RISK DATA OPEN STANDARD

RISK DATA OPEN STANDARD

Overview and Concepts

JANUARY 2020

Contents

It's Time for a Better Data Model

The Risk Data Open Standard (RDOS)is a data specification that can represent a complete and unambiguous expression of risk, from exposure and coverage through analysis. It is flexible and extensible, able to accommodate any type of model and any line of business. It can be implemented on any data technology platform, making it a future-proof vehicle for industry-wide information exchange.

Limitations of Existing Data Standards

The Exposure Data Module (EDM) and Results Data Module (RDM) have been widely used for data exchange across the insurance marketplace for over 20 years. These data specifications were developed by Risk Management Solutions (RMS) to support RMS catastrophe models, and they are well-documented, allowing users to dig into the databases with their own queries and extract data. Other similar data formats from other providers exist, and these are also used for data exchange and analysis (though to a lesser extent), but all existing formats share key shortcomings that are blockers to improved efficiency, automation, better models, and the analysis of new types of risk.

The existing data standards have these key limitations:

- **Rigid**–The EDM and other formats were built to support specific model approaches and algorithms. An effective industry standard should have the flexibility to accommodate any modeling approach. Current and future models must be able to handle larger data sets, unique and complex contract forms, and the underlying technology that make new modeling approaches possible.
- **Loss of Information**–Today's commonly used data exchange formats do not capture all of the information required to show a counter party, or even another user in the same company, the complete, unambiguous picture of how exposure and coverage interact, and how models were used to produce analysis results. This lack of completeness forces skilled analysts to do forensic analysis to understand a submission or past analysis. A modern standard should contain all of the information required to understand an analysis, from exposure and coverage through results.
- **Property-focused**–Current data formats were created to support property risks. Over time, models have expanded beyond property, but the data formats have either constrained modeling options, or completely new formats have been required to support the new models. A new data model must have the flexibility to cover any line of business as well as emerging or unknown risks that can spring from an evolving economy and present new demand and opportunities in risk financing.

Design Goals of the Risk Data Open Standard

The Risk Data Open Standard (RDOS) was designed with the following goals:

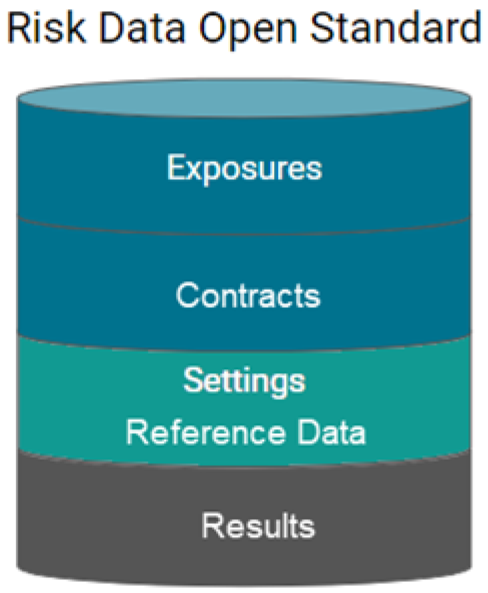
- Represent exposure and coverage clearly, with the flexibility to handle any new or complex coverage that is needed in the real world, as well as the coverage interaction and business structure that has generated specific financial perspectives.

- Express exposures in a way that allows new lines of business or types of risk to be included in the schema.
- Provide a complete, repeatable, and auditable data set that provides all of the information needed to understand how exposure was analyzed, including the models, settings, and reference data that were used to produce results. This includes changes to model data that comprise a bespoke view of risk.
- Use a modular design that simplifies modifications in the future.

RDOS Includes Complete Information

The EDM contains exposure and contract terms. The RDM contains modeling results with limited information about how models are used to produce the results. The RDM contains only limited information about the links between the results and the exposure (EDM) that produced the results. Typical client environments have hundreds of unlinked EDMs and RDMs.

In contrast, the RDOS contains both exposure and results and adds more complete information about model settings, links between exposure and results, and other reference data. This information is unified in the same data model, providing a clear picture about deriving analysis results.



RDOS Facilitates Data Exchange

RDOS is a unified data model that consolidates exposure, contract coverage, reference data, and results. EDMs and RDMs are full subsets of RDOS. This means that EDMs and RDMs can be transformed to RDOS with no data loss.

