CslaExtension Quick Reference

Contents

[Description 1](#_Toc260644069)

[Installation 1](#_Toc260644070)

[Uninstall 2](#_Toc260644071)

[CslaExtension Custom Properties 2](#_Toc260644072)

[Conceptual Entity Model (designer) custom properties 2](#_Toc260644073)

[Entity custom properties 3](#_Toc260644074)

[Entity Property custom properties 4](#_Toc260644075)

[CslaExtension template 5](#_Toc260644076)

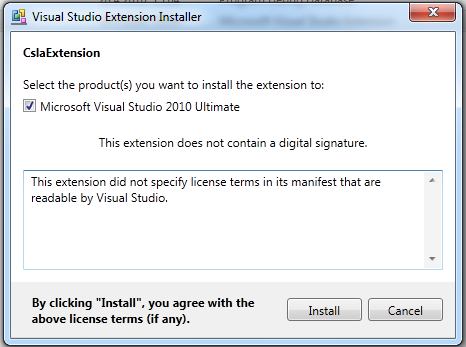
# Description

CslaExtension for Visual Studio is an extension for generation of CSLA objects from an ADO.NET Entity Framework Model. CslaExtension adds custom properties to the EF model, and a custom template for generation of CSLA classes.

Custom properties are used to define CSLA classes that will be generated from the EF model, property generation rules, CSLA business rules, relations and so on. T4 template which is installed with CslaExtension is then used to generate CSLA classes using values from custom properties values.

# Installation

Install extension using provided VSIX file (or download from Visual Studio Gallery http://visualstudiogallery.msdn.microsoft.com) Double click VSIX file:



Click Install and you are done. Now open Visual Studio and go to [Tools -> Extension Manager] and you should see CslaExtension on the list of installed extensions.

# Uninstall

Open Visual Studio and go to [Tools->Extension Manager], find CslaExtension and click Uninstall.

# CslaExtension Custom Properties

Custom properties are added to the following objects in the EF model:

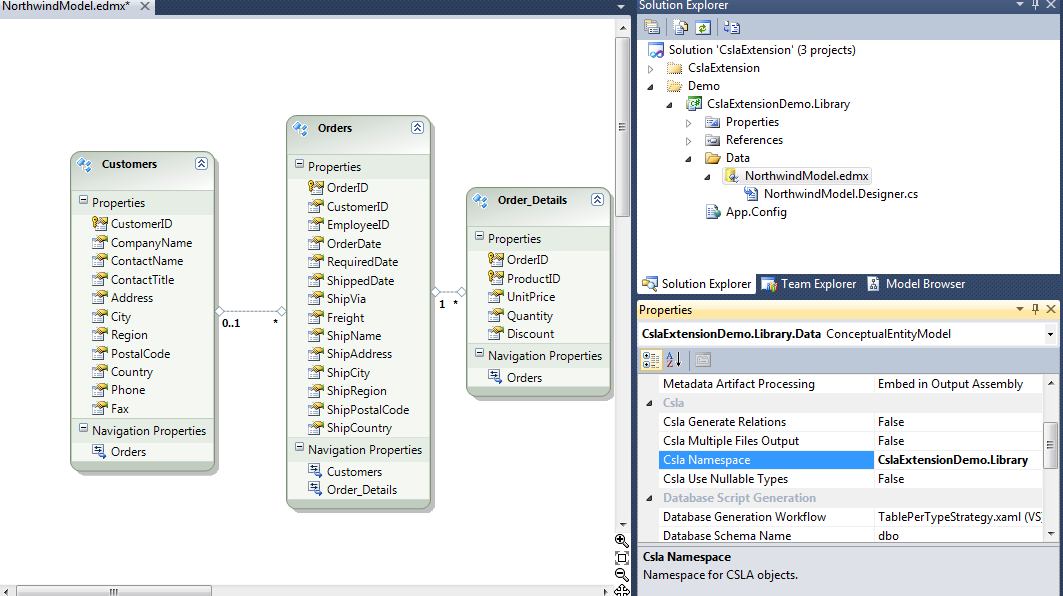
* Conceptual Entity Model (designer)
* Entity
* Entity property
* Entity navigation property

CslaExtension custom properties are grouped in Csla region, and begin with Csla keyword.

