CDL NGFM Prototype

The purpose of the CDL NGFM Prototype is to facilitate the development and quality assurance of the production Contract Definition Language (CDL) NextGen Financial Model (NGFM) system. In addition to bespoke testing of coded contract losses, the prototype incorporates Period Event Table (PET) simulation and the contract model to produce a Period Loss Table (PLT).

Note that all calculations and storage are confined to the Prototype itself (in-memory), a local file path, or a sandboxed database. The results of any of the Prototype's calculations are not stored back into an RDM or EDM; however, RDMs and EDMs are referenced by the Prototype.

This document serves as a User Guide and summary code view of the CDL NGFM Prototype.

Contents

[1 Prototype Installation 3](#_Toc327800211)

[1.1 Source Code 3](#_Toc327800212)

[1.2 Application Installation 3](#_Toc327800213)

[2 User Interface 3](#_Toc327800214)

[2.1 Primary Window 3](#_Toc327800215)

[2.2 Database Connection Settings 4](#_Toc327800216)

[2.3 Portfolio Loader (Migration Tool) 5](#_Toc327800217)

[2.4 Contract Model Settings 5](#_Toc327800218)

[2.5 Generators 6](#_Toc327800219)

[2.5.1 Event Details Generator 6](#_Toc327800220)

[2.5.2 PET Generator 6](#_Toc327800221)

[2.5.3 PLT Generator 6](#_Toc327800222)

[2.5.4 SELT Generator 7](#_Toc327800223)

[3 Tutorial (PLT Simulation) 7](#_Toc327800224)

[3.1 Configuration of Database Settings 7](#_Toc327800225)

[3.2 Loading of Portfolio 8](#_Toc327800226)

[3.3 Editing CDL Text 8](#_Toc327800227)

[3.4. Selecting CDL Allocation Scheme 9](#_Toc327800228)

[3.6 Generating Period Event Table 9](#_Toc327800229)

[3.8 Viewing Status and Results 10](#_Toc327800230)

[4. Overview of Source Code Assemblies 11](#_Toc327800231)

[4.1 Contract Model 11](#_Toc327800232)

[4.2 Execution 11](#_Toc327800233)

[4.3 Simulation 11](#_Toc327800234)

[4.4 Tests 11](#_Toc327800235)

[4.5 Utilities 11](#_Toc327800236)

[4.6 Solution 11](#_Toc327800237)

[5. Appendix 12](#_Toc327800238)

[5.1. Listing of Input Files with Headers 12](#_Toc327800239)

[6. Referenced Functional Specifications 14](#_Toc327800240)

# ****1 Prototype Installation****

## ****1.1 Source Code****

Source code is available on the Team Foundation Server ca1tfs10\NextGenProduct at NGModelCertificationTools\CDLNGFMPrototype. The solution was initially developed under the Visual Studio 2010 IDE.

* F# PowerPack (v2.0.0.0): <http://fsharppowerpack.codeplex.com/releases/view/45593>
* Visual Studio 2010 SDK: <http://www.microsoft.com/download/en/details.aspx?id=2680>

In addition to the above prerequisites, the following external libraries are required (for convenience they are included in the directory NGModelCertificationTools\CDLNGFMPrototype\lib.

* MigrationTool
* Extreme Numerics

## ****1.2 Application Installation****

For application user installation, the following prerequisites must be installed:

* .Net Framework v4: <http://msdn.microsoft.com/en-us/library/5a4x27ek.aspx>
* F# PowerPack (v2.0.0.0): <http://fsharppowerpack.codeplex.com/releases/view/45593>
* Visual Studio 2010 SDK: <http://www.microsoft.com/download/en/details.aspx?id=2680>

Once installed, create a folder on the target machine, copying over all .dll, .exe and .config files located in \NGModelCertificationTools\CDLNGFMPrototype\bin