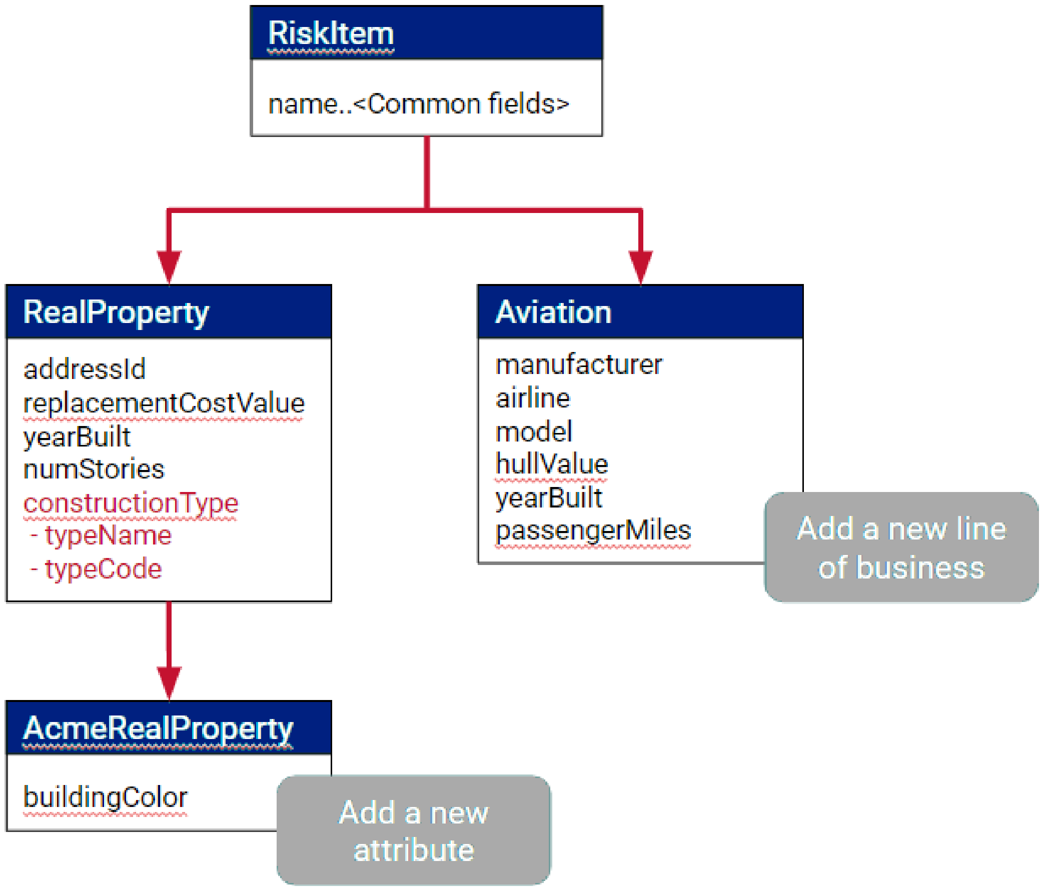
Exchange of RDOS is easy, efficient, and flexible. The default exchange format of RDOS includes the schema definitions along with the data so that the schema is available regardless of choice of data processing framework or programming language.

External systems can consume RDOS directly or use transformations to consume RDOS in other formats. When RDOS is received by a consuming system, the system can determine the schema and hierarchy without any additional external information. This facilitates easy exchange for any type of risk extended by different parties. During exchange, you can map a counter-party's entity definitions to your own, or you can choose to accept only the attributes you have in common.

RDOS Provides Extensibility

Insurers and reinsurers increasingly seek to apply enhanced pricing and exposure analytics to lines of business beyond property catastrophe. The EDM and RDM data models are rigid and difficult to extend to new classes of risk and associated coverage.

RDOS schema extensibility empowers open modeling and the ability to customize a personal view of risk. The RiskItem entity, which already includes the sub-type RealProperty, can be extended to new sub-types that represent new lines of business, like Aviation. Existing attributes, like RealProperty attributes, can be extended to capture new ones.



RDOS Defines Unified Exposure Entities

In the EDM, property characteristics and insurance/reinsurance structures are spread across multiple tables, sometimes redundantly. Many EDM entities combine both the description of risk and the financial/coverage terms. This means that you have to structure joins across multiple tables to derive data insights.

In contrast, the RDOS is modeled with the following well-defined entities to address specific domain concepts.

Contract–The central component of the data model is the contract, which is a binding agreement between the insurance provider and insured customer.

Risk–Every contract has a single subject, which for primary insurance can be a single risk or a schedule. A risk is a collection of primary exposures, and a schedule is a collection of risks.

