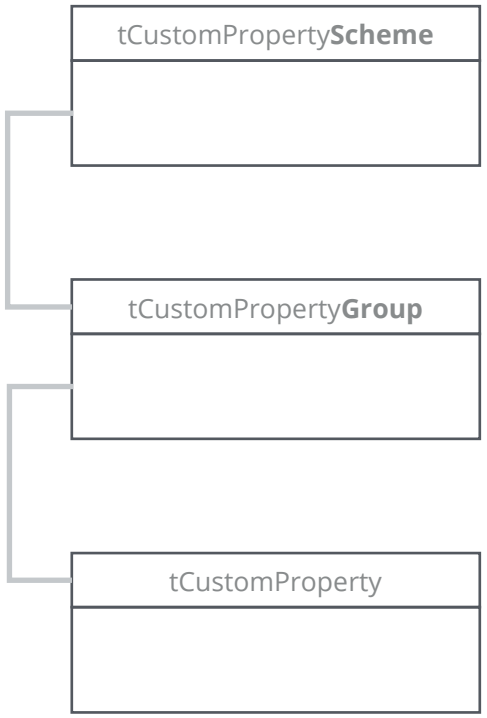
Custom Properties

Configuration Hierarchy



Scheme

A collection of Custom Properties that can be applied to one or more Entities.

Example

A collection of properties applied to a System called Packaging Line.

Group

A simple grouping mechanism for organizing large numbers of Custom Properties within a Scheme.

Example

Properties separates into two groups, Front of Line and End of Line.

Property

A single attribute definition that can hold entity specific values.

Property

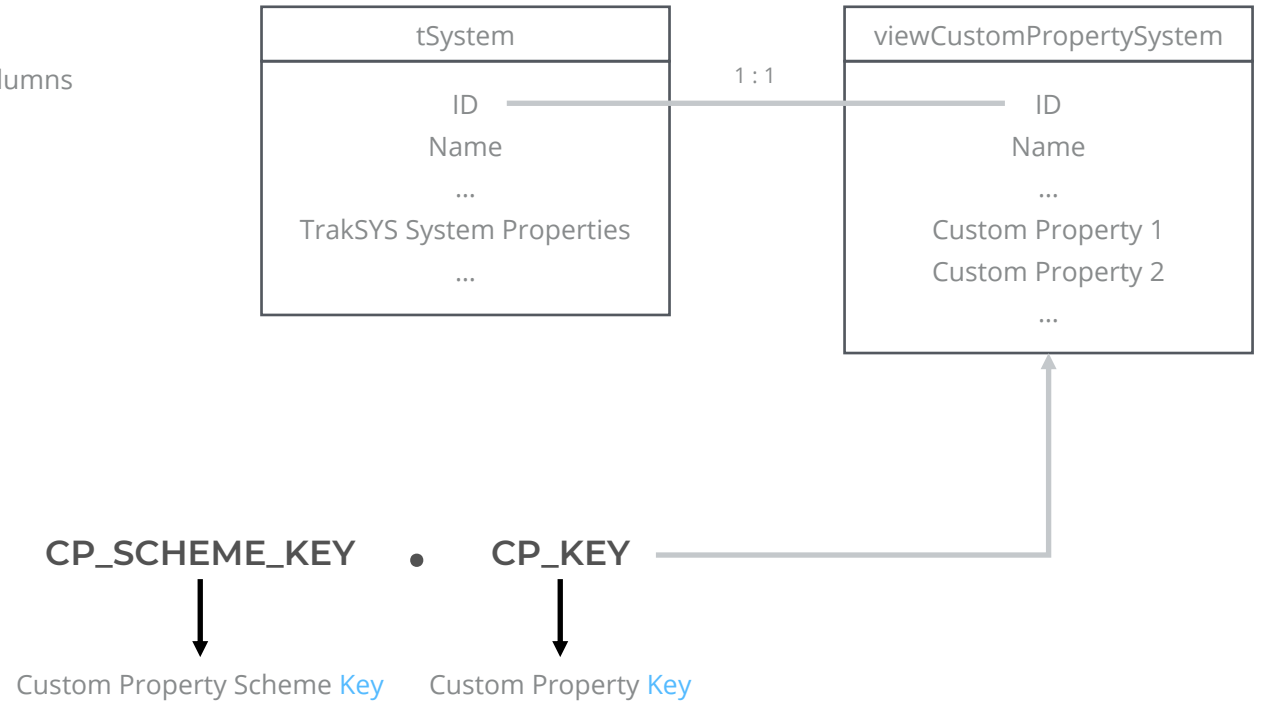
End of Line properties Packaging Mode, Allow Dual Packing, and Require Labeling.

