

2012



## GETTING STARTED WITH CODE ON TIME

Got data? Generate modern web apps in minutes. Learn to create sophisticated web apps with Code On Time application generator for ASP.NET, Azure, DotNetNuke, and SharePoint.

## Contents

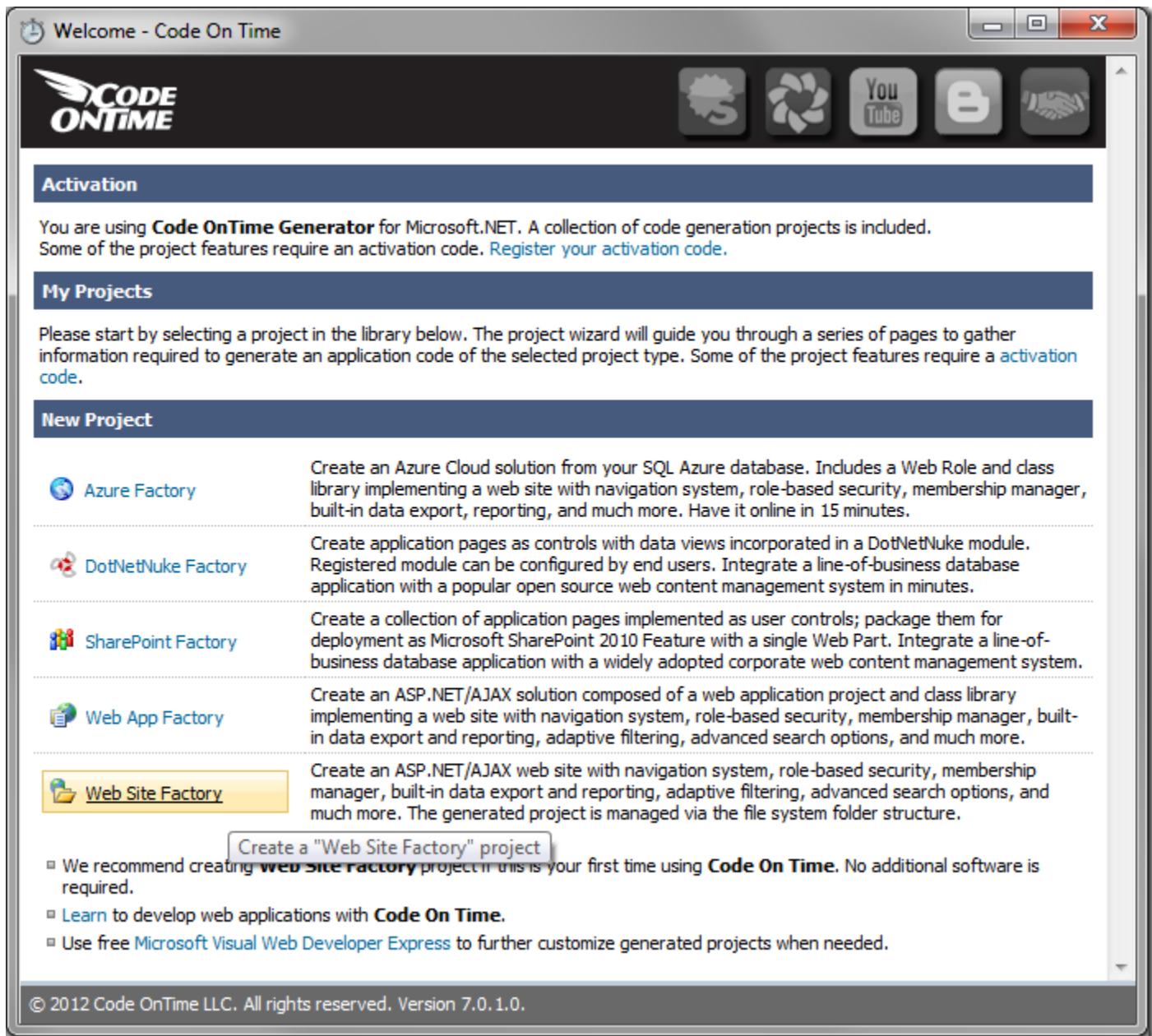
Creating an App.....	3
Using the Project Designer.....	12
Project Browser .....	13
Project Explorer.....	15
Navigate To .....	17
Creating a Three-Level Master-Detail Page .....	19
Creating the Page .....	19
Establishing Master-Detail Relationships .....	22
Rearranging Data Fields .....	28
Moving .....	28
Deleting.....	28
Adding.....	29
Designing Forms.....	32
Floating Category Data Fields.....	32
Configuring a Multi-Column Form.....	34
Custom Category Template .....	38
Configuring Charts .....	42
Creating the View .....	42
Viewing the Results .....	44
Creating Many-to-Many Fields .....	48
Changing the Child Views.....	54
Creating Calculated Fields.....	57
Creating the virtual Field.....	57
Updating the Custom Category Template .....	60
SQL Formula .....	62
Calculating Field Values Just-in-Time .....	63
SQL Business Rule.....	63
Code Business Rule .....	65
JavaScript Business Rule .....	68
Implementing a Custom Action .....	70
Adding a Custom Action .....	70
Creating the Confirmation Controller.....	71
Creating an SQL Business Rule to Handle the Action.....	75



## Creating an App

Let's create a new *Web Site Factory* project.

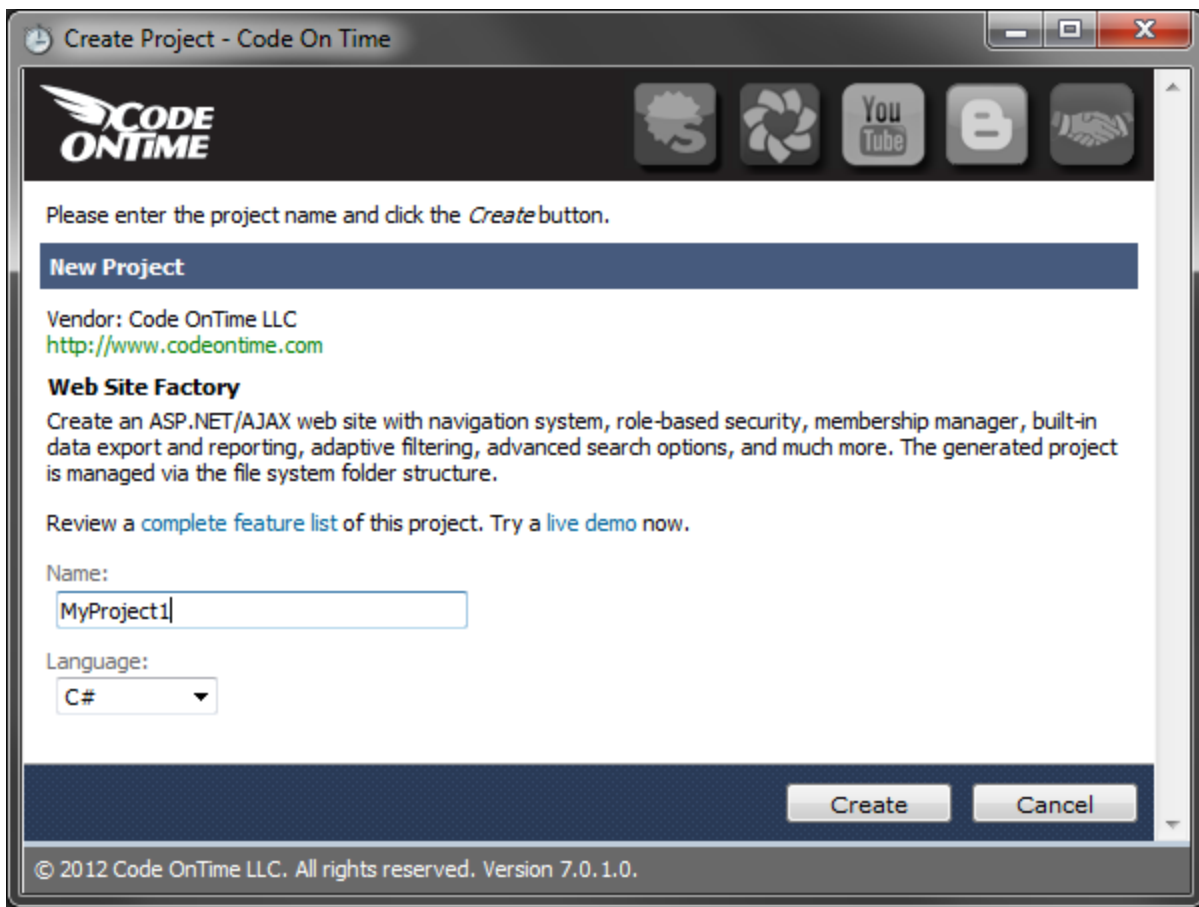
Start [Code On Time](#) web application generator and select *Web Site Factory* under *New Project*.



The next page will prompt to specify a name and programming language of the project.