Risk Item–A risk item represents something that can be damaged by events. Risk items can be of multiple types, such as real property, contained property, time element, and so on. Each of the sub-types is a separate risk item. The level of risk associated with risk items is the risk exposure, which indicates to what extent the risk items are covered by the insuring party for different coverage types.

Risk Exposure–The Risk Exposure entity provides the link between a risk, risk item, and its member exposures, such as building, content, or business interruption exposure. It stores the insurable interest, which refers to the ownership share. For some lines of business, like offshore platform, it could be less than 100% because these can have shared ownership. Insurable interest scales the valuation before the ground-up loss takes effect.

Portfolio–A grouping of contracts based on some arbitrary criteria is a portfolio. The portfolio may be based on geographical region, business unit, or some other criteria that makes sense within the organization. A contract can be a member of multiple portfolios.

Structure–A structure is a container for financial positions and treaties used to define business hierarchies and contract interactions.

ExposureSet–An exposure set is a container for primary contracts and associated portfolios, risks, and risk items.

See Exposure Schema for more information.

Implement RDOS to Suit your Organization

Risk Data Open Standard is a *logical data model*, meaning that it provides schema specifications and business rules to standardize the data model independent of a specific database technology. The database technology is your choice depending on your own access patterns, performance requirements, and scaling needs. Select from the following technology options:

- Relational–Implement the same RDOS specification as a traditional relational database using SQL, Postgres, MySql, Access, and others.
- Interchange format–Keep the RDOS specification in simple, light-weight interchange format using JSON or CSV.

To help you adopt the Risk Data Open Standard, the Steering Committee is providing scripts that implement the schema as SQL tables to simplify relational querying and user understanding.

RDOS Provides an Unambiguous Picture of Financial Perspectives

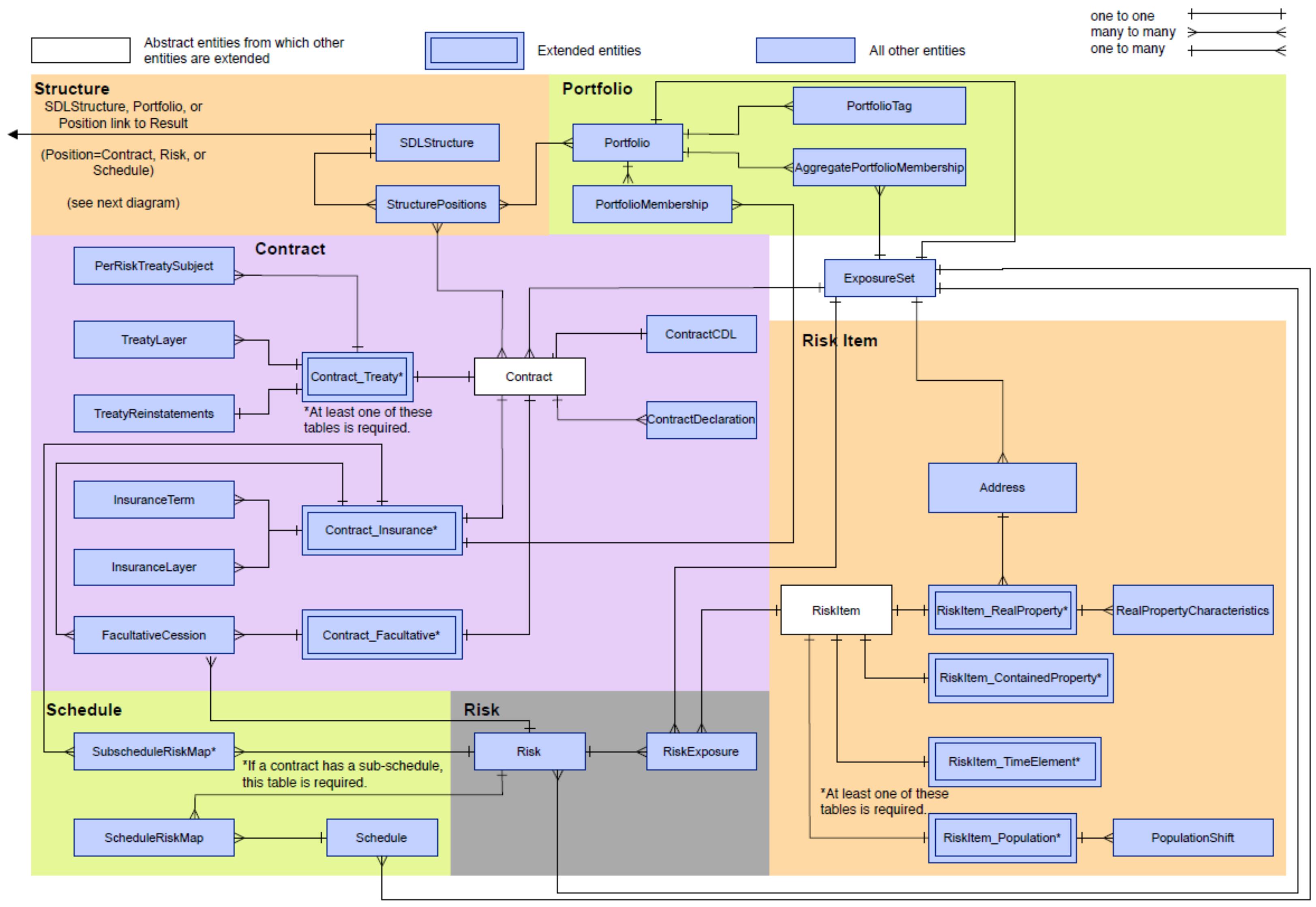
Contract Definition Language (CDL) enables the expression and evaluation of many types of insurance and reinsurance contracts, but there are two important aspects of coverage that contract terms alone cannot describe:

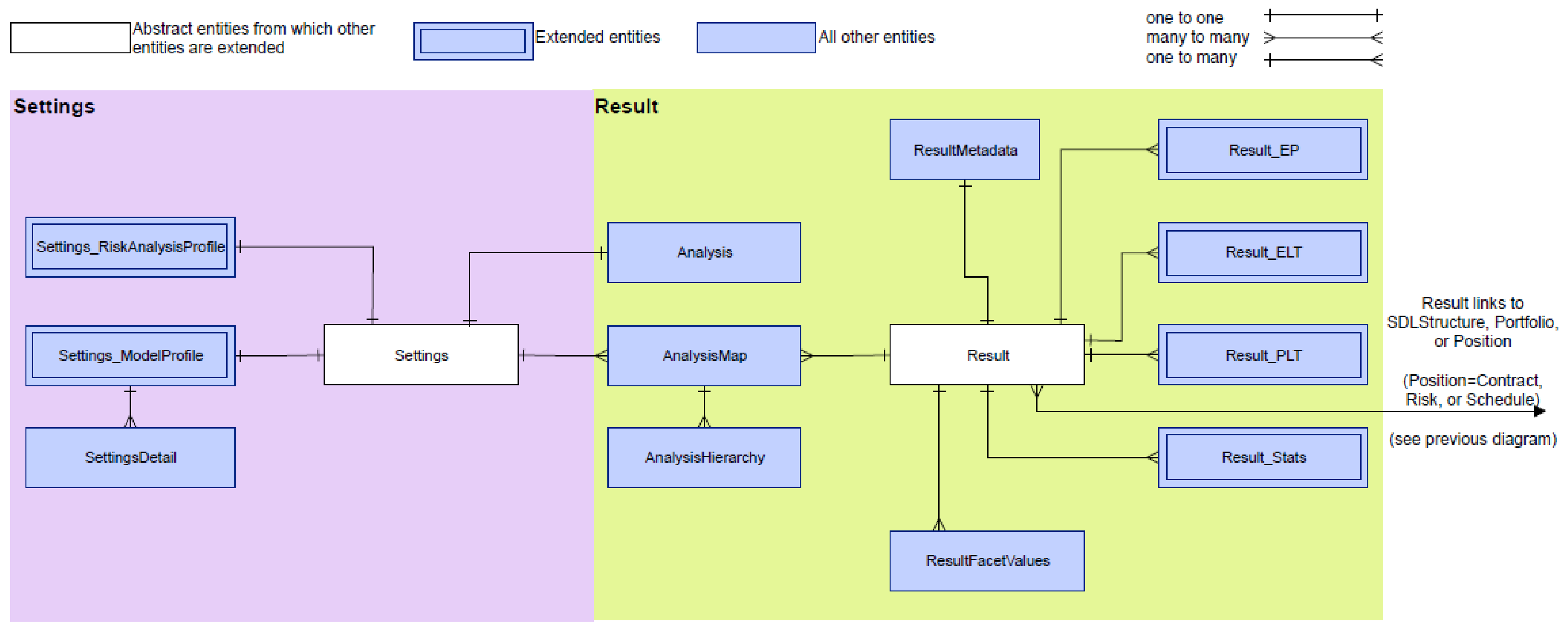
- Forming a precise subject of a treaty, e.g., the all-perils treaty benefits from the Florida Hurricane Cat Fund
- Specifying and evaluating financial positions, e.g., Gross, Ceded, and Net-of-Retro for each treaty

Structure Definition Language (SDL) enables industry professionals to solve these problems quickly, accurately, and transparently. A Structure written in SDL models the flow of quantities such as risk, loss, premium, and expenses as they pass through CDL contracts and other transformations.

