OData Entity Data Model for RESO

What is an EDMX?

An Entity Data Model that describes the entities, their data and the relationships between them. In the Microsoft domain an entity data model is exposed using an EDMX file (.edmx). OData uses EDMX to define resources on a service and then OData clients use the EDMX to reliably consume those resources.

More information is available in the included RESO Web API Version 1.0.2 Standards Specification Documentation.

This document briefly describes the EDMX content provided by the RESO Web API to represent RESO Data Dictionary resources. It serves as an outline for how EDMX documents are produced and consumed.

Please refer to the OData 4.0 specification "Common Schema Definition Language" for comprehensive and normative documentation pertaining to EDMX and OData Metadata requirements.

The EDMX Format

The following XML is the header of an EDMX metadata document. This section of the document declares the XML details for using the EDMX format. There are two XML namespace declarations. The first declares the EDMX schema and the second declares the EDM schema. OData leverages these XML Schemas to implement the metadata for the transport.

```
<?xml version="1.0" encoding="utf-16" ?>
 <edmx:edmx version="4.0" xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx">
  <edmx:dataservices>
  <schema namespace="OData.Models" xmlns="http://docs.oasis-open.org/odata/ns/edm">
[...]
```

Please note that all EDMX documents must conform to the above declarations as specified by the OData standard.

EDMX Samples

The following EDMX samples are specific to the RESO Web API Standard and details the proper use of Annotations in the metadata to support the RESO Web API.

Sample EDMX for Data System implementation is located in Section 8 – Appendices, Appendix 4 Data System XML Schema of the included RESO Web API version 1.0.2 Standards Specification Documentation.

The Data System Resource is an OData Resource defined to represent the data and service endpoints available to a RESO Web API client. This resource is fully described with samples in the RESO Web API Specification. See section 2.6.1 and Appendix 4 within the included RESO Web API Version 1.0.2 Standards Specification Documentation.

Sample EDMX for Data Dictionary Property Implementation

The following sample is a snippet of the top and bottom of a Property Resource as described by the OData EDMX metadata. Please note that the majority of the content is omitted for brevity.

```
<EntityType Name="Property">
<Key>
<PropertyRef Name="PropertyID" />
</Key>
<Property Name="PropertyID" Type="Edm.Int32" Nullable="false" />
<Property Name="PropertyType" Type="RESO.DataStandards.DD13.PropertyType"
Nullable="false" />
<Property Name="AssociationFeeIncludes"
Type="RESO.DataStandards.DD13.AssociationFeeIncludes" />
<Property Name="AboveGradeFinishedArea" Type="Edm.Int32" />
[...]
<Property Name="YearsCurrentOwner" Type="Edm.Int32" />
<Property Name="Zoning" Type="Edm.String" />
<Property Name="ZoningDescription" Type="Edm.String" />
<NavigationProperty Name="GreenVerifications"
Type="Collection(RESO.DataStandards.DD13.GreenVerification)" />
<NavigationProperty Name="Rooms" Type="Collection(RESO.DataStandards.DD13.Room)" />
<NavigationProperty Name="UnitTypes"
Type="Collection(RESO.DataStandards.DD13.UnitType)" />
</EntityType>
```

This section of the content has 4 critical XML tags. These are described as below.

EntityType - This tag declares a given entity and wraps all other tags in this document model.
Key - This tag declares the Key or Unique ID field of the Resource if one is present.
Property - This tag declares a data point or field within the declared Entity Type.
NavigationProperty - This tag declares any relationship between the declared Entity and other Entities as provided by the service. In SQL terms this is akin to a foreign key declaration.

All Resources defined by the Data Dictionary will use these tags to define the data fields available for a given resource. Additionally the server may provide an additional XML tag as an annotation to declare the Standard Name that applies to the metadata element. Please see Section 2.4.8 - Annotations within the included RESO Web API Version 1.0.2 EDMX Standards Specification Documentation.

The following sample demonstrates how to assign a RESO Data Dictionary Standard Name to a field in a given RESO Web API EDMX document.

```
<EntityType Name="Property">
<Key>
<PropertyRef Name="PropertyID" />
</Key>
</Property>
<Property Name="GarageParking_Ext_1" Type="PropertyEnums.GarageParking_Ext_1">
<Annotation Term="RESO.OData.Metadata.StandardName" String="GarageParking" />
</Property>
```

In this sample we see the ‘Annotation’ tag used to declare additional information for the metadata. However, since Standard Name is not natively part of EDMX metadata, this annotation must be declared using the ‘Term’ to define what the annotation means. The RESO Web API defines the term RESO.OData.Metadata.StandardName to allow servers to declare a standard name without breaking the OData specification requirements. This Annotation term is usable for more than a field name and may also apply to an Property (as above), an EntityType (the resource name), EnumType (an enumeration), and an enumeration Member (lookup value). The following samples demonstrate each of these cases.