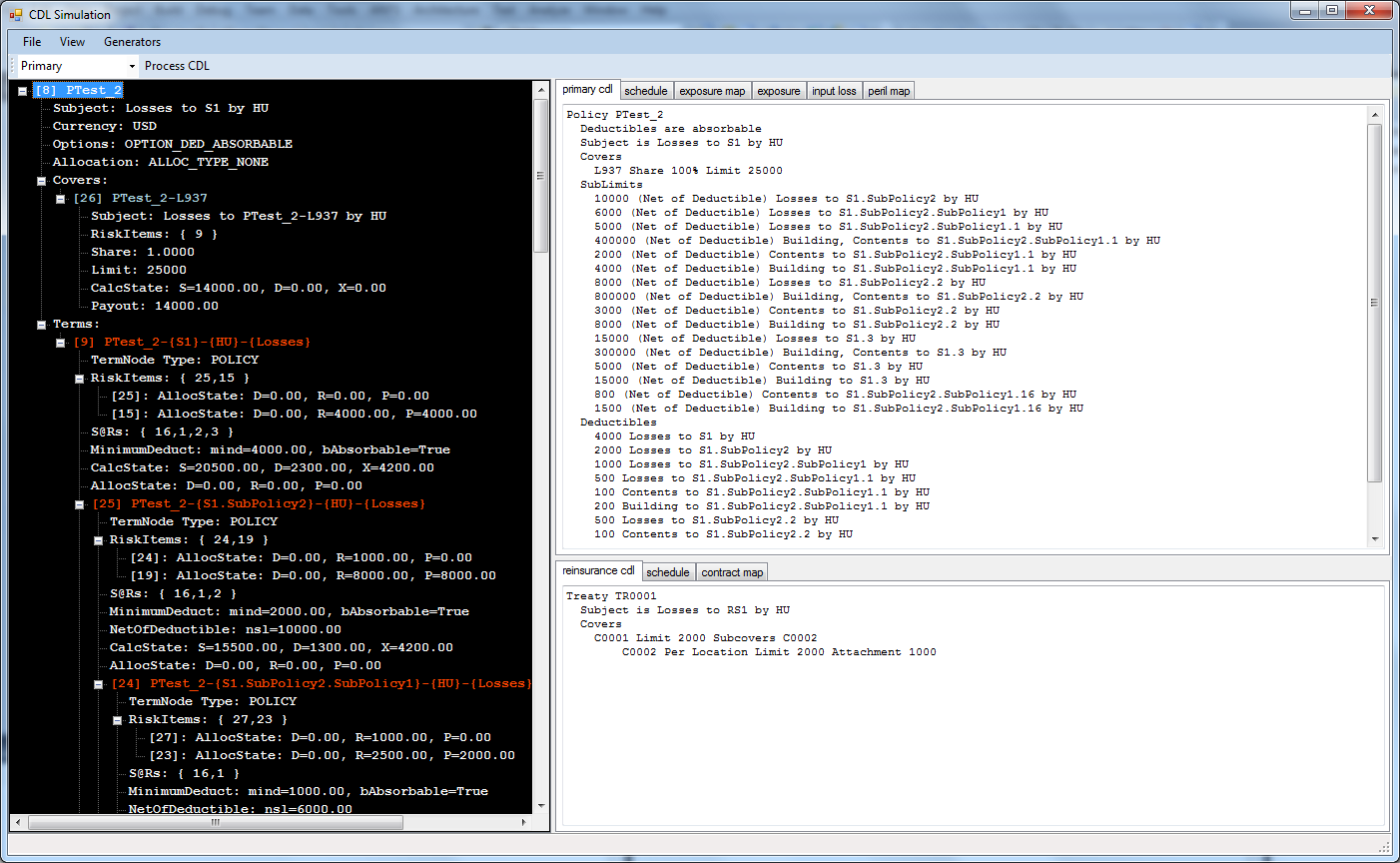
# 2 User Interface

## 2.1 Primary Window

The primary user interface (Figure 1) consists of Contract Definition Language (CDL) input as well as a graphical representation (tree hierarchy) of the processed input. The input on the right side of the primary window is further separated into two sections: the top Primary CDL and the bottom Reinsurance CDL. The input on the right side of the primary user window is fully editable.