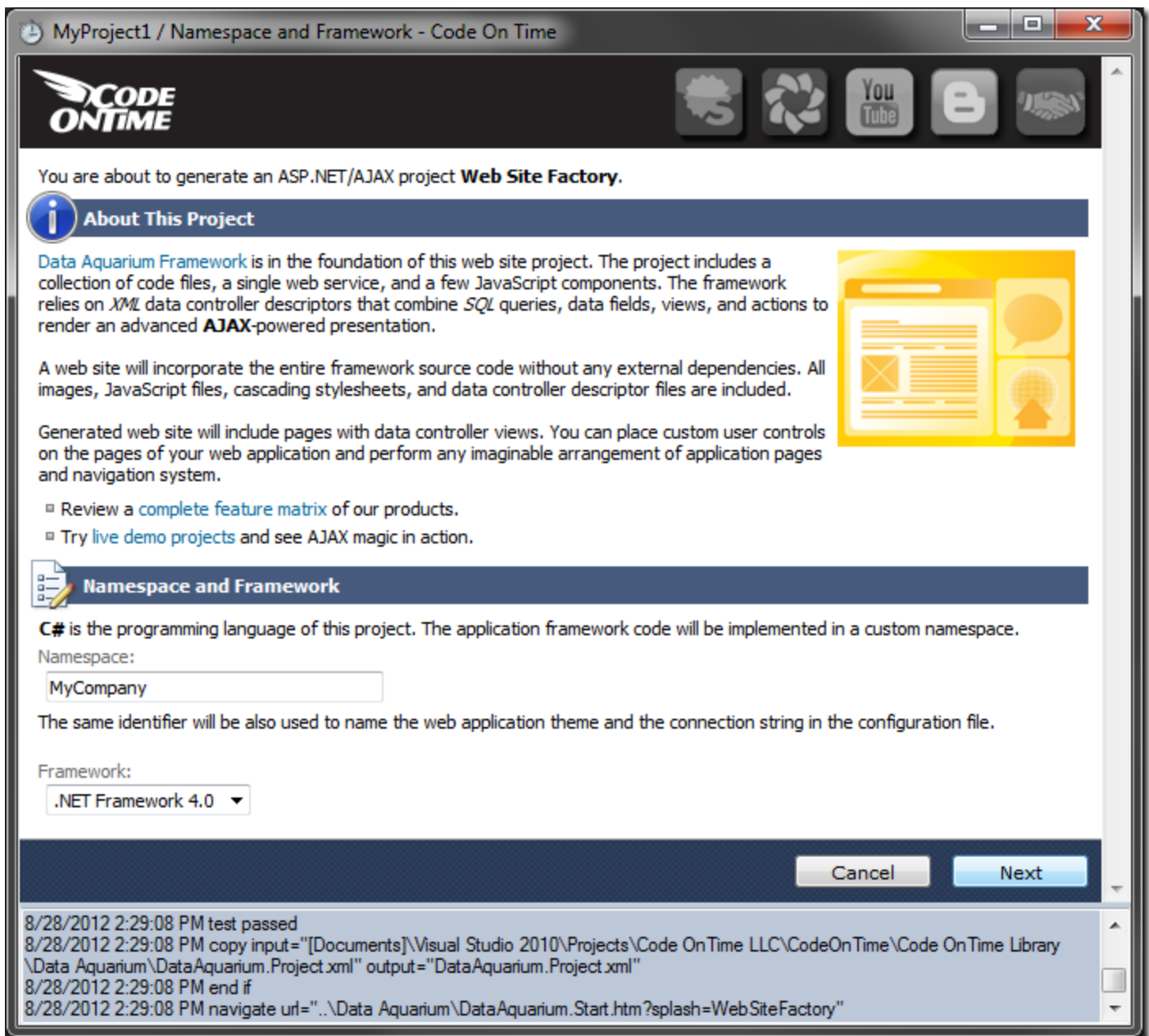
Select the programming language of your choice and specify the following name:

Property	Value
Name	MyProject1



Press *Create* button to create the project.

The next page allows changing the *Namespace* and the version of *Microsoft .NET Framework*. Leave the default values and press *Next*.



This page allows specifying the *Data Provider* and *Connection String*. Select your data provider from the list.

This tutorial uses *Microsoft SQL Server* database engine. If you do not have a database server, consider installing [Microsoft SQL Server 2012 Express](#).

Click on the button to the right of the *Connection String* field to open the *Connection String Configuration* page.

Please specify a database connection string for this project.

### Database Connection

A valid connection string compatible with the selected data provider is required to generate the project. Data controller descriptors are automatically built from your database and are easy to maintain with the provided XML schemas.

Data Provider:

.Net Framework Provider for SqlServer (System.Data.SqlClient)

Connection String:

...

**All database tables and views are included in this project.** [Change](#)

Configure database connection string and install database features.

If you have *SQL Server Express* edition, specify the following:

Property	Value
Server	.\SQLEXPRESS
Database	Northwind
Sample Tables	Northwind

To the right of *Database* field, click on the *Create* button and confirm to create the database.

### SQL Server Connection

Server:

.\SQLEXPRESS

☒ Windows authentication

☐ SQL Server authentication

User name:

Password:

Database:

Northwind

Create

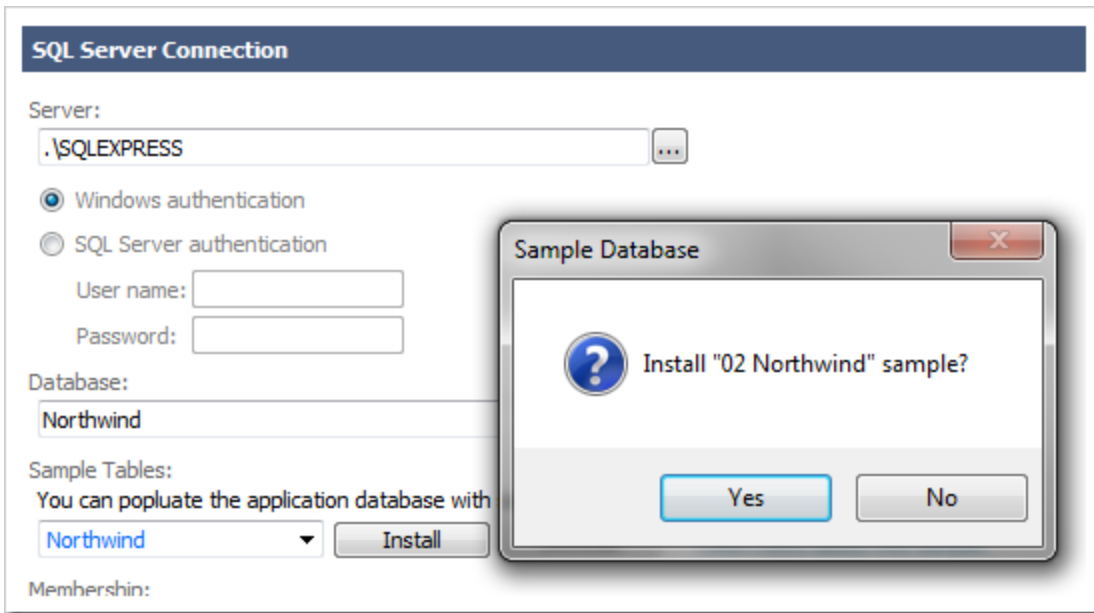
Confirm

?

Create database 'Northwind' on the specified database server?

Yes No

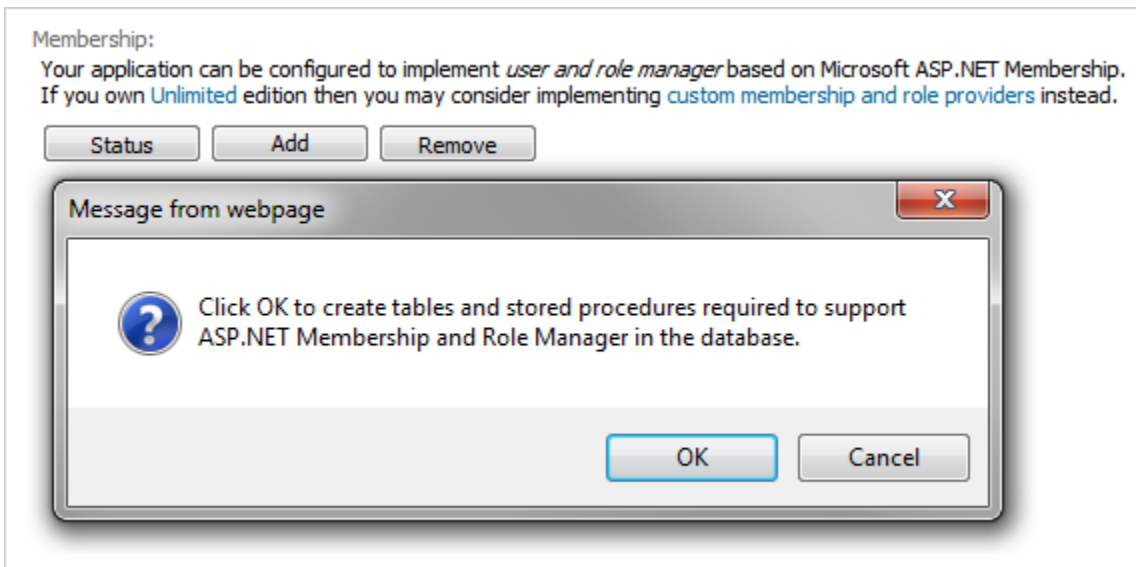
To the right of the *Sample Tables* dropdown, press *Install* and confirm to populate the project database with the *Northwind* dataset.



*Code On Time* does not offer tools to design databases. Use your favorite database management tools, such as *Microsoft SQL Server Management Studio*, to work with the database schema.

Our application will have built-in user and role management system. *Code On Time* applications rely on the security infrastructure available in ASP.NET.

Under *Membership* section, press *Add* and confirm to add *ASP.NET Membership* to the database.



You will see a confirmation when the membership has been configured.

Press *OK* to finish configuration of the database connection string.

MyProject1 / SQL Server Connection - Code On Time

**CODE ON TIME**

Please specify SQL Server name and database.

### SQL Server Connection

Server:

☒ Windows authentication  
☐ SQL Server authentication

User name:   
Password:

Database:

Sample Tables:  
You can populate the application database with sample tables.