Entity Type Example

```
<EntityType Name="MyPropertyListings">
 <Annotation Term="RESO.OData.Metadata.StandardName" String="Property" />
<Key>
 <PropertyRef Name="ID" />
</Key>
[...]
```

Enum Type Example

```
<EnumType Name="GarageParking_Ext_1" UnderlyingType="Edm.Int64" IsFlags="true">
<Annotation Term="RESO.OData.Metadata.StandardName" String="GarageParking" />
<Member Name="SpacesPerSqFt" Value="1" />
<Member Name="ParkingLot1Block" Value="2" />
<Member Name="ElectricCarHookup" Value="4" />
```

Enumeration Member Example

<EnumType name="AccessibilityFeatures_Flags" isflags="true">
<Member name="Doors_Swing_in" value="1">
<Annotation term="RESO.OData.Metadata.StandardName" string="Doors - Swing in" />
</Member>
<Member name="Elevator" value="2" />
<Member name="Entry_Slope_less_than_1_foot" value="4">
<Annotation term="RESO.OData.Metadata.StandardName" string="Entry Slope less than 1 foot" />
</Member>
<Member name="Grab_Bars_in_Bathroom_s_" value="8">
<Annotation term="RESO.OData.Metadata.StandardName" string="Grab Bars in Bathroom(s)" />
</Member>

The following samples demonstrate how to declare single and multi-select enumerations using the Annotation method provided by the RESO Web API Standards Specification Documentation.

Single Select Enumeration Example

Single select enumerations in EDMX are used without modification. The EDMX representation provides all of the necessary functionality to describe and use single valued enumerations. The following link refers to how these are defined in the RESO Web API. Please see Section 2.4.9 – Single Valued Lookups within the included RESO Web API Version 1.0.2 EDMX Standards Specification Documentation.

The following sample demonstrates how a single select enumeration is declared for a given field in an entity.

<Property Name="StructureSQFTSource" Type="PropertyEnums.StructureSQFTSource">
<Annotation Term="RESO.OData.Metadata.StandardName" String="StructureSQFTSource" />
</Property>

This sample demonstrates how the enumeration and its values are defined.

<EnumType Name="StructureSQFTSource">
<Member Name="Appraiser" Value="0" />
<Member Name="Assessor" Value="1" />
[...]
<Member Name="DMV" Value="16" />
<Member Name="HCD" Value="17" />

</EnumType>

Multi-select Enumeration Example

Multi-select enumerations are very similar to single select with a very important limitation. Any given multi-select enumeration in EDMX is limited to 64 potential values. The following link refers to the Multi-Valued Lookup specification in the RESO Web API. This section fully describes how multi-select enumerations are defined and provided in EDMX metadata. Samples are included for quick reference within Section 2.4.10 – Multi-Valued Lookups in the included RESO Web API Version 1.0.2 Standards Specification Documentation.

The following sample describes which fields are defined as Multi-Select lookups.

```
<Property Name="GarageParking" Type="PropertyEnums.GarageParking">
<Annotation Term="RESO.OData.Metadata.StandardName" String="GarageParking" />
</Property>
<Property Name="GarageParking_Ext_1" Type="PropertyEnums.GarageParking_Ext_1">
<Annotation Term="RESO.OData.Metadata.StandardName" String="GarageParking" />
</Property>
```

The following two samples demonstrate how a Multi-select enumeration is represented using two enumeration types declared in the EDMX metadata. Please see Section 2.4.10 – Multi-Valued Lookups in the included RESO Web API Version 1.0.2 Standards Specification Documentation.

```
<EnumType Name="GarageParking" UnderlyingType="Edm.Int64" IsFlags="true">
<Member Name="AttachedGarage" Value="1" />
<Member Name="Garage" Value="2" />
[...]
<Member Name="LeasedParking" Value="2305843009213693952" />
<Member Name="AddlSpotsAvailable" Value="4611686018427387904" />
</EnumType>

<EnumType Name="GarageParking_Ext_1" UnderlyingType="Edm.Int64" IsFlags="true">
<Member Name="SpacesPerSqFt" Value="1" />
<Member Name="ParkingLot1Block" Value="2" />
[...]
<Member Name="ElectricGate" Value="32" />
<Member Name="Storage_Airplane" Value="64" />
</EnumType>
```