Once CDL grammar has been entered, the "Process CDL" button at the top left of the window will update the portfolio tree hierarchy. If there are any errors in the input, however, the tree will not process. Error messages regarding faulty input have yet to be implemented.

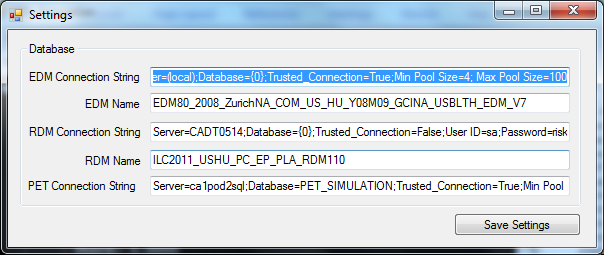
The default view of the portfolio is on the Primary portfolio. Changing the drop down to 'Reinsurance' displays the processed Reinsurance Tree Hierarchy. To expand all tree nodes within the specified portfolio, select the menu item View -> Expand Tree.



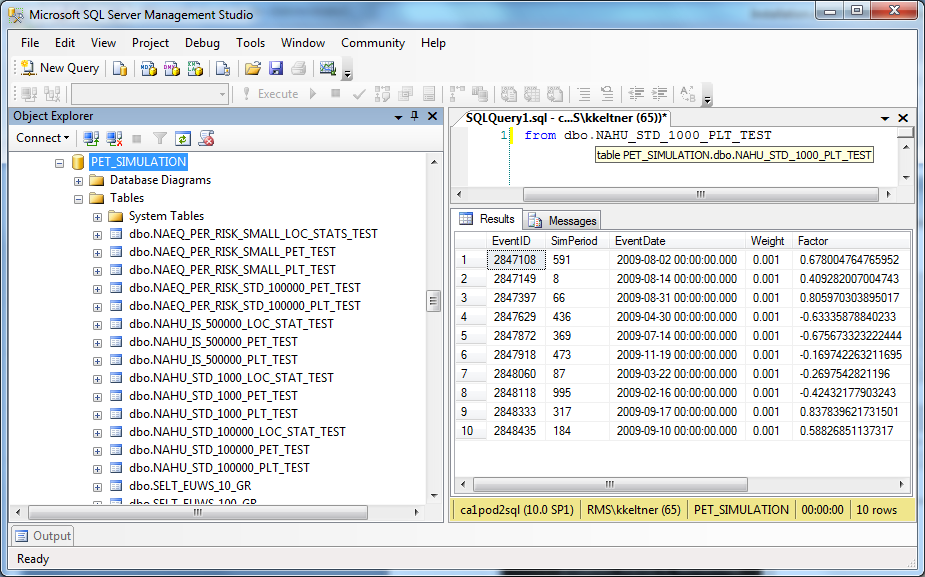
*Figure 1*

## 2.2 Database Connection Settings

The Database Connection Settings dialog (View->Database Settings) allows for user specification of Database connection strings for use by the Prototype. If you are unfamiliar with the syntax of Connection Strings within the .Net framework, contact a developer who can help you enter a correct string. Note that the Prototype expects 'Database={0}' within the connection string as the Prototype will dynamically insert the Database name specified in the EDM Name field. RDM behaves similarly. (Figure 2)



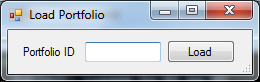
*Figure 2*



*Figure 3*

## 2.3 Portfolio Loader (Migration Tool)

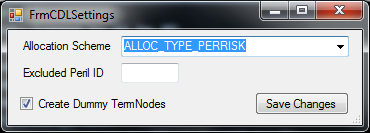
The portfolio loader (File -> Load Portfolio) provides access to the migration tool from within the CDL NGFM prototype. After the EDM has been specified and saved in the Database Connection Settings dialog (Section 2.2), enter a Portfolio ID into the portfolio loader dialog (Figure X) and click Load. If the migration tool runs without exception, the corresponding CDL files will be loaded in the Primary User Interface.