[Learn more about this sample.](#)

Membership:  
Your application can be configured to implement *user and role manager* based on Microsoft ASP.NET Membership.  
If you own [Unlimited](#) edition then you may consider implementing [custom membership and role providers](#) instead.

8/31/2012 11:58:34 AM =====

If you are using a trial version of the application generator, the project size is limited to 10 tables or views. To select a subset of tables for the project, click the *Change* button next to "All database tables and views are included in this project". Include the following tables: *Categories*, *Customers*, *Employees*, *EmployeeTerritories*, *Order Details*, *Orders*, *Products*, *Shippers*, *Suppliers*, and *Territories*.

Please specify a database connection string for this project.

### Database Connection

A valid connection string compatible with the selected data provider is required to generate the project, with the provided XML schemas.

Data Provider:

Connection String:

**All database tables and views are included in this project.** [Change](#)

Database Object Filter

Tables / Views 10 objects selected

- ☐ dbo.Alphabetical list of products
- ☒ dbo.Categories
- ☐ dbo.Category Sales for 1997
- ☐ dbo.Current Product List
- ☐ dbo.Customer and Suppliers by City
- ☐ dbo.CustomerCustomerDemo

Press *Next* twice to reach the *Reporting* configuration page. Enable dynamic and static reports in the application.

Please specify your reporting preferences.

### Reporting

Support for [Microsoft Reporting Services](#) technology can be embedded into your application. Dynamic reports are automatically created on-the-fly without a need for a standalone reporting server. Static reports are supported as well.

Application users will be able to render reports as *Adobe PDF*, *Microsoft Office Excel*, or *Tiff* image file.

The Microsoft report viewer is required to run the reports. The *Microsoft Report Viewer Redistributable Package* includes Windows Forms and ASP.NET Web server controls for viewing reports designed using Microsoft reporting technology.

- [Download the redistributable report viewer package for Microsoft.NET 3.5.](#)
- [Download the redistributable report viewer package for Microsoft.NET 4.0.](#)

**Attention:**  
 If you have Microsoft Visual Studio 2008/2010 installed on your computer then the viewer is likely installed already. Users of Visual Web Developer Express 2008/2010 will have to install the report viewer if reports are enabled for this project.

You have to install the viewer on the server machine when the generated application is deployed.

Reports:  
☒

Hold down *Shift* key and press *Next* – this will skip directly to the *Summary* page. Press the *Generate* button to start creation of the web application.

The analysis of your database has been completed. Application generation will start immediately when you click *Generate* button.



## Summary

The following data controllers are included in the application baseline.

Categories  
Customers  
Employees  
OrderDetails  
Orders  
Products  
Shippers  
Suppliers



### Design

Open project designer to customize the app. Customization will be preserved between code generation sessions.



### Settings

Change settings of the project. Some changes will effect only new pages and refreshed data controllers.



### Refresh

Refresh database tables and views included in the project if the physical database structure has changed.



### Generate

Generate and compile the web application source code. The generated app will start in the default web browser.

When generation is complete, the web site will open in your default browser.

http://localhost:43526/Pages/Home.aspx

Start

# MyCompany

Home

Home > Home

## Start

Please select a page link in the table of contents below.

### Site Map

Home

### Instructions

Two standard user accounts are automatically created when application is initialized if membership option has been selected for this application.

The administrative account **admin** is authorized to access all areas of the web site and membership manager. The standard **user** account is allowed to access all areas of the web site with the exception of membership manager.

Move the mouse pointer over the link *Login to this web site* on the right-hand side at the top of the page and sign in with one of the accounts listed below.

Administrative account:  
**admin / admin123%**

Standard user account:  
**user / user123%**

Login to this website

User Name:

Password:

☐ Remember me next time

[Forgot your password?](#)

[Sign up now](#)

[Login](#)

© 2012 MyCompany. All rights reserved.


## Using the Project Designer

The [Project Designer](#) allows customizing the pages and data controllers of the web application.

To activate the *Project Designer*, click on a project name on the start page of the application generator.


**My Projects**


This is the list of the projects that you have created. Click on the project name to open and generate the project one more time. Note that most projects will automatically backup the previous version of the source code prior to generating and overwriting the existing files.


#	Project Name	Type	Last Modified
1.	 <b>MyProject1</b>	Web Site Factory	8/30/2012 3:02 PM


- [Create new](#) Select "MyProject1" project.
- [Learn](#) to develop web applications.
- [Request](#) help with your project.
- [Buy](#) a product edition now.
- [Order](#) our consulting services online.
- [Subscribe](#) to our newsletter.


Then, click on the *Design* button.


**Design**  
Open in Project Designer to customize the web site.


**Develop**  
Open in Visual Studio to customize the source code.


**Settings**  
View and change the settings of the web site.

**Browse**  
View the site in your default web browser.

**Refresh**  
Refresh database tables and views included in the project.

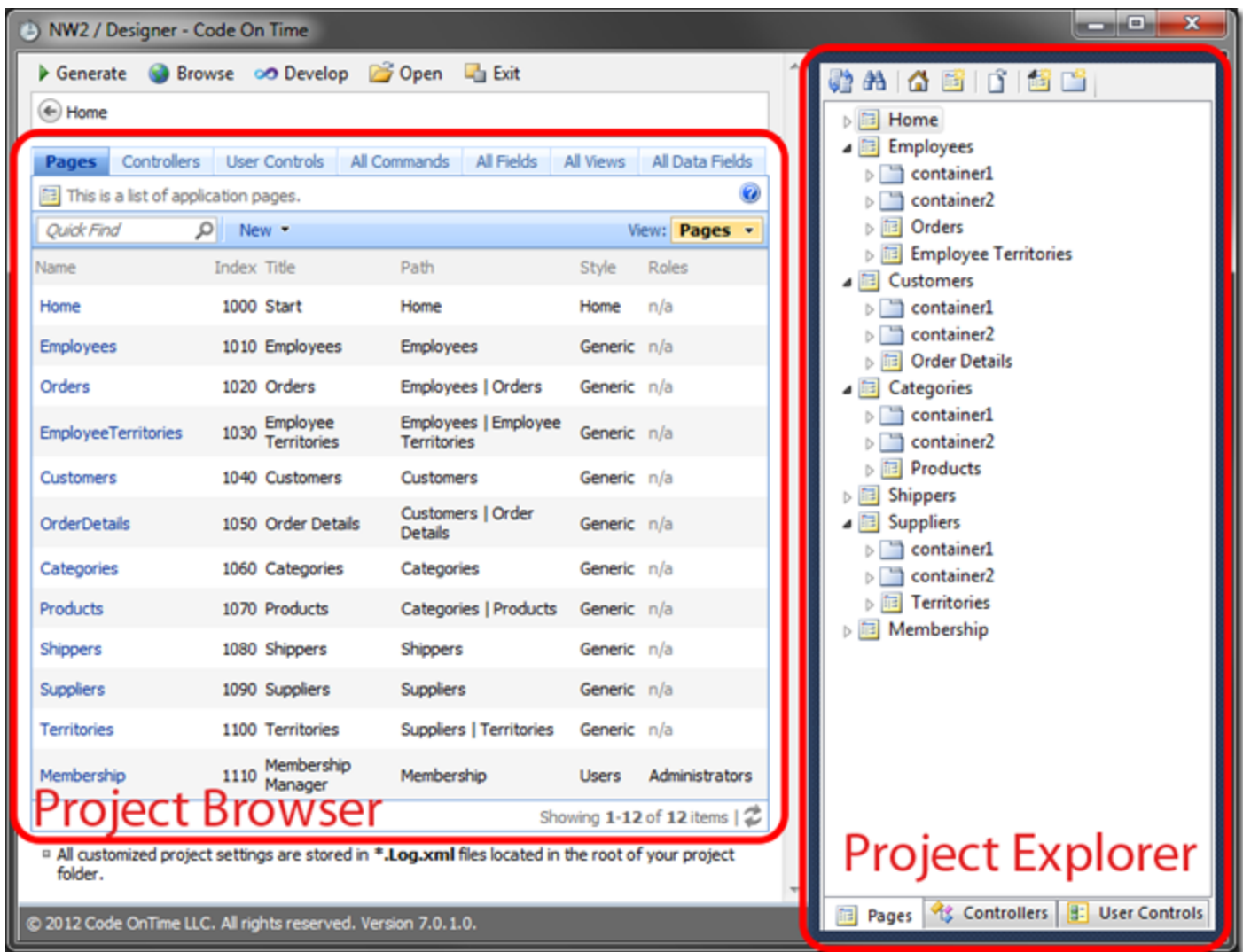
**Publish**  
Publish the web site for deployment.

**Open**  
Open in Windows Explorer to view the project files.

**Delete**  
Delete the entire site from your computer.

The *Project Designer* will open.