*Note: to activate properties window select desired object and press F4.*

### Conceptual Entity Model (designer) custom properties

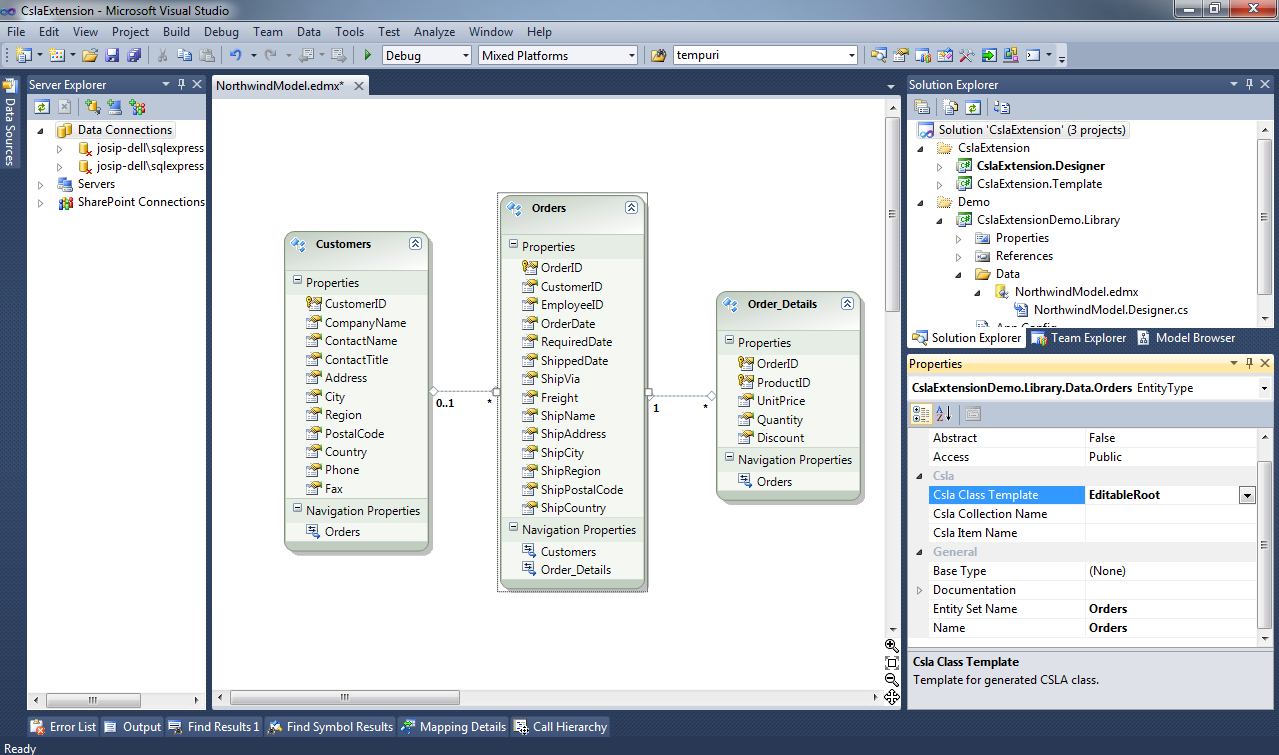
When empty designer area is selected, Conceptual Entity Model properties are shown.



* **Csla Generate Relations:** When true, navigation (relation) properties will be generated by default. You can override this default behavior on Navigation Property using their Csla Setter and Csla Getter properties. Navigation properties can be single or list properties. In example above, Order can have one Customer and many Order\_Details.
* **Csla Multiple Files Output:**
  + **AllClassesInSingleFile** – As name says it generates all classes in single file. This is default behavior.
  + **EachClassInSeparateFile** – It makes separate file for each generated class.
  + **CommonAndServerCodeForEachClass** – Used mainly for Silverlight development. For each class generated code is split in two files: common and server-only code (this file has suffix “Server”; for example “Order.Server.cs”). File with common code should be linked in Silverlight only project.
* **CslaNamespace:** Namespace where CSLA classes will be generated. If empty, code will be generated in current file location default namespace. This can be a problem if generated CSLA classes and EF objects are in the same namespace. It is recommended to edit this property and set namespace for generated CSLA objects.
* **Csla Use Nullable Types:** If true, for all data types, nullable type will be used where possible. This is useful in scenarios where you want to create new object and don't want to set default values (e.g. what is default for Birthdates?). In this case, nullable type will be used for generated property (e.g. DateTime?, int?...), but generator will also generate BusinessRule PropertyRequired so object can't be saved to database until this property has value.