Read next: RDOS Concepts

Entity Relationship Diagram


Abstract entities from which other entities are extended
Extended entities
All other entities
one to one
many to many
one to many
Structure
SDLStructure, Portfolio, or Position link to Result
(Position=Contract, Risk, or Schedule)
(see next diagram)
SDLStructure
StructurePositions
Portfolio
PortfolioTag
AggregatePortfolioMembership
PortfolioMembership
ExposureSet
Contract
PerRiskTreatySubject
TreatyLayer
TreatyReinstatements
Contract_Treaty*
*At least one of these tables is required.
Contract
ContractCDL
ContractDeclaration
InsuranceTerm
InsuranceLayer
Contract_Insurance*
FacultativeCession
Contract_Facultative*
Risk Item
Address
RiskItem
RiskItem_RealProperty*
RealPropertyCharacteristics
RiskItem_ContainedProperty*
RiskItem_TimeElement*
*At least one of these tables is required.
RiskItem_Population*
PopulationShift
Schedule
SubscheduleRiskMap*
*If a contract has a sub-schedule, this table is required.
ScheduleRiskMap
Schedule
Risk
Risk
RiskExposure


Abstract entities from which other entities are extended
Extended entities
All other entities
one to one
many to many
one to many
Settings
Result
Settings_RiskAnalysisProfile
Settings_ModelProfile
SettingsDetail
Settings
Analysis
AnalysisMap
AnalysisHierarchy
ResultMetadata
Result
ResultFacetValues
Result_EP
Result_ELT
Result_PLT
Result_Stats
Result links to SDLStructure, Portfolio, or Position
(Position=Contract, Risk, or Schedule)
(see previous diagram)

RDOS Concepts

This document discusses the framework of the Risk Data Open Standard (RDOS) concepts and schema. It describes the entities that comprise the exposure, settings, results, and reference schemas. While it provides some field-level details, it primarily focuses on main areas and concepts. For the full specification of fields, data types, and values, see the RDOS schema specification RDOS_Schema.xlsx.

Conventions

Table names–Table names are written as one word using initial caps for each word, e.g., AggregatePortfolioMembership. When a table is extended to create a new table, the name is written as ParentTable_ExtendedTable, e.g., RiskItem_RealProperty.

Field names–Field names are written as one word using *camel* case, where the first letter of the first word is lowercase, and the first letters of subsequent words are uppercase, e.g., replacementCostValue.

Abstract and Extended Entities

RDOS provides a layer of abstraction between the underlying storage technology and the physical representation. It includes the following abstract entities that act as master tables or templates from which other tables have been extended. The extended tables include all the fields from the abstract entities and add their own fields; see RDOS Entity Relationship Diagram and RDOS Provides Extensibility.

- Contract–Extended to create:
    - Contract_Insurance
    - Contract_Facultative
    - Contract_Treaty
- RiskItem–Extended to create:
    - RiskItem_RealProperty
    - RiskItem_ContainedProperty
    - RiskItem_TimeElement
    - RiskItem_Population

- Result–Extended to create:
    - Result_EP
    - Result_ELT
    - Result_PLT
    - Result_Stats
- Settings–Extended to create:
    - Settings_RiskAnalysisProfile
    - Settings_ModelProfile

Read next: Exposure Schema

Exposure Schema

The following entities in the exposure schema capture the physical location, characteristics, financial details, and organization of the exposures in a data set. See Entity Relationship Diagram for a visual representation of the schema. See the *Exposure Schema* tab of the RDOS schema specification RDOS_Schema.xlsx for the full specification of fields, data types, and valid values.

Contract

The foundational concept of the RDOS data model is the contract, which is a binding agreement between the insurance provider and insured customer. The RDOS allows contract expression through the Contract Definition Language (CDL). The **ContractCDL** table stores the CDL string. For more information about CDL, see Contract Definition Language Semantics and Detailed FinancialContract Model Examples.