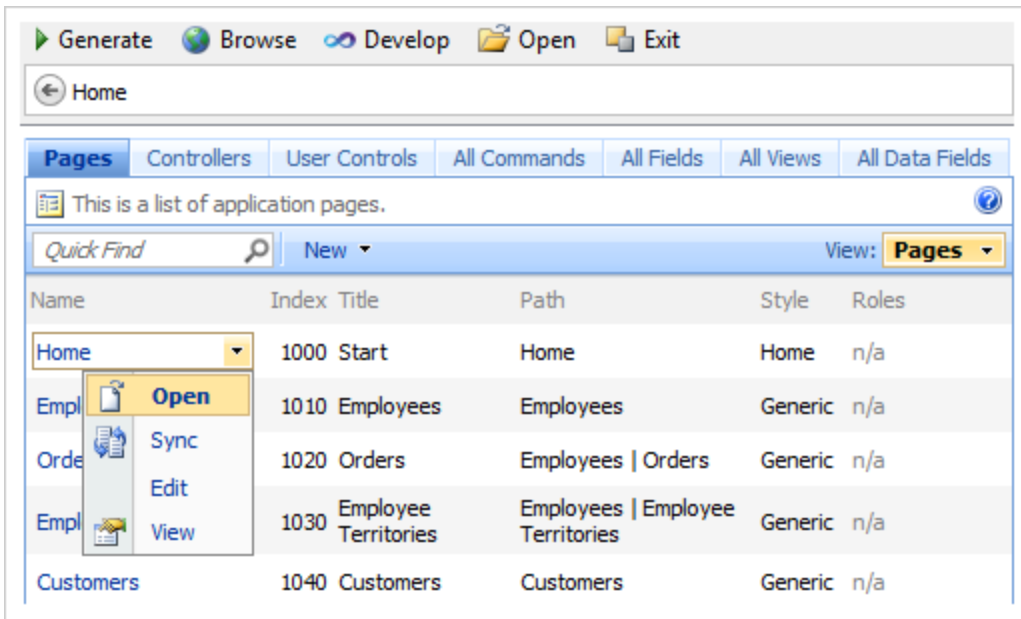
On the left side of the screen is the *Project Browser*. The right side contains the *Project Explorer*.



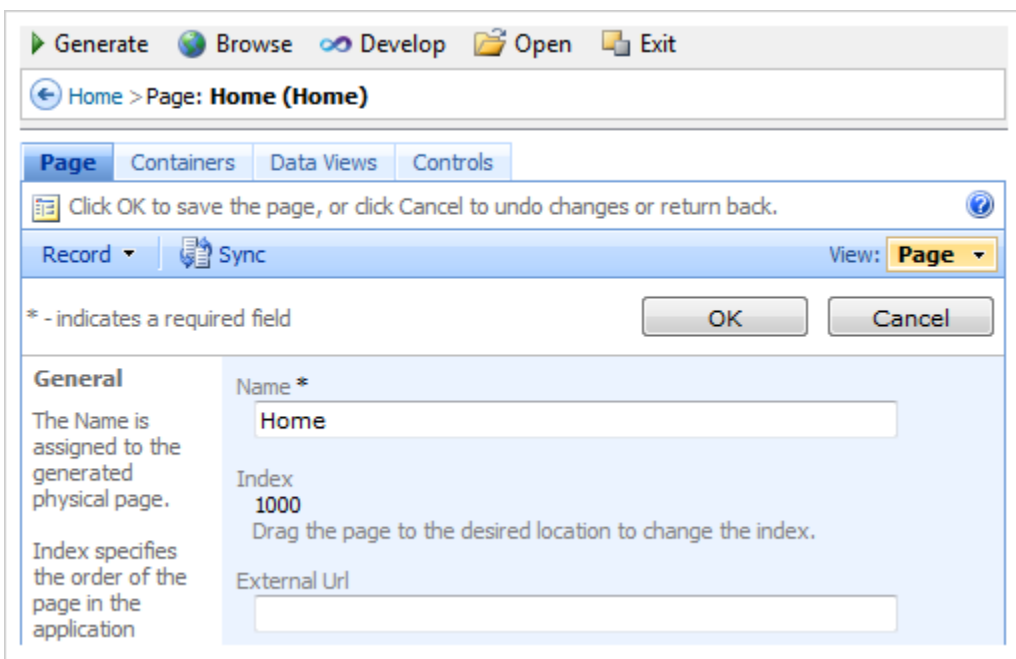
## Project Browser

The *Project Browser* allows navigating the project configuration elements. The tabs display lists of [pages](#), [data controllers](#), [user controls](#), [commands](#), [fields](#), [views](#), and [data fields](#).

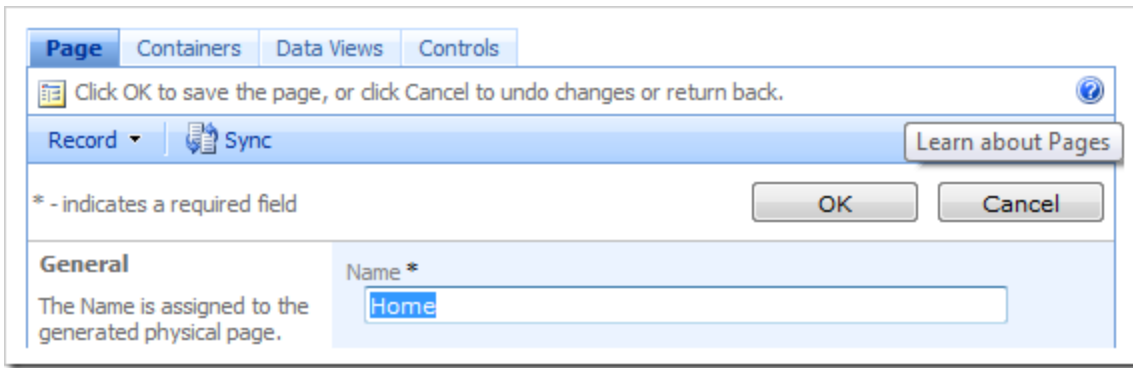
Navigate to the properties page of an element by clicking on the link in the first column or using the context menu option *Open*.



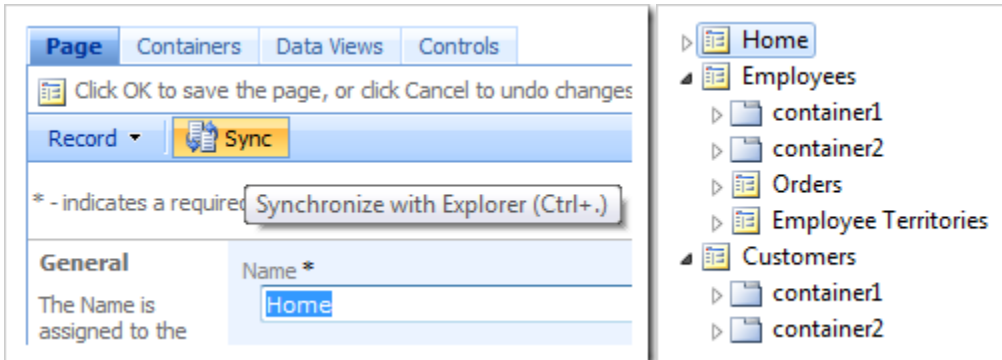
The properties page of the project configuration element will open. The breadcrumbs in the path area above the *Project Browser* show the location of the object. The tabs at the top of the page will change to reflect project configuration elements that belong to the currently selected element.



Quickly access the [Learning System](#) article for the selected project configuration element by clicking on the question mark icon in the top right corner.

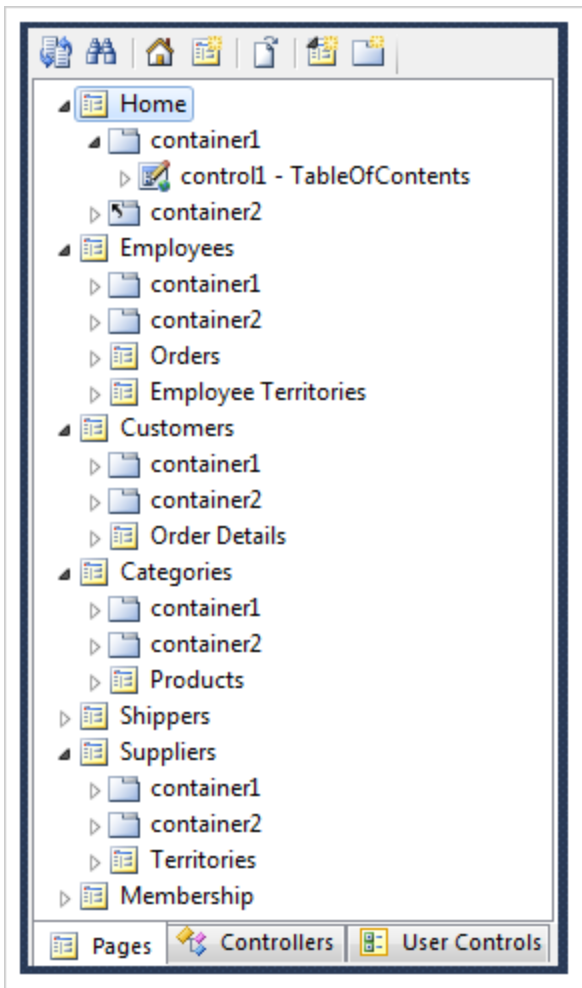


The *Project Explorer* can be synchronized with the current *Project Browser* element by using the [Sync](#) command on the action bar or pressing *Ctrl+Period* keyboard shortcut ("Ctrl" key and "." key).

