

# What is it?

---

Simple Create, Update and Delete operations of PLEXOS Input data.

Nuget package:

[https://dev.azure.com/energy-exemplar/Cloud%20Platform/\\_artifacts/feed/dotnet/NuGet/EnergyExemplar.Plexos.Input.CRUD](https://dev.azure.com/energy-exemplar/Cloud%20Platform/_artifacts/feed/dotnet/NuGet/EnergyExemplar.Plexos.Input.CRUD)

The CRUD API was designed to make updating and inserting PLEXOS input data simple and reliable. Validation and an implicit transaction assures that data is saved into a valid state and consumer is made aware of any problems. Bulk operations are supported and up to 40,000 operations were tested to ensure API is performant.

What is supported:

Type:	CREATE	UPDATE	DELETE	IS_ENABLED	OTHER
CustomColumn	✓	✓	✓		
Object	✓	✓	✓		
AttributeData	✓	✓	✓		
ObjectMemo	✓	✓	✓		
Membership	✓		✓		
Data	✓	✓	✓		
Band	✓	✓	✓		
DateFrom	✓	✓	✓		
DateTo	✓	✓	✓		
DataMemo	✓	✓	✓		
Tag	✓		✓		
Text	✓	✓	✓		
Class			✓		
Collection			✓		
Attribute			✓		
Property			✓		
PropertyReport				✓	
Report	✓	✓	✓		
GeometryData	✓	✓	✓		

## Validation:

Care is taken to ensure data integrity is upheld anytime CRUD API is used. Several validation rules have been identified and are enforced, notable examples include:

System object can never be deleted. Only classes, collections, properties and attributes that IS\_ENABLED can be used. Updating these entities may be required before additional operations will succeed. Cannot change classes on objects once they are created. Property values and Attribute values are validated against Input Mask, and Validation Mask to ensure only valid options are persisted. Object class, membership to collection to class is enforced. Membership 1:1 or 1:Many relationships is enforced Validation on all text fields to guard against sql/javascript injection. All PLACEHOLDERS are verified against current operation

## Contracts:

When using API via JSON files (CLI) or REST (Studies API) the following contract is expected providing deletes, updates and adds. This is also the order that will be used when validating and saving input data.

```
{  
  "deletes": [  
    ...  
  ],  
  "updates": [  
    ...  
  ],  
  "adds": [  
    ...  
  ]  
}
```

## Deletes:

Actual database keys are required. Keys are always STRINGS to allow for PLACEHOLDERS

Cascade delete has been enabled on PlexosInputDataContext. When a Object is deleted, all memberships related to that object, and all data related to those memberships will also be deleted. This can be used to delete whole branches quickly by only specifying an entry point. Individual items can also be deleted by providing minimum identifying information:

```
{  
  "deletes": [  
    { "type": "CustomColumn", "columnId": "1", "classId": 2 },  
    { "type": "Object", "objectId": "2210" },  
    { "type": "ObjectMemo", "objectId": "2210" },  
    { "type": "Membership", "membershipId": "500" },  
  ]  
}
```

```
{
  "type": "Data", "dataId": "45" },
  { "type": "Band", "dataId": "45" },
  { "type": "DateFrom", "dataId": "45" },
  { "type": "DateTo", "dataId": "45" },
  { "type": "DataMemo", "dataId": "45" },
  { "type": "Tag", "dataId": "45", "ObjectId": "2210" }

  { "type": "Text", "dataId": "45" }
],
"updates": [
  ...
],
"adds": [
  ...
]
}
```

## Updates:

Actual database keys are required. Keys are always STRINGS to allow for PLACEHOLDERS