Tables to represent insurance contracts:

- **Contract_Insurance**–Extended from **Contract** to create the insurance contract record. For reinsurance contracts, use the **Contract_Treaty** table. The subject of a contract is a riskId or a scheduleId.
- **InsuranceContractLayer** and **InsuranceContractTerm**–Define the contract if you do not use CDL. For more information about the lossTypeCode and causeOfLossCodes in both of these tables, see Causes of Loss and Loss Types.
    - Contract Layers–Layers define the promise of payment.
    - Contract Terms–Contracts include two types of *terms*–deductible and sublimit–which reduce subject losses. Terms of a contract define their scope through a subject constraint (for example, contents losses to a certain location by hurricane, losses to a certain subschedule of locations by earthquake). Terms also typically have an amount (e.g., 10k USD, 2% RCV Covered) and perhaps some modifiers to their behavior (e.g., franchise deductible, aggregate sublimit). A contract can have as many deductibles and as many sublimits as needed.

Tables to represent treaties:

- **Contract_Facultative** and **FacultativeCession**–Define facultative contracts.
- **PerRiskTreatySubject**–Defines the portfolios that are the subjects of per risk treaties.
- **Contract_Treaty**–Creates the reinsurance contract record.
- **TreatyLayer** and **TreatyReinstatements**–Define the treaty if you do not use CDL.

Table applicable to all contracts:

- **ContractDeclaration**–Defines declarations for all contracts if you do not use CDL. Stores declarations per contract and per contract layer. Identifies each contract using the contractId field. Declarations act as metadata to aid search and filter, such as contractCedantCode, underwriterShortName, producerShortName, branchShortName, brokerShortName, and multiple legacyUserId and legacyUserTxt fields for free-form entry.

Risk and Risk Items

Contracts cover one or more risks, captured in the **Risk** table.

Each risk contains one or more risk items, which can be of multiple types, such as real property, contained property, time element, and population. Each of the sub-types is a separate risk item and together, a collection of risk items represent the risk associated with the contract.

When you have information about a group of buildings in aggregate, but do not have individual building level information, use the **Risk.numberOfUnits** field to indicate the number of individual risks that are represented by the aggregate risk.

Tables to represent risks and risk items:

- **Address**–Captures data about a real property risk item's geographic location, such as:
  - streetAddress–The building number and street
  - postalCode–The predominant postal code, such as 5-digit ZIP Code in the U.S.
  - cityName–City name
  - admin1–Administrative divisions are geographic divisions that vary by country. Admin1 represents large geographic divisions, such as states and provinces, such as state in the U.S.
  - countryCode and countryScheme–Store the code and scheme used for the code, such as ISO2N for US.
  - many more
- **Population** and **PopulationShift**–Capture data about human populations, such as for workers compensation analysis.
- **RiskItem_ContainedProperty**–Captures contents exposure.

- **RiskItem_RealProperty**–Captures building exposure and acts as a parent risk item for any RiskItem_ContainedProperty and RiskItem_TimeElement. Stores the following attributes that are often used to adjust the vulnerability of exposure for peril model analyses:
  - constructionSchemeName and constructionCode
  - occupancyschemeName and occupancyCode
  - floorArea
  - numberOfStories
  - yearBuilt
- **RiskItem_TimeElement**–Captures business interruption exposure.
- **RealPropertyCharacteristics**–Captures additional modifiers associated with a real property risk item. Refer to the documentation for specific peril models to know which modifiers are considered. The RiskItem tables in the Reference schema store the available modifier options.

The level of risk associated with risk items is the risk exposure, captured in the **RiskExposure** tabled. The insurableInterest field indicates to what extent the risk items are covered by the insuring party for different coverage types.

Schedule

When multiple risks are associated with a contract, they are combined into a schedule, which allows the contract to apply the same terms and conditions to all the risks collectively, rather than listing each one individually. Schedules are captured in the **Schedule** table. The **ScheduleRiskMap** table maps risks to the schedules with which they are associated.

Like schedules, sub-schedules define a named collection of risks. They are an optional subset of the risks in a schedule that is covered by a contract. Use them in a contract to replace a list of risks when you want to define constraints on sublimits, deductibles, and covers. The **SubscheduleRiskMap** table maps risks to the sub-schedules with which they are associated.

Portfolio

A grouping of contracts is a portfolio, and portfolio records are captured in the **Portfolio** table. Contracts that comprise a portfolio are stored in the **PortfolioMembership** table. The portfolio may be based on geographical region, business unit, or some other criteria that makes sense within an organization.