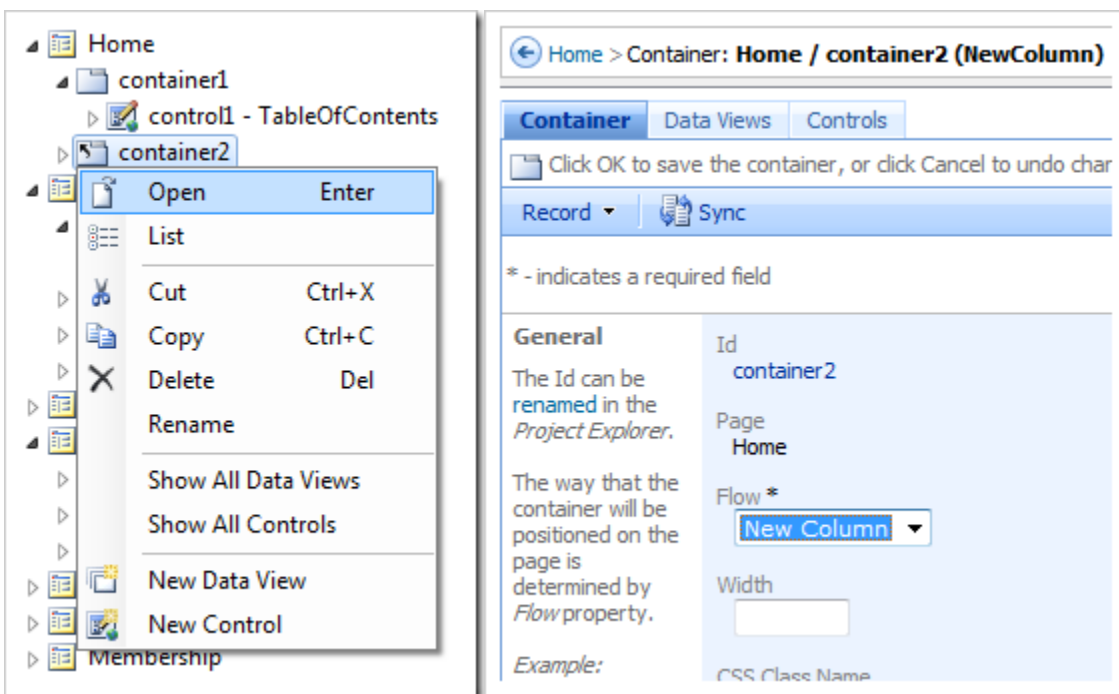


## Project Explorer

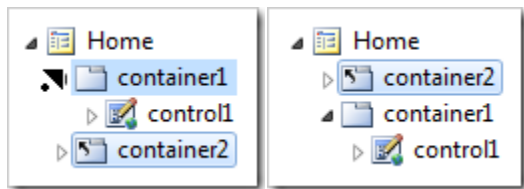
The *Project Explorer* displays a logical hierarchy of project configuration elements separated into three tabs: *Pages*, *Controllers*, and *User Controls*.



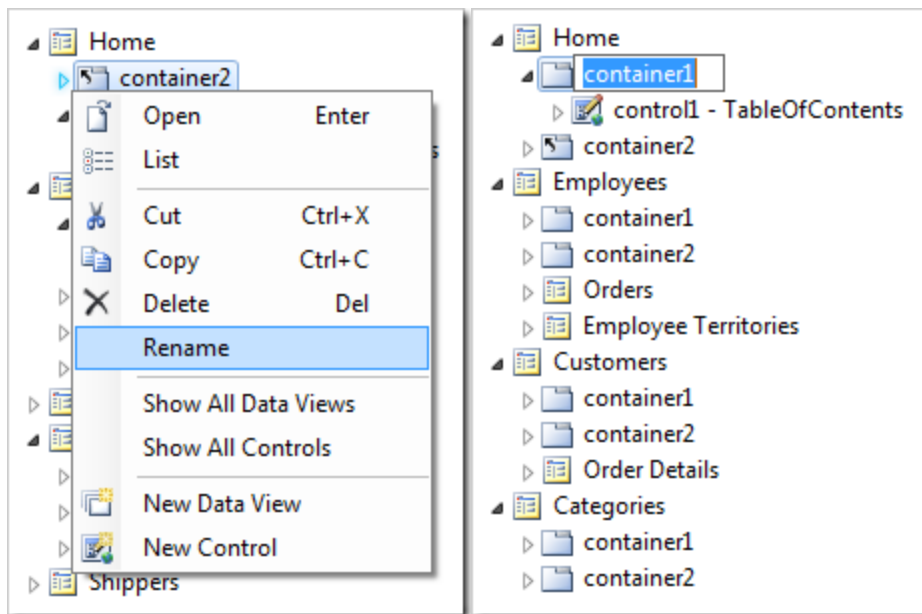
Double-click on a node in the *Project Explorer* or use the *Open* context menu option to display the properties page for the relevant element.



Drag project elements to change the application configuration. As an alternative, you can use Cut/Copy/Paste options in the context menu of project configuration element nodes. These operations can be performed on multiple selected elements of the same type.



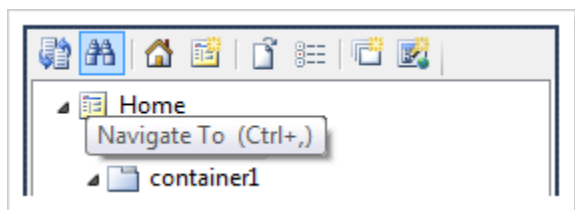
Node names of project configuration elements can be changed using the *Rename* context menu option.



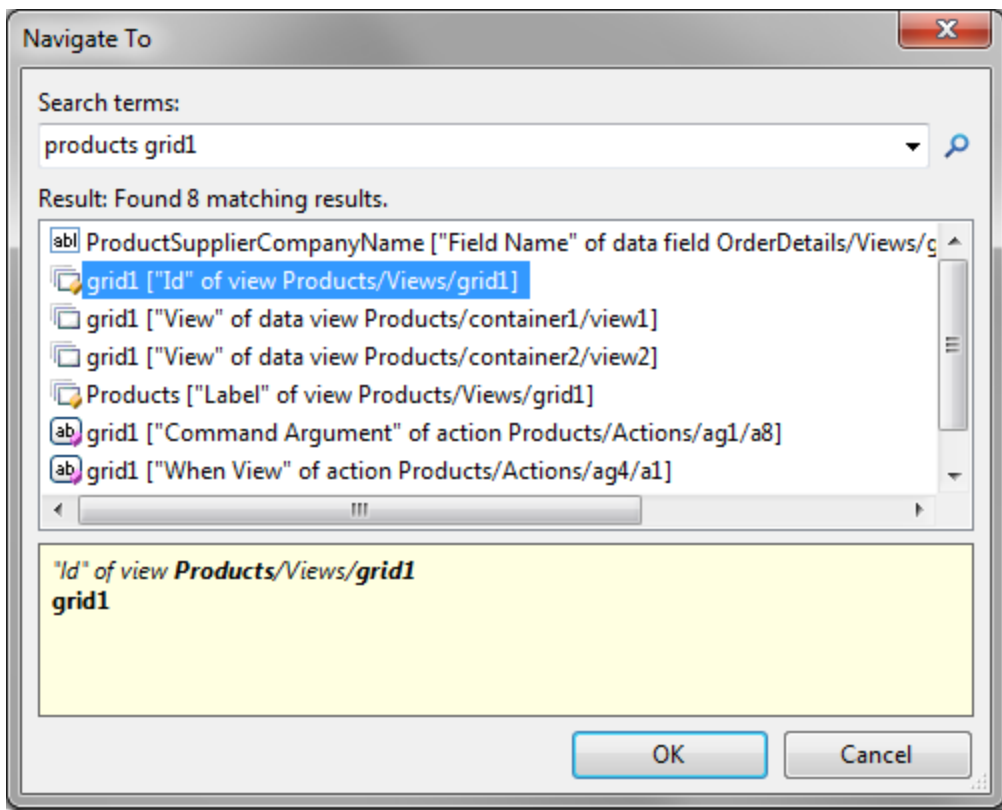
## Navigate To

The [Navigate To](#) window will search for project configuration elements by property values as soon as you type a sample search criteria.

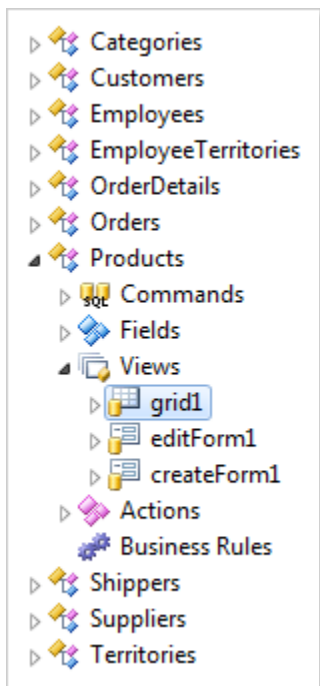
The window can be activated by clicking on the *Navigate To* icon on the *Project Explorer* toolbar or pressing the *Ctrl+Comma* keyboard shortcut ("Ctrl" key and "," key).



The *Navigate To* window will open. Start typing search terms. Matching entries will appear as soon as you stop typing.

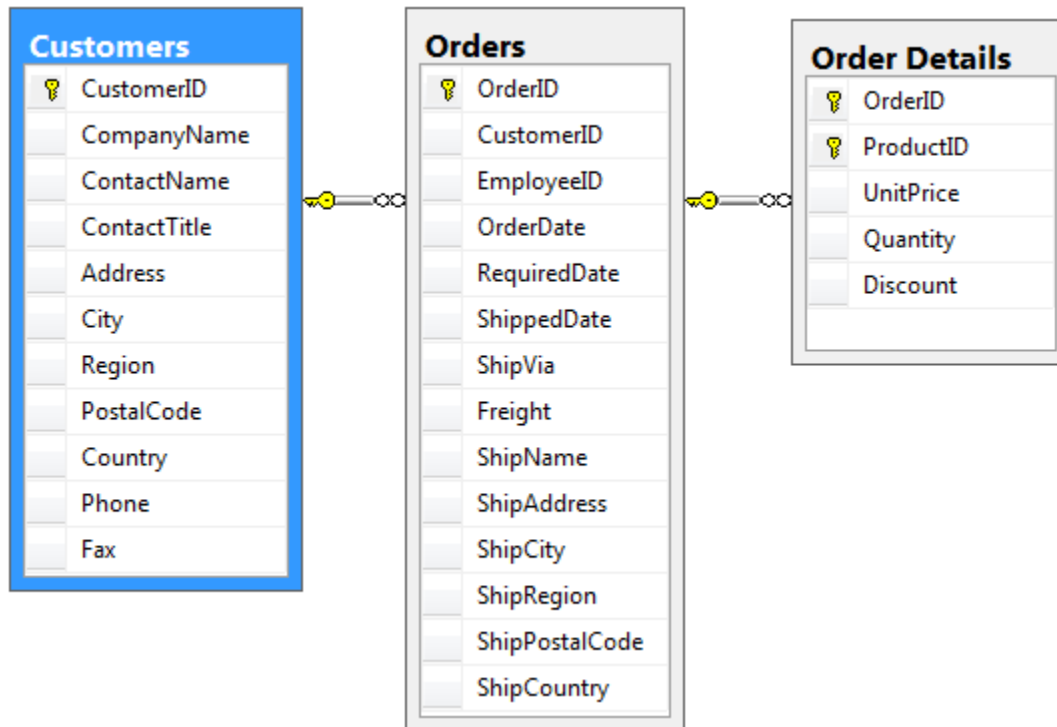


Highlight a result and press **OK** or press *Enter* on the keyboard to open the properties page of the element in the *Project Browser* and have it highlighted in the *Project Explorer*.



