Uniform distribution

The uniform distribution $X \sim unif(a, b)$ is used as a source of uncertainty in the HailAUS 7 implementation. Its analytical mean and variance are

$$E[X] = \frac{(a+b)}{2} \quad \text{and} \quad Var[X] = \frac{(E[X]-a)^2}{12}$$

Exponential of the normal distribution

The normal distribution $X \sim N(0, \sigma)$ is used as a source of uncertainty in the HailAUS 7 implementation. It is furthermore transformed by the exponential function so that the analytical mean and variance are

$$E[e^X] = e^{\sigma^2} \quad \text{and} \quad \text{Var}[e^X] = E[X]^2(E[X]^2 - 1)$$

Note that the normal distribution in the implementation was truncated. Computing an analytical formula for the exponential of that distribution is extremely difficult so we have approximated it with the un-truncated version. This implies that the analytical variance is more generous.

VALIDATION

Methodology

The validation of the OASISLMF integration of Risk Frontiers HailAUS 7 model is achieved through an exact ground-up loss comparison between the OASISLMF implementation with the Multi-Peril Workbench output. Thus, this validation phase takes results from the Multi-Peril Workbench as ground truth whose validation can be in turn found in the official manual. The losses were generated from within the worker docker only as, at the time of this validation, few issues prevented us from executing large analysis from the Oasis UI.

The comparison uses both absolute and relative differences. Absolute difference looks at the difference between two numbers while relative difference looks at the ratio between the absolute difference and the expected output. Relative difference is particularly useful when comparing results that are very large or very small as the floating-point representation of real numbers introduces inaccuracies.

Important note: we are only validating the ground-up loss produced by the OASISLMF integration of HailAUS 7 and do not perform any formal check on the correctness of losses after the application of the OASISLMF financial module. We assume that that financial module is correct, and its validity is ensured by the OASISLMF development team.

Validation Input Exposure

The input exposure used in the validation is built from market portfolios, which are split into two major categories:

1. Building, Contents and Business Interruption coverages: the market portfolios are built from the Nexis 2016 dataset.
2. Motor coverage: the market portfolio is built from the Red Book data set.

These market portfolios were converted into RF's exposure format and the Open Exposure Data (OED) format ensuring that we execute both implementations on the exact same inputs. More details regarding the input exposures are given in tables below.

Validation Accuracy

We have performed an end-to-end validation of the OasisLMF worker implementation. That is, losses were generated on a Linux platform where the integration was run. The ground up losses were in the form of binary files, which were converted into csv using the `gultocsv` tool and

then imported into a SQL server and compared with the corresponding losses generated by the Multi-Peril Workbench. Run from the above input exposure shows that the OASISLMF implementation produces losses that are practically the same as the Multi-Peril Workbench. Some very small fluctuations were observed (absolute differences well below 10^{-6}) which are expected due to numerical conversion and slight difference in the implementation.

Note: The seed for the random sampling must match for both the OasisLMF implementation and the Multi-Peril Workbench execution to obtain accuracy at the loss per location and event level. To achieve this, the input portfolio in the Multi-Peril Workbench must contain a `location_id` field that is set to the line number for each row. (This convention is used in the OasisLMF implementation.) This field forms part of the seed for any random sampling.

Validation Results

Exposure at latitude/longitude level

For this test we have selected random 200 GNAF points around Australia and built a small test portfolio for each of the combination shown in the tables below. These results also apply to larger portfolios.

The following table shows the maximum **absolute differences** between the output losses, **per event and location (EL)**, from the Multi-Peril Workbench and the OASISLMF integration. As expected, the results are extremely close and the only observed fluctuations are strictly below E-06 which is due to various numerical conversion and difference in data storage. Motor 0 to 4 corresponds to a motor exposure with age group 0 to 4 (0 being unknown and 4 for most recent cars).

EL Loss	Building	Contents	BI	Motor 0	Motor 1	Motor 2	Motor 3	Motor 4
Residential	5.00E-07	5.00E-07		5.00E-07	5.00E-07	5.00E-07	5.00E-07	5.00E-07
Commercial	5.00E-07	5.00E-07	5.00E-07					
Industrial	5.00E-07	5.00E-07	5.00E-07					

Since the losses are practically similar at event and location level, the **Exceedance Probability** curves and **Annual Losses** are also very close through the floating-point errors are

compounded. The following tables shows the maximum **relative difference** between the Average Annual Losses (AAL) and losses at Return Interval of 200 produced by the Multi-Peril Workbench and the OASISLMF integration.

The following table shows that the AAL are matching up to a variation of less than 0.0008%. That slightly visible fluctuation is due to having a small portfolio size. Practically, this variation is negligible.

AAL	Building	Contents	BI	Motor 0	Motor 1	Motor 2	Motor 3	Motor 4
Residential	0.0004%	0.0007%		0.0001%	0.0002%	0.0001%	0.0001%	0.0001%
Commercial	0.0005%	0.0001%	0.0001%					
Industrial	0.0008%	0.0002%	0.0001%					

The following table shows that at ARI 200, the outputs are practically the same.