For aggregate portfolios, the **AggregatePortfolioMembership** table captures aggregate geographic and financial information.

Define the metadata tags you want to use for portfolios in the **PortfolioTag** table.

Structure

Finance and accounting use the concept of *positions* to express how a result should be calculated and how it belongs to an organization. A structure is a container for these positions. RDOS expresses the position framework and captures the loss flow using Structure Definition Language (SDL).

Portfolios and **SDLStructures** are both structures. Structures contain positions. Portfolios *implicitly* define the positions they contain, such as Gross and Net Loss Pre-Cat. SDLStructures *explicitly* define the positions they contain, some of which can be cat treaties. SDLStructure positions can depend on Portfolio positions.

The **SDLStructure** table stores the SDL string. The **StructurePositions** table allows you to define structures without using SDL.

Exposure Set

If you want to group primary contracts and associated portfolios, risks, and risk items, you can define an exposure set, stored in the **ExposureSet** table.

Geocoding Details

Geocoding is the process of translating an address to an exact location. It’s the foundation of catastrophe analytics, and its importance cannot be understated. Small differences in exposure location can mean large changes in loss, which impacts underwriting, capital management, and reinsurance purchasing.

RDOS provides numerous fields in the **Address** table that can be populated by geocoding engines. These fields capture detailed information about the geocoding process and about the exposure and its location.

The RDOS uses the following **Address** table fields to support two standards used to describe the geocoding resolution of a specific address:

- **geoModelResolutionCode**–Stores the 0-14 match levels used by many geocoding vendors to represent a range of geocoding resolutions, from high-resolution (parcel, building, street) to postal code, neighborhood, and district.

- **ACORDResolutionCode**–Stores the 99 match levels developed by the Association for Cooperative Operations Research and Development (ACORD).

The RDOS uses the following fields to provide a measure of geocoding accuracy and confidence:

- **geocodingMatchConfidence**–Stores the percentage confidence that applies to the achieved geocoding match. Different geocoding vendors use different approaches for determining confidence thresholds. **Uncertainty** and **uncertaintyBuffer** fields are also available.

- **geocodingLocationCode**–Provides additional accuracy measure for U.S. high-resolution geocoding.

- **geocodingMatchCode**–Stores codes from the geocoder to indicate portions of the input address that matched, did not match, or were modified during the geocoding process.

The following fields provide information about the geocoding data and vintage and an audit trail:

- **geocodingDataSourceId**–Identifies the source of the geocoding data.

- **geoProductVersion**–Identifies the data vintage and software version.

- **isSubmittedAddress**–Can provide an audit trail by indicating whether the address has been modified by the geocoder or whether the original submitted address is unchanged.

- **geoCodedDate**–Records the date the location was geocoded.

The **Address** table provides many detailed geography fields for user input or geocoder input, including:

- Globally unique **GeoId** fields for all address components

- CRESTA **Zones 1-5**

- **Administrative** divisions, levels 1-5, which are geographic divisions that vary by country. E.g., Admin1 represents State in the U.S. and Province in Canada.

- **Building** and **parcel** identifiers

- **Points of interest**

- **Quadkey**–Quadkeys uniquely identify map tiles at a particular level of detail. Map tiles optimize the performance of map retrieval and display.

Read next: Settings Schema

Settings Schema

The entities in the settings schema store the settings used for running analyses. See Table 1 for their descriptions. For the full specification of fields, data types, and values, see the *Results & Settings Schema* tab of the RDOS schema specification RDOS_Schema.xlsx.

Table 1: RDOS Settings Entities

Table	Description
Settings_ ModelProfile	Stores the settings for a given model profile Id. Model profiles are used to run a single type of analysis on a portfolio. Settings include analysis type (EP, footprint, hazard), software version, region code, peril code, analysis currency, loss table type (ELT, PLT). It also identifies the date the model profile was created and who created it.
Settings_ RiskAnalysisProfile	Stores the settings for a given risk analysis profile Id. Risk analysis profiles combine multiple model profiles. Settings include peril code, region code, model profile Id, and simulation set Id.
SettingsDetail	Stores the individual settings and their value that have been defined for a model profile, such as a setting to indicate whether or not post-loss amplification is included in the profile.

Results Schema

The entities in the results schema store the results from running analyses. See Table 2 for their descriptions. For the full specification of fields, data types, and values, see the *Results & Settings Schema* tab of the RDOS schema specification RDOS_Schema.xlsx.