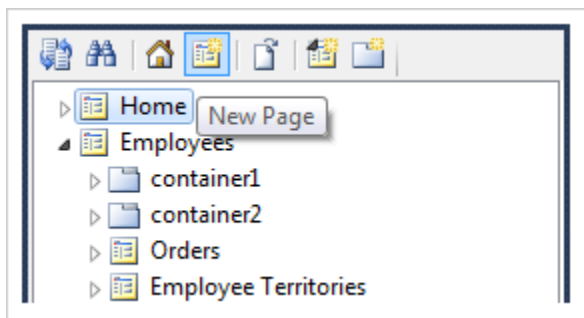
## Creating a Three-Level Master-Detail Page

Let's create a page with a three-level master-detail relationship between *Customers*, *Orders*, and *Order Details* controllers.



### Creating the Page

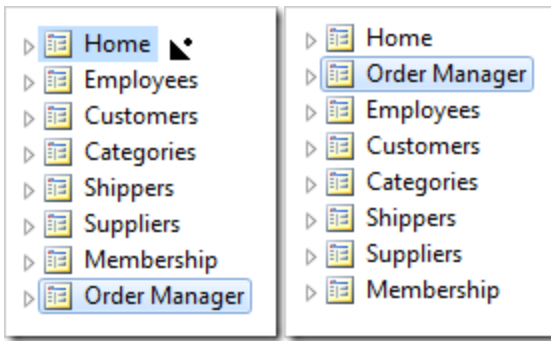
Start the [Project Designer](#). On the toolbar of the *Project Explorer*, click on the *New Page* icon.



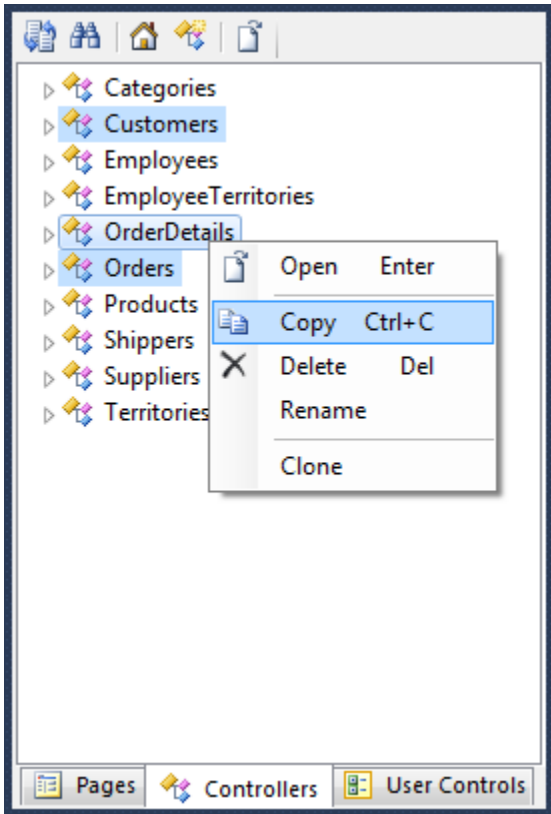
Enter "OrderManager" in the page *Name* and press *OK* to create a new page.

Property	Value
Name	OrderManager

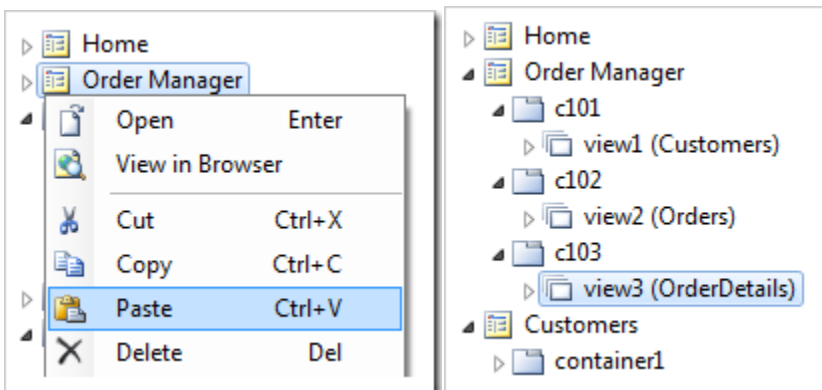
The *Order Manager* page will be added to the bottom of the list of pages in the *Project Explorer*. Drag *Order Manager* node to the right side of *Home* node to place it after the page.



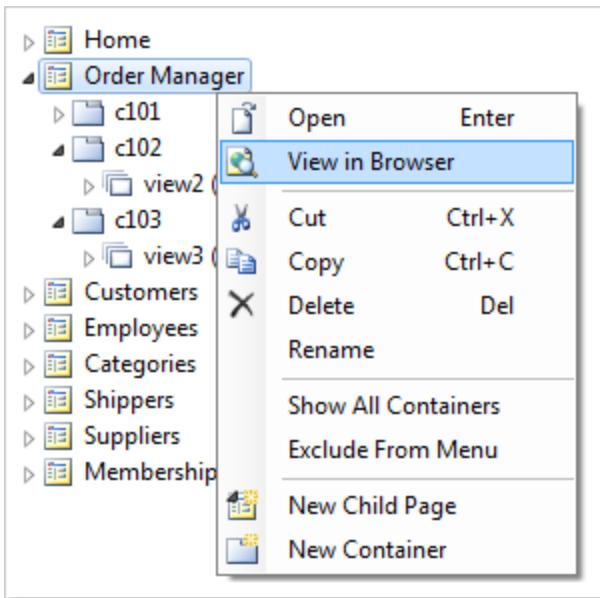
Switch to the *Controllers* tab in the *Project Explorer*. While holding *Ctrl*, select the controllers in the following order: *Customers*, *Orders*, *OrderDetails*. Right-click on *OrderDetails*, and press *Copy*.



Switch back to the *Pages* tab. Right-click on *Order Manager* page node, and press *Paste* option. The controllers will be instantiated as *data views* in separate *containers* on the *page*.



Right-click on *Order Manager* page node, and press *View in Browser*.



Navigate to the *Order Manager* page. Three views will be available on the page, but they do not have any master-detail relationships established between them. The three data views are completely independent from each other.

Home > Order Manager

## Order Manager

This is a list of customers.

Quick Find  New Customers Edit Delete Actions Report View: **Customers**

A filter has been applied. **Company Name** equals **Antonio Moreno Taquería.** X

Customer #	Company Name	Contact Name	Contact Title	Address	City	Region	Postal Code	Country	Phone
ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	n/a	05023	Mexico	(5) 555-3932

Showing 1-1 of 1 items | [Refresh](#)

This is a list of orders.

Quick Find  New Orders Actions Report View: **Orders**

A filter has been applied. **Order Date** equals **7/31/1996.** X

Customer Company Name	Employee Last Name	Order Date	Required Date	Shipped Date	Ship Via Company Name	Freight	Ship Name	Ship Address	Ship City
White Clover Markets	Buchanan	7/31/1996	8/14/1996	8/9/1996	Speedy Express	\$4.56	White Clover Markets	1029 - 12th Ave. S.	Seattle

Showing 1-1 of 1 items | [Refresh](#)

This is a list of order details.

Quick Find  New Order Details Actions Report View: **Order Details**

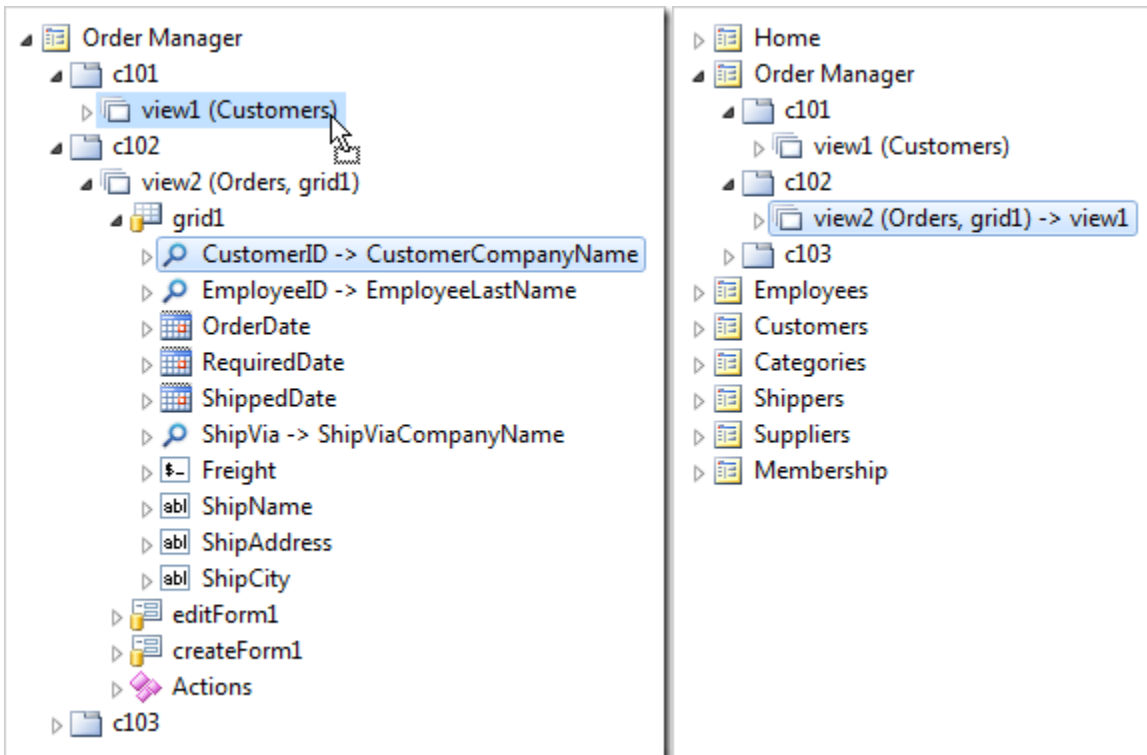
A filter has been applied. **Product Name** equals **Carnarvon Tigers.** **Quantity** equals **6.** X