ARI 200	Building	Contents	BI	Motor 0	Motor 1	Motor 2	Motor 3	Motor 4
Residential	0.0000%	0.0000%		0.0000%	0.0000%	0.0000%	0.0000%	0.0000%
Commercial	0.0000%	0.0000%	0.0000%					
Industrial	0.0000%	0.0000%	0.0000%					

Exposure at Postcode level

For this test, we have selected a market portfolio built from the Nexis and Red Book data set. The exposure is uniformly at Postcode level. The tables only show maximum **relative differences** as we are approaching large losses and the representation inaccuracies become apparent.

EL Loss	Building	Contents	BI	Motor 0	Motor 1	Motor 2	Motor 3	Motor 4
Residential	0.1614%	0.0783%		0.0013%	0.0004%	0.0005%	0.0012%	0.0020%
Commercial	0.7575%	0.4445%	4.0963%					
Industrial	0.0034%	0.0022%	0.0133%					

In the yellow cell, the relative differences seem to be high. Note however that these are the maximum differences and the average difference is much smaller than E-06 (with a standard

deviation of less than E-04). The maximum 4% difference is attached to a loss in the order of E-05.

The following table shows that the AAL are practically indistinguishable. In contrast, to the lat/lon validation, the AAL produced by the OASISLMF integration is much closer to the output of the Multi-Peril Workbench as we have considerably more risk points in the market portfolio.

AAL	Building	Contents	BI	Motor 0	Motor 1	Motor 2	Motor 3	Motor 4
Residential	0.0000%	0.0000%		0.0000%	0.0000%	0.0000%	0.0000%	0.0000%
Commercial	0.0000%	0.0000%	0.0000%					
Industrial	0.0000%	0.0000%	0.0000%					

The following table shows that at ARI 200, the outputs are practically the same.

ARI 200	Building	Contents	BI	Motor 0	Motor 1	Motor 2	Motor 3	Motor 4
Residential	0.0000%	0.0000%		0.0000%	0.0000%	0.0000%	0.0000%	0.0000%
Commercial	0.0000%	0.0000%	0.0000%					
Industrial	0.0000%	0.0000%	0.0000%					

APPENDIX

Correspondence between OED location and RF portfolio input format

OED Field name	Req	RF Field Name/usage	Description	Type [default]
LocUserDef1	O	address_id	GNAF id	String []
Latitude	O	latitude	Location latitude	Decimal [0]
Longitude	O	longitude	Location longitude	Decimal [0]
PostalCode	O	postcode	Location postcode	Integer [0]
LowResCresta	O	cresta	Cresta code	Integer [0]
HighResCresta	O	lca_zone	ICA Zone	Integer [0]
AreaCode	O	state	State	Char(3) []
CountryCode	R	country_code	Country code	Char(2) [AU]
LocCurrency	R	currency	Location currency	Char(3) [AUD]
LocPerilsCovered	R			Integer [0]
OccupancyCode	R	line_of_business	OED occupancy code	Integer [1050]
ConstructionCode	O	construction_type_id	OED construction code	Integer [5000]
LocationNumber	R	loc_id	Location ID	Integer
BuildingTIV	R	building_sum_insured	Building sum insured	Decimal [0]
OtherTIV	R	motor_sum_insured	Motor sum insured	Decimal [0]
ContentsTIV	R	contents_sum_insured	Contents sum insured	Decimal [0]
BITIV	R	business_interruption_sum_insured	BI sum insured	Decimal [0]
FirstFloorHeight	O	floor_height	Floor height	Decimal [0]
FirstFloorHeightUnit	O		Floor height unit	String [m]
NumberOfStories	O	Combined to infer building_type_id,	Number of stories	Integer [0]
BaseFloodElevation	O	high_rise	Building elevation	Decimal [0]
BaseFloodElevationUnit	O		Building elevation unit	String [m]
YearBuilt	O	year_num	Year built	Integer [0]
LocUserDef2	O	Pre_post_ratio	Proportion of building built before 1981	Decimal [-1]
LocDed1Building	CR	building_deductible	Building deductible	Decimal [0]
LocDedType1Building	O	deductible_type*	Building ded. type	Integer [0]
LocDed2Other	CR	motor_deductible	Motor deductible	Decimal [0]
LocDedType2Other	O	deductible_type*	Motor deductible type	Integer [0]
LocDed3Contents	CR	contents_deductible	Contents deductible	Decimal [0]
LocDedType3Contents	O	deductible_type*	Contents ded. type	Integer [0]
LocDed4BI	CR	business_interruption_deductible	BI deductible	Decimal [0]
LocDedType4BI	O	deductible_type*	BI deductible type	Integer [0]
LocLimit1Building	CR	building_limit	Building limit	Decimal [0]
LocLimitType1Building	O	limit_type*	Building limit type	Integer [0]
LocLimit2Other	CR	motor_limit	Motor limit	Decimal [0]
LocLimitType2Other	O	limit_type*	Motor limit type	Integer [0]
LocLimit3Contents	CR	contents_limit	Contents limit	Decimal [0]
LocLimitType3Contents	O	limit_type*	Contents limit type	Integer [0]
LocLimit4BI	CR	business_interruption_limit	BI limit	Decimal [0]
LocLimitType4BI	O	limit_type*	BI limit type	Integer [0]

O: optional, R: required, CR: conditionally required