*Figure 4*

## 2.4 Contract Model Settings

The Contract Model Settings dialog (View -> CDL Settings)

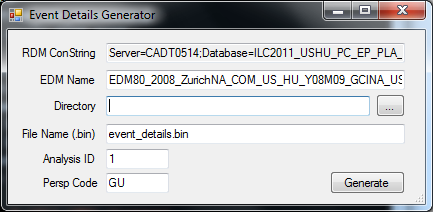


*Figure 5*

# 2.5 Generators

## 2.5.1 Event Details Generator

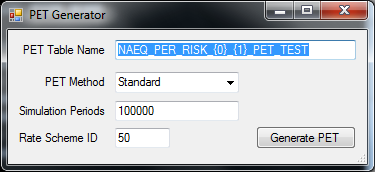
The generation of a PLT requires that an Event Details binary file is available (these can be quite large ~15GB). The binary file that is generated serves as a cached Event Loss Table (ELT) with information for all location coverages that are associated with an event. Figure X, displays the dialog that generates the Event Details binary file, showing which RDM joined with which EDM will be queried (the RDM and EDM must reside on the same Database Server), the output directory, the name of the binary file to be generated, the RDM Analysis ID and Persp Code.



*Figure 6*

## 2.5.2 PET Generator

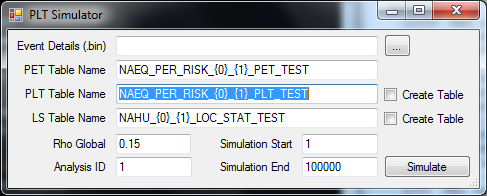
The Period Event Table (PET) generator builds a PET and stores the results where specified in the Database Settings dialog. There are currently two options for generation, standard sampling and importance sampling.



*Figure 7*

## **2.5.3 PLT Generator**

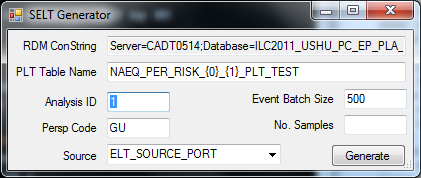
Once an Event Details binary file and a PET have been generated, the (Period Loss Table) PLT generator will simulate and store results where specified in the PLT Simulator dialog. Note that simulations can be executed in parallel as each simulation is an independent task. When performing multiple simulations, however, duplicate Event Details binary files should be referenced for best performance. As an example for four simulations of total size 100000: create tables for the first simulation with specified Simulation Start=1, Simulation End=25000; uncheck create tables for the second simulation with specified Simulation Start=25001, Simulation End=50000;...; uncheck create tables for the fourth simulation with specified Simulation Start=75001, Simulation End=100000.



*Figure 8*

## 2.5.4 SELT Generator

The Spray Mode Event Loss Table (SELT) Generator was built fairly quickly for test purposes (i.e. financial structures are hard coded within the solution) and performs a basic SELT simulation. The SELT simulation implements latin hypercube sampling with bin size equal to No. Samples and stores the generated results into a PLT as specified in the dialog. When querying the results, keep in mind GRLoss correspond to the first treaty and TRLoss correspond to the second treaty that have been hard coded.



*Figure 9*

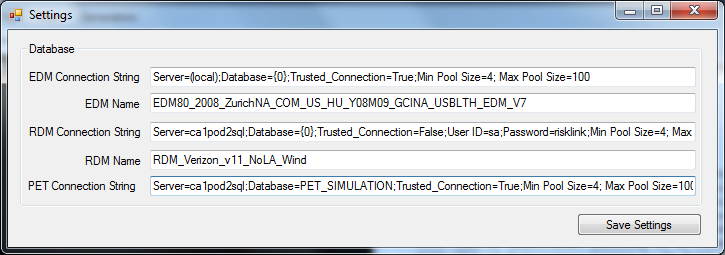
# 3 Tutorial (PLT Simulation)