Order Customer #	Product Name	Unit Price	Quantity	Discount	Order Customer Company Name	Order Employee Last Name	Order Ship Via Company Name	Product Category Name	Product Supplier Company Name
SPECD	Carnarvon Tigers	\$62.50	6	0	Spécialités du monde	Leverling	United Package	Seafood	Pavlova, Ltd.

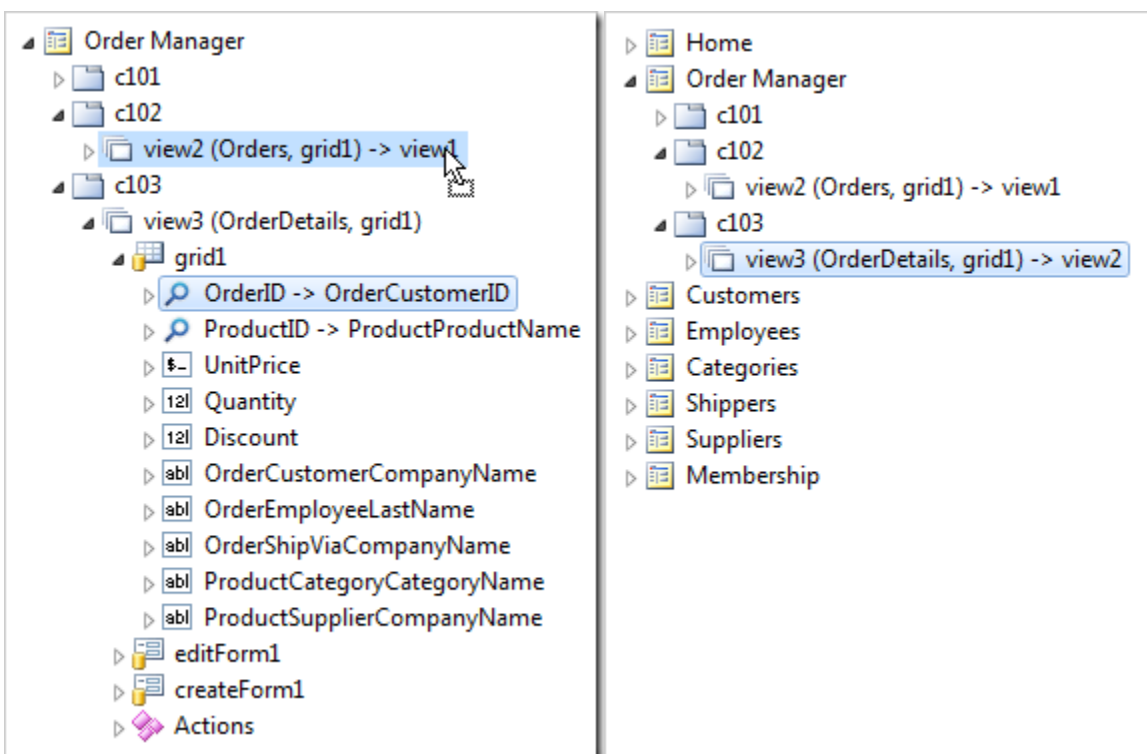
Showing 1-1 of 1 items | [Refresh](#)

## Establishing Master-Detail Relationships

Switch back to the *Project Designer*. In the *Project Explorer*, drag the data field node *Order Manager / c102 / view2 / grid1 / CustomerID* onto the view node *Order Manager / c101 / view1*. This will configure a master-detail relationship between *view2* and *view1*, with *CustomerID* as the filter field.



Next, drag the data field node *Order Manager / c103 / view3 / grid1 / OrderID* and drop it onto view node *Order Manager / c102 / view2*.




On the toolbar, press *Browse*.


Only the *Customers* data view is now visible on the *Order Manager* page.

[Home](#) > [Order Manager](#)

## Order Manager

This is a list of customers.



 New Customers


Actions ▾

Report ▾

View: **Customers** ▾

Customer #	Company Name	Contact Name	Contact Title	Address	City	Region	Postal Code	Country	Phone
<a href="#">ALFKI</a>	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	n/a	12209	Germany	030-0074321
<a href="#">ANATR</a>	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	n/a	05021	Mexico	(5) 555-4729
<a href="#">ANTON</a>	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	n/a	05023	Mexico	(5) 555-3932
<a href="#">AROUT</a>	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	n/a	WA1 1DP	UK	(171) 555-7788
<a href="#">BERGS</a>	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8	Luleå	n/a	S-958 22	Sweden	0921-12 34 65
<a href="#">BLAUS</a>	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim	n/a	68306	Germany	0621-08460
<a href="#">BLONP</a>	Blondesddsl père et fils	Frédérique Citeaux	Marketing Manager	24, place Kléber	Strasbourg	n/a	67000	France	88.60.15.31
<a href="#">BOLID</a>	Bólido Comidas preparadas	Martín Sommer	Owner	C/ Araquil, 67	Madrid	n/a	28023	Spain	(91) 555 22 82
<a href="#">BONAP</a>	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers	Marseille	n/a	13008	France	91.24.45.40
<a href="#">BOTTM</a>	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.	Tsawassen BC		T2F 8M4	Canada	(604) 555-4729

[« Previous](#) | Page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) | [Next »](#)

Items per page: [10](#), [15](#), [20](#), [25](#) | Showing **1-10** of **91** items | 

Select a record from the list of customers, and orders related to the selection will appear in a data view underneath.

# Order Manager

This is a list of customers.

Quick Find New Customers Edit Delete Actions Report View: **Customers**

A filter has been applied. **Company Name** equals **Ana Trujillo Emparedados y helados.**

Customer #	Company Name	Contact Name	Contact Title	Address	City	Region	Postal Code	Country	Phone
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	n/a	05021	Mexico	(5) 555-4729

Showing 1-1 of 1 items |

## Orders

This is a list of orders.

Quick Find New Orders Actions Report View: **Orders**

Employee Last Name	Order Date	Required Date	Shipped Date	Ship Via Company Name	Freight	Ship Name	Ship Address	Ship City
King	9/18/1996	10/16/1996	9/24/1996	Federal Shipping	\$1.61	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	México D.F.
Leverling	8/8/1997	9/5/1997	8/14/1997	Speedy Express	\$43.90	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	México D.F.
Leverling	11/28/1997	12/26/1997	12/12/1997	Federal Shipping	\$11.99	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	México D.F.
Peacock	3/4/1998	4/1/1998	3/11/1998	Federal Shipping	\$39.92	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	México D.F.

Showing 1-4 of 4 items |

Select an order, and order details for that order will be displayed.

This is a list of customers.

	<input type="text" value="Quick Find"/>		New Customers	Edit	Delete	Actions ▾	Report ▾	View: <b>Customers</b> ▾
A filter has been applied. <b>Company Name</b> equals <b>Ana Trujillo Emparedados y helados.</b>								
Customer #	Company Name	Contact Name	Contact Title	Address	City	Region	Postal Code	Country Phone
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	n/a	05021	Mexico (5) 555-4729
Showing 1-1 of 1 items								

## Orders

This is a list of orders.

	<input type="text" value="Quick Find"/>		New Orders	Edit	Delete	Actions ▾	Report ▾	View: <b>Orders</b> ▾
A filter has been applied. <b>Employee Last Name</b> equals <b>King.</b>								
Employee Last Name	Order Date	Required Date	Shipped Date	Ship Via Company Name	Freight	Ship Name	Ship Address	Ship City
King	9/18/1996	10/16/1996	9/24/1996	Federal Shipping	\$1.61	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	México D.F.
Showing 1-1 of 1 items								

## Order Details

This is a list of order details.

<div><div><div></div></div><div>Quick Find</div><div><div></div></div></div>	<div><div><div></div></div><div>New Order Details</div></div>			<div><div></div>Actions</div>	<div><div></div>Report</div>	<div>View: <b>Order Details</b></div>		
Product Name	Unit Price	Quantity	Discount	Order Customer Company Name	Order Employee Last Name	Order Ship Via Company Name	Product Category Name	Product Supplier Company Name
Gudbrandsdalsost	\$28.80	1	0	Ana Trujillo Emparedados y helados	King	Federal Shipping	Dairy Products	Norske Meierier
Outback Lager	\$12.00	5	0	Ana Trujillo Emparedados y helados	King	Federal Shipping	Beverages	Pavlova, Ltd.
Showing 1-2 of 2 items   <div><div></div></div>								

Select a child order detail. The record will be displayed in a modal form.