Updates on Memberships and Tags are not allowed.

```
{
"deletes": [
  ...
],
"updates": [
  {"type": "Class", "classId":4, "isEnabled" : true},
  {"type": "Collection", "collectionId":5, "isEnabled" : true},
  {"type": "Attribute", "attributeId":1, "isEnabled" : true},
  {"type": "Property", "propertyId":1, "isEnabled" : true},
  {"type": "PropertyReport", "propertyId":1, "isLT" : true, "isMT" : true,
  "isST" : true, "isPA" : true, "isPeriod" : true, "isQuantity" : true, "isVisible"
  : true},
  {"type": "CustomColumn", "ColumnId": "1", "ClassId": 2, "Name": "CUSTOM Renamed"},

  {"type": "Object", "ObjectId": "2210", "CategoryId": 94, "ClassId": 2, "Name": "Salt
  Lake Energy Systems (Unit4) Renamed", "Description": "Salt Lake Energy Systems
  "}
```

```
(Unit4) Renamed"},

{ "type": "ObjectMemo", "ColumnId": "1", "ObjectId": "2210", "Value": "SOME CUSTOM VALUE Renamed"},

{ "type": "Data", "DataId": "32322", "PropertyId": 38, "Value": -1 }, 

{ "type": "Band", "DataId": "32325", "BandId": 2 }, 

{ "type": "DateTo", "DataId": "32322", "Date": "2022-12-30T00:00:00"},

{ "type": "DataMemo", "DataId": "32322", "Value": "some new value"},

{ "type": "Text", "DataId": "32322", "ClassId": 73, "Value": "Load Traces\\2016 Wind Traces\\MLWF1_REF_2010.csv"}], "adds": [ ... ]}
```

## Adds:

PLACEHOLDERS can be used for additional operations requiring relationship to data in single operation:  
 PLACEHOLDERS are unique per TYPE, so Object and Membership may both have "\$1" as placeholder and is valid, but 2 Objects with same placeholder is not.

```
{
"deletes": [
  ...
],
"updates": [
  ...
],
"adds": [
  { "type": "CustomColumn", "columnId": "$1", "classId": 2, "name": "CUSTOM" },

  //objects can have attributes attached as part of the object.
  { "type": "Object", "objectId": "$1", "classid": 2, "categoryId": 94, "name": "Salt Lake Energy Systems (Unit4)", "description": "Salt Lake Energy Systems (Unit4)", "attributes": [ {"attributeId": 1, "value": 40.550}, {"attributeId": 2, "value": -111.225} ] },

  { "type": "ObjectMemo", "columnId": "$1", "objectId": "$1", "value": "SOME CUSTOM VALUE"},

  //memberships can have data attached as part of the object.
  { "type": "Membership", "membershipId": "$2", "collectionId": 1, "parentObjectId": "1", "childObjectId": "$1", "data": [ { "propertyId": 9, "value": 1}, { "propertyId": 21, "value": 2} ] },
```

```
//data can have band, dateFrom and dateTo on object. Text, tags etc may be
considered.

{ "type": "Data", "dataId": "$3", "membershipId": "$2", "propertyId": 38,
"value": -1,"dateFrom": "2022-01-01", "band": 2},

{ "type": "Band", "dataId": "$3", "bandId": 2 },

{ "type": "DateTo", "dataId": "$3", "date" : "2022-12-30"},

{ "type": "DataMemo", "dataId": "$3", "value": "=-10*1/(175)"},

{ "type": "Tag", "dataId": "$3", "objectId": "$10", "actionId": 1},

{ "type": "Text", "dataId": "$3", "classId": 73, "value": "Load Traces\\2016
Wind Traces\\MLWF1_REF_2010.csv"}
]
}
```

## Configuration:

All required types can be configured by calling extension methods in API. PlexosInputDbContext is required and should be configured by consumer.

```
public void ConfigureServices(IServiceCollection services)
{
    ...

    services.AddDbContext<PlexosInputDbContext>(d =>
    {
        CustomSqliteConnectionStringBuilder builder = new
        CustomSqliteConnectionStringBuilder { DataSource = dbFilePath };
        d.UseSqlite(builder.ConnectionString, options => {
            options.UseNetTopologySuite(); });

        d.UseQueryTrackingBehavior(QueryTrackingBehavior.TrackAll);
        d.EnableDetailedErrors();
        d.EnableSensitiveDataLogging();
    })
    .AddCrudAPI(); // registers CRUD API

    ...
}
```

## Caching:

In order for validation against data to stay performant we may have to cache some data. Currently this cache is configured per call against API, but a longer lived cache with refresh capability may be required:

To register your own managed cache you can simply register a scoped instance of DataCache and let dependency injection handle the rest.