### Entity custom properties

Following properties are added to a single Entity:

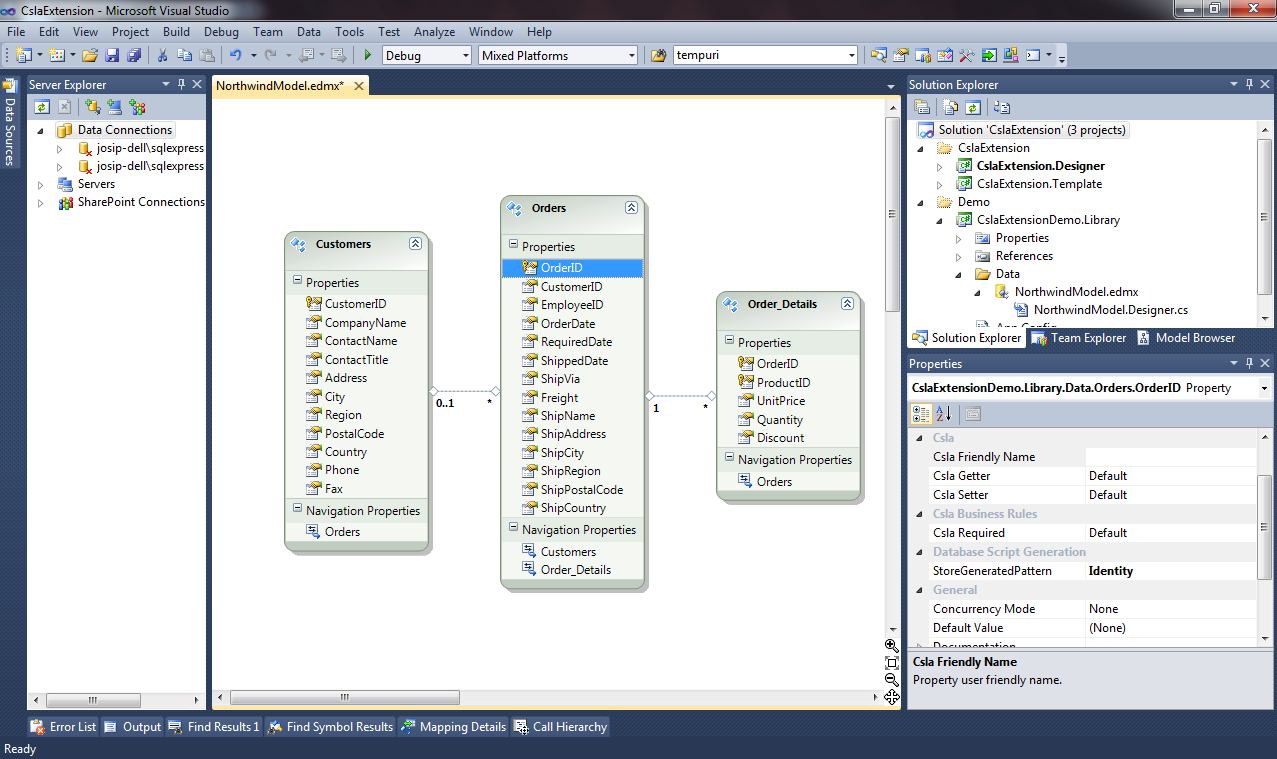


* **Csla Class Template:** Template that will be used for CSLA class generation. Available templates are: Editable Root, EditableRootList, EditableChild, EditableChildList, ReadOnlyRoot, ReadOnlyRootList, ReadOnlyChild and ReadOnlyChildList.

If selected template is List (e.g. EditableRootList) then two classes will be generated; one for collection of items, and one that represents single item.

* **Csla Collection Name:** Name of the generated collection class. If empty, default collection name will be the same as the EF Entity Name, suffixed with “List” (e.g. OrdersList, CustomersList)
* **Csla Item Name:** Name of the generated class that represents single item. If empty, CSLA class name will be same as the EF Entity name (e.g. Orders, Customers).

### Entity Property custom properties



When entity property is selected (e.g. OrderID) following custom properties are available:

* **Csla Friendly Name:** Friendly name for the property. Used in CSLA business rules or in user interface elements.
* **Csla Getter:** Property Getter modifier. By default modifier is the same as for the property in EF model (e.g. if property in EF model is internal, property generated in CSLA class will also be internal). Available values are: Default, Private, Public, Protected, Internal, None. If None is selected, getter will not be generated.
* **Csla Setter:** Property Setter modifier. Same as for the Getter.
* **Csla Required:** If “Required” Business Rule will be generated for selected property. By default, business rule will be generated depending on value of Nullable property in EF model. This is especially useful when nullable types are used for all generated properties (see ConceptualEntity: Csla Use Nullable Types). Available values are: Default, Yes, No.

# CslaExtension template

CslaExtension template is a T4 template that generates CSLA classes from ADO.NET EntityFramework model. When CslaExtension is installed, you get custom properties on EF model, and a custom item template for generation of CSLA classes. Custom template is available from VS project’s “Add new item..” menu, or when you right click on empty area of EF designer and choose “Add code generation item..”.