This simple tutorial shows how to perform a PLT Simulation that utilizes the Contract Model for financial model calculations. Note that the entire front to back process has not been thoroughly tested and that results should

## 3.1 Configuration of Database Settings

Within the Database Settings Dialog (Section 2.2), specify the following EDM and RDMs:

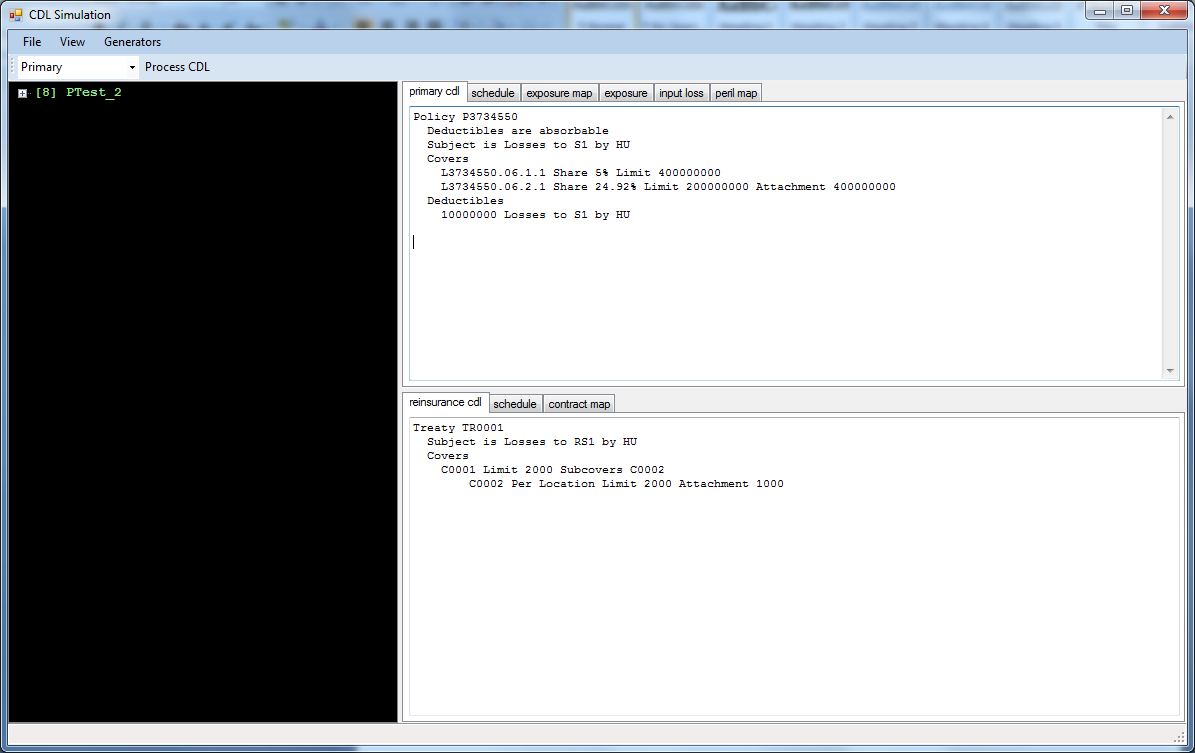
* EDM: EDM80\_2008\_ZurichNA\_COM\_US\_HU\_Y08M09\_GCINA\_USBLTH\_EDM\_V7
* RDM: RDM\_Verizon\_v11\_NoLA\_Wind



*Figure 10*

## 3.2 Loading of Portfolio

Using the portfolio loader (Section 2.3), specify Portfolio ID 4 and click Load. The "primary cdl" tab, should be similar to that in Figure X.