Views

Views

- Extends the configuration tables with Custom Property Columns
- Built-In and part of the standard Schema
- Automatically updated as Properties are Managed

- viewCustomPropertyArea
- viewCustomPropertyDevice
- viewCustomPropertyEventCategory
- viewCustomPropertyEvenCode
- ...

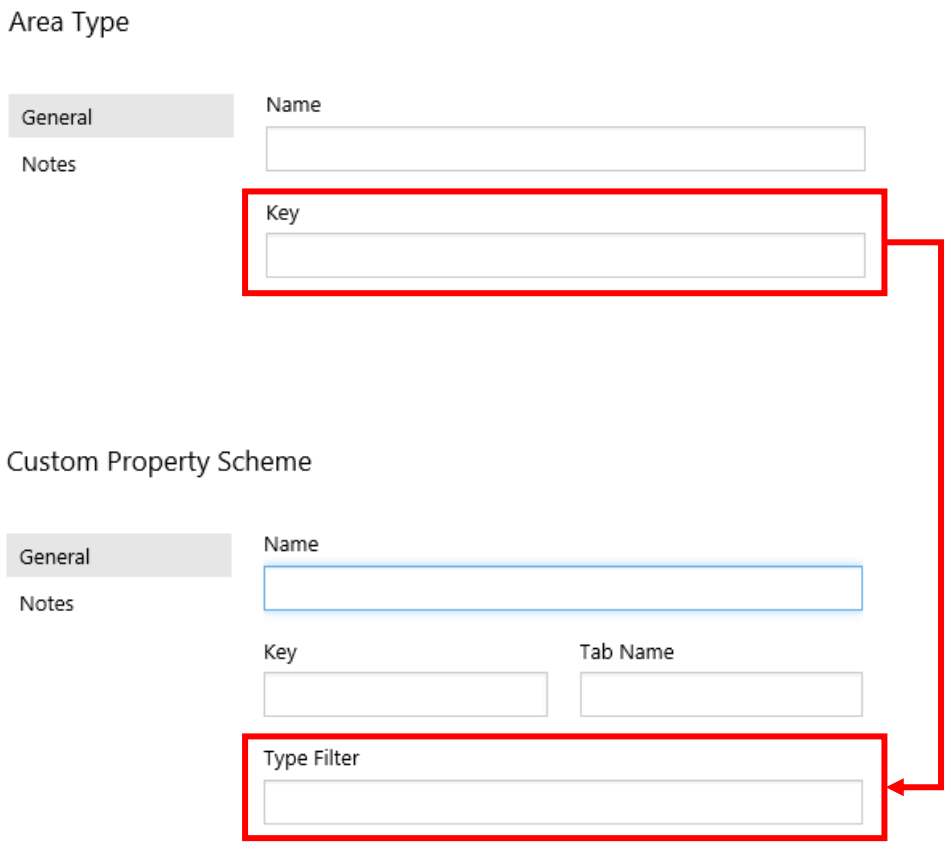


Type Specific Schemes

Entities supporting Type

The following entities may have Custom Property Schemes that are only displayed for specific Type values...

- Area
- Event Category
- Event Code
- Event Definition
- Function Definition
- Journal Category
- KPI Calculation
- KPI Counter
- Location
- Parameter Definition
- Product
- Recipe Step Definition
- Recipe
- Sample Definition
- Step Category
- Step Code
- System
- Task Definition
- Transfer Definition
- User



API Techniques

Note that error handling is omitted for brevity and clarity.

Loading Single Entity

When loading a single entity, the Custom Property values are automatically loaded.

```
var ed = this.Ets.Api.Data.DbEventDefinition.Load.ByID(38);
bool cpValue = ed.CustomProperties.GetAsBool("PD_SCHEME_KEY.PD_KEY", false);
```

Loading List of Entities

When loading a list of entities, the Custom Property values are NOT automatically loaded and must be done for each entity.

```
var eds = this.Ets.Api.Data.DbEventDefinition.GetList.ForSystemID(4);
foreach (var ed in eds)
{
  this.Ets.Api.Data.CustomProperty.LoadModel(ed);
  bool cpValue = ed.CustomProperties.GetAsBool("PD_SCHEME_KEY.PD_KEY", false);
}
```

API Techniques

Note that error handling is omitted for brevity and clarity.

Updating Custom Properties

A special merge command must be included when updating custom property values.

```
var uow = this.Ets.Api.CreateUnitOfWork();

var ed = this.Ets.Api.Data.DbEventDefinition.Load.ByID(38);
ed.CustomProperties["PD_SCHEME_KEY.PD_KEY"].Value = "new value";

this.Ets.Api.Data.CustomProperty.MergeCustomProperties(ed.CustomProperties, ed.ID, "EventDefinition", uow);
this.Ets.Api.Data.DbEventDefinition.Save.UpdateExisting(ed, uow);
```