Table 2: RDOS Result Entities

Table	Description
Analysis	Stores the analysis Id, settings Id, and structure Id of every analysis, and indicates whether the analysis was run using a model profile or a risk analysis profile
AnalysisHierarchy	Indicates the model profiles that are included in the risk analysis profile for every analysis that is run using a risk analysis profile
AnalysisMap	Maps analysis Id to results Id
Result_ELT	Stores the event loss table per result Id
Result_EP	Stores the EP results per result Id, return period, and facet, including AEP, OEP, TCE-AEP, TCE-OEP, and EEF
Result_PLT	Stores the period loss table per result Id
Result_Stats	Stores the statistical results per result Id and facet, including AAL, StdDev, and CV
ResultFacetValues	Defines the granularity that applies to each facet Id, including granularities like admin1Code, admin2Code, countryCode, lineOfBusinessCode, cedantCode, and postalCodeGeoId
ResultMetadata	Indicates the structure and position analyzed for every result Id. It also indicates the type of results that are available (EP, stats, ELT, PLT) and the granularity of the analysis (AdminLOB, CedantLOB).

Reference Schema

The entities in the reference schema store reference data used by Contract Definition Language (CDL) and by analyses. Many of the tables include seeded data to which you can add your own. See for their descriptions. For the full specification of fields, data types, and values, see the *Reference Schema* and *Reference Schema--RiskItem* tabs of the RDOS schema specification RDOS_Schema.xlsx.

The entities are grouped as follows:

- Address tables
- Contract/Financial tables
- Population tables
- Portfolio and Structure tables
- Results and Settings tables
- Risk Item tables

Causes of Loss and Loss Types

Cause of Loss

Cause of loss is a collection of perils covered by a contract layer and specified in the Reference schema, **ContractLayer.causeOfLossCode** field. Peril models assess damage from natural catastrophe causes of loss (such as windstorm, earthquake, or flood) to real property. In the same way that some perils include sub-perils (or secondary perils), causes of loss can include subsidiary causes. For example, the earthquake cause of loss could include earthquake-fire following, earthquake-ground shaking, and earthquake-sprinkler leakage.

The Reference schema allows causes of loss to be grouped to form other causes of loss. The following Reference schema tables store information about causes of loss:

- Reference schema, CauseOfLoss table—Defines cause of loss codes used by contracts.
- Reference schema, CauseOfLossHierarchy table—Defines cause of loss groupings. For example, *windstorm wind* and *windstorm storm surge* can be combined into *windstorm*. Contract layers can cover windstorm or cover only windstorm wind or only windstorm storm surge.

The Reference schema defines codes for the causes of loss in Table 3. The cause of loss ALL acts as the parent cause of loss. Map any new cause of loss to a parent cause of loss.

Table 3: Causes of Loss

Cause of Loss Code	**Description**	Parent Code
ALL	All perils	ALL
CS	Severe Convective Storm	ALL
EQ	Earthquake	ALL
FL	Flood	ALL
FR	Fire	ALL
TR	Terrorism	ALL
WS	Windstorm/Hurricane	ALL
WT	Winterstorm	ALL

Loss Types

Loss types represent the kind of exposure that damage to a risk item can cause an exposed party. For example, the following CDL applies a policy coverage level deductible for loss type *building*:

```
Deductibles
 5k for Building
```

The following Reference schema tables store information about loss types:

- Reference schema, LossType table—Defines loss type codes used by contracts.
- Reference schema, LossTypeHierarchy table—Maps any new loss types to parent loss types. See Table 4.

Table 4: LossTypeHierarchy Table

Loss Type	**Parent Loss Type**
BI	Business interruption
Building	Property
Casualty	Loss

Loss Type	Parent Loss Type
Contents	Property
CovA	Building
CovB	Building
CovC	Contents
CovD	BI
Property	Loss

Cause of Loss and Loss Type Hierarchies

Cause of Loss and Loss Type hierarchies are used when you define which term of a CDL contract to process first. For example, for the two terms below:

```
Deductibles
 5k for Building
 10k for Property
```

The deductible for Building would be processed first and send its output as input to the deductible for Property since Building is a child loss type of Property (see Table 4), and loss type is the only constraint on these terms.

Similarly, if the causeOfLossHierarchy defines CS, FL, FR, WS, WT as children of "Climate" which is a child of "ALL," then the terms below in a CDL contract:

```
Sublimits
 100k by WS
 100k by FL
 100k by EQ
 200k Aggregate by Climate
```

the sublimits by WS and FL are processed before (and send their output to) the sublimit by Climate. The sublimit by EQ does not affect any of the other sublimits.