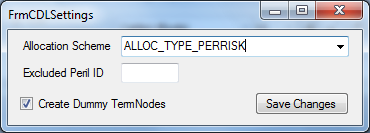
*Figure 11*

## 3.3 Editing CDL Text

While the CDL for the Primary portfolio will be valid (it was processed by the migration tool), the Reinsurance portfolio CDL must now be updated to reference the new Primary portfolio. Specifically, we must update the "contract map" tab to ensure that the Reinsurance contract is correctly pointing to the Primary contract. To update the reference, change the text currently displayed in "contract map" from "0,PTest\_2" to "0,P3734550". Now click the 'Process CDL' button at the top of the main window (this may take a moment).

## 3.4. Selecting CDL Allocation Scheme

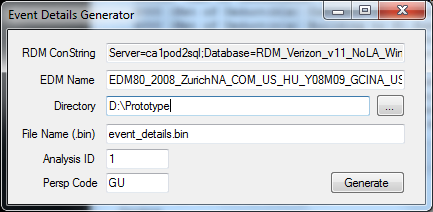
You will need to specify an allocation scheme for processing the Reinsurance Contract payout. In the Contract Model Settings (Section 2.4) select ALLOC\_TYPE\_PERRISK as the specified Allocation Scheme and click 'Save Changes'.



*Figure 12*

**3.5 Generating Event Details Binary File**

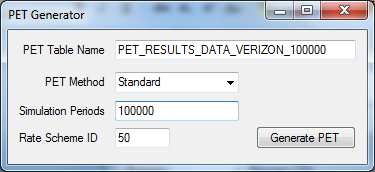
In order to complete a PLT Simulation, an Event Details Binary File needs to be available. Open the Event Details Generator 2.5.1 and specify a directory to save the .bin file (e.g 'D:\Prototype'). Click 'Generate'. Since the Event Details binaries are quite large (~16GB for this ELT) this procedure will take some time - you will, however, be notified when complete.



*Figure 13*

## 3.6 Generating Period Event Table

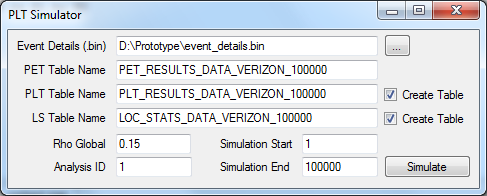
Once the Event Details Binary File has been created, the final requirement for generating a PLT is to have access to a generated PET. Open the Period Event Table Generator (Section 2.5.2) and specify the PET Table Name 'PET\_RESULTS\_DATA\_VERIZON\_100000.' Since we're operating on the WS peril, we need to specify Rate Scheme ID 50. Clicking Generate will then create the PET 'PET\_RESULTS\_DATA\_VERIZON\_100000' in the PET\_SIMULATION Database.



*Figure 14*

**3.7 Generating Period Loss Table**

With the Event Details binary file and the PET generated, you can now perform a PLT simulation. Open the PLT Generator (Section 5.2.3) and click '...' to browse for the Event Details binary file that you just generated. You must also specify the PET Table Name from Section 3.6 in addition to specifying the desired Table Names for the simulated results (PLT Table Name and LS Table Name). If the tables do not exist, be sure to check Create Table for both PLT and LS (Location Statistics). Click Simulate when the fields have been entered to execute a simulation task.



*Figure 15*

## 3.8 Viewing Status and Results

Once the simulation has completed (you can view whether or not a simulation task is still executing through View -> Task Monitor) the results will be available in the PET\_SIMULATION Database as specified in the Database Settings dialog.

# 4. Overview of Source Code Assemblies

The following section enumerates the assemblies (.dll's) that are included in the CDL NGFM Prototype system.

## 4.1 Contract Model

* **RMS.CDLModel**: Domain model of the prototype. This assembly includes the rich object model for structuring a portfolio of (re)insurance portfolios as defined via the contract model.
* **RMS.CDLParser**: F# Lexing and Parsing engine for the contract model
* **RMS.CDLProcessor**: Assembly housing operations on the RMS.CDLModel.

## 4.2 Execution

* **RMS.CDLSimulation**: Entry point of the application (Windows Form)
* **RMS.PETSimulation**: Legacy console version of PET Generation and PLT Simulation

## 4.3 Simulation

* **RMS.Simulation**: Assembly implementing PET Generation, PLT Simulation and supporting calculatons

## 4.4 Tests

* **RMS.CDLModelTest**: Unit tests for the RMS.CDLModel assembly
* **RMS.PETGeneratorTest**: Legacy test console for PET Generation
* **RMS.PETMergeTest**: Legacy test console for PET Merge Operation

## 4.5 Utilities

* **RMS.Utilities**: Database and File Handling, miscellaneous supporting (utility) classes.

## 4.6 Solution

As this solution is a prototype, please be aware that it has not gone through a formal quality assurance process and therefore not all functionality is guaranteed to work as expected. While the prototype integrates the contract model with loss simulation engine to produce a PLT, the integration / refactoring of the two systems into a single system has left some less than desirable designs that needs to be refactored. Specifically, within the RMS.CDLProcessor assembly, the base class InsuredLossProcessor.cs and all related classes could bennefit from refactoring. In addition the two independent systems supported logging via CDLMessage.cs (depricated) and SimulationMsg.cs (RMS.Utilities). Logging is currently not supported, however not much work would be required to facilitate logging through the SimulationMsg class.

# 5. Appendix

## 5.1. Listing of Input Files with Headers

|  |  |
| --- | --- |
| **EXPOSURE\_ID** | **LOCATION\_CVG\_ID** |
| 0 | 1 |
| 1 | 2 |
| 2 | 5 |
| 3 | 6 |
| 4 | 9 |
| 5 | 10 |
| 6 | 53 |
| 7 | 54 |

***Exposure Map***

|  |  |  |  |
| --- | --- | --- | --- |
| **EXPOSURE\_ID** | **EXPOSURE\_VALUE** |  |  |
| 0 | 5000 | 1 | 1 |
| 1 | 2500 | 1 | 1 |
| 2 | 10000 | 1 | 2 |
| 3 | 5000 | 1 | 2 |
| 4 | 20000 | 1 | 3 |
| 5 | 10000 | 1 | 3 |
| 6 | 2000 | 1 | 16 |
| 7 | 1000 | 1 | 16 |

***Exposure***

|  |  |  |
| --- | --- | --- |
| **PERIL\_ID** | **EXPOSURE\_ID** | **DAMAGE\_RATIO** |
| 2 | 0 | 0.5 |
| 2 | 1 | 0.2 |
| 2 | 2 | 0.9 |
| 2 | 3 | 0.5 |
| 2 | 4 | 0.2 |
| 2 | 5 | 0.1 |
| 2 | 6 | 0.4 |
| 2 | 7 | 0.2 |

***Input Loss***

|  |  |
| --- | --- |
| **PERIL\_ID** | **PERIL\_DESCR** |
| 1 | EQ |
| 2 | HU |

***Peril Map***

|  |  |
| --- | --- |
| **CONTRACT\_ID** | **CONTRACT\_NAME** |
| 0 | PTest\_2 |

***Contract Map***

# 6. Referenced Functional Specifications

* **Proportional Allocation:** <http://sharepoint/ProdSolutions/NGFM/CM-CDL/CDL%20for%20Beta%202%20Project%20Documents/FS001_CDL_PropAllocContractLosses.docx>
* **Resolution:** <http://sharepoint/ProdSolutions/NGFM/CM-CDL/CDL%20for%20Beta%202%20Project%20Documents/FS_006_CDL_Resolution%20in%20CDL.docx>
* **Contract Model:** <http://sharepoint/ProdSolutions/NGFM/CM-CDL/Project%20Documents/CDL%20Version%200%201.docx>
* **Per Risk Allocation:** FS\_Facultative\_Per-Risk\_v2.docx (included within solution)
* **CAT Reinsurance:** <http://sharepoint/ProdSolutions/NGFM/CM-CDL/CDL%20for%20Beta%202%20Project%20Documents/FS_007_CDL_CATReinsurance.docx>