# Order Manager

This is a list of customers.

Quick Find

New Customers

Edit

Delete

Actions

Report

View: Customers

A filter has been applied. Company Name equals Ana Trujillo Emparedados y helados.

Customer#	Company Name	Contact Name	Contact Title	Address	City	Region	Postal Code	Country	Phone
ANATR	Ana y he								

## Orders

This is a list of orders.

Quick Find

A filter has been applied.

Employee Last Name	Order ID	Order Date	Order Due Date	Order Ship Date	Order Ship Via	Order Ship Via Company	Order Ship Via Name	Order Ship Via Address	Order Ship Via City	Order Ship Via Region	Order Ship Via Postal Code	Order Ship Via Country	Order Ship Via Phone
King	9												

## Order Details

This is a list of order details.

Quick Find

A filter has been applied.

Product Name	Unit Price	Quantity	Discount
Gudbrandsdalsost	\$28.80	1	0

### Reference Information

Additional details about order details are provided in the reference information section.

Order Customer Company Name	Ana Trujillo Emparedados y helados
Order Employee Last Name	King
Order Ship Via Company Name	Federal Shipping
Product Category Name	Dairy Products
Product Supplier Company Name	Norske Meierier

↑ ↓

\* - indicates a required field

OK Delete Cancel

If a master data view enters "New" mode, the child data views will automatically become hidden.

For example, if a new customer is created, the data views *Orders* and *Order Details* will be invisible. If a new order is created, only *Order Details* data view will disappear.

Please fill this form and click OK button to create a new customers record. Click Cancel to return to the previous screen.

Record ▾

View: **New Customers** ▾

\* - indicates a required field

OKCancel

**New Customers**

Complete the form. Make sure to enter all required fields.

Customer# *	<input type="text"/>
Company Name *	<input type="text"/>
Contact Name	<input type="text"/>
Contact Title	<input type="text"/>
Address	<input type="text"/>
City	<input type="text"/>
Region	<input type="text"/>
Postal Code	<input type="text"/>
Country	<input type="text"/>
Phone	<input type="text"/>
Fax	<input type="text"/>

\* - indicates a required field

OKCancel

## Rearranging Data Fields

The application generator composes this view of *Customers* from the database.

Home > Order Manager

Order Manager

This is a list of customers.

Quick Find

New Customers

Actions

Report

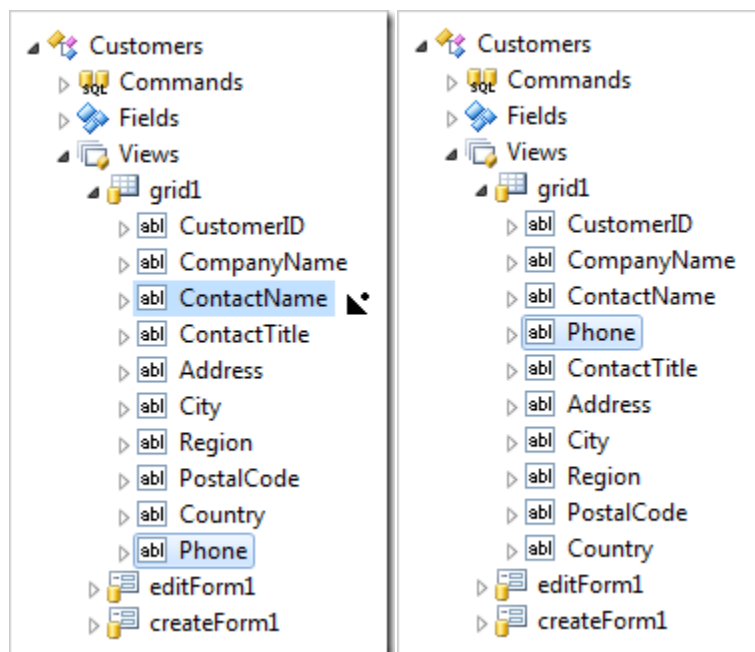
View: Customers

Customer #	Company Name	Contact Name	Contact Title	Address	City	Region	Postal Code	Country	Phone
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	n/a	12209	Germany	030-0074321
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	n/a	05021	Mexico	(5) 555-4729
ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	n/a	05023	Mexico	(5) 555-3932

Let's change some of the data fields presented in the view. For example, let's move *Phone* data field next to *Contact Name*, remove the *Region* data field, and add *Fax* to the view.

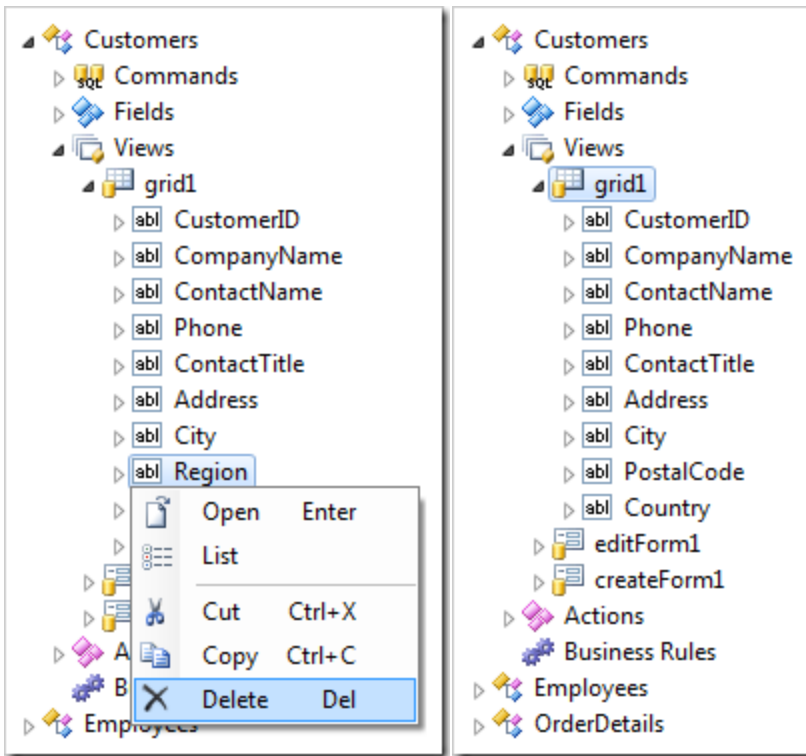
### Moving

Switch to the [Project Designer](#). In the *Project Explorer*, activate the *Controllers* tab and expand to *Order Manager / c101 / view1 / grid1* data view node. Drag *Phone* data field and drop it on the right side of *ContactName* data field to place it after the target.



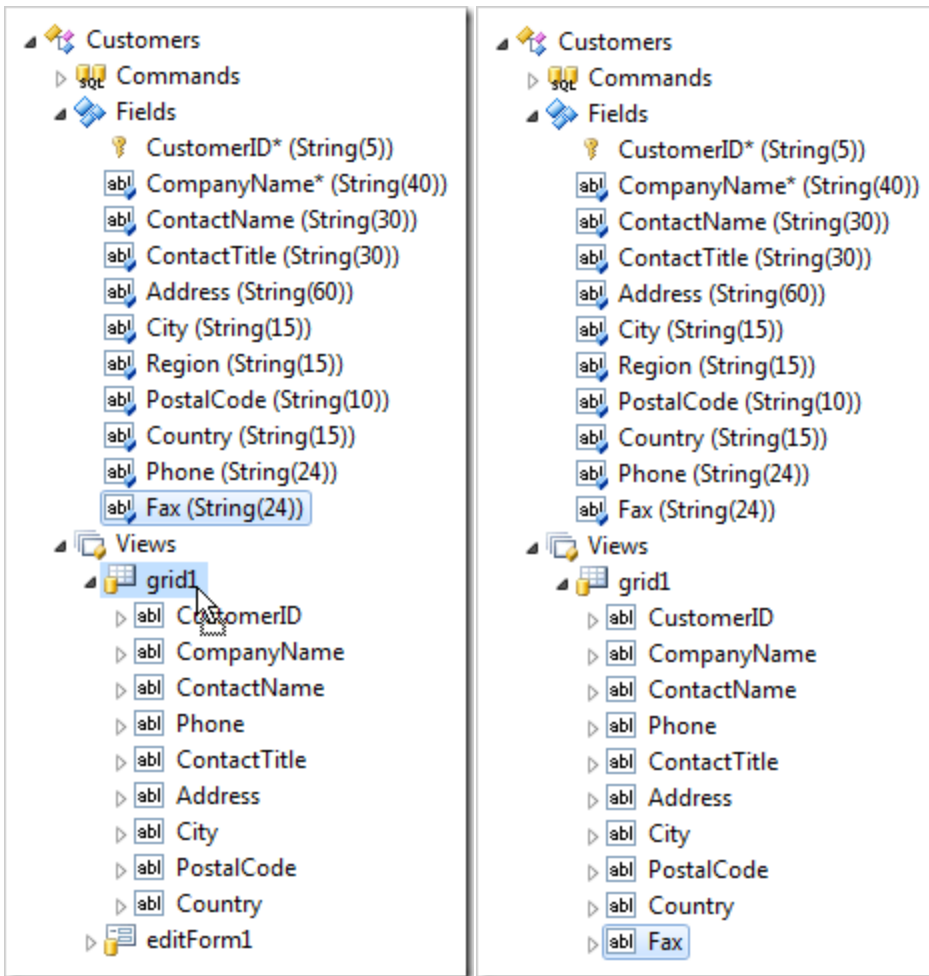
### Deleting

Right-click on *Order Manager / c101 / view1 / grid1 / Region* data field node, and select *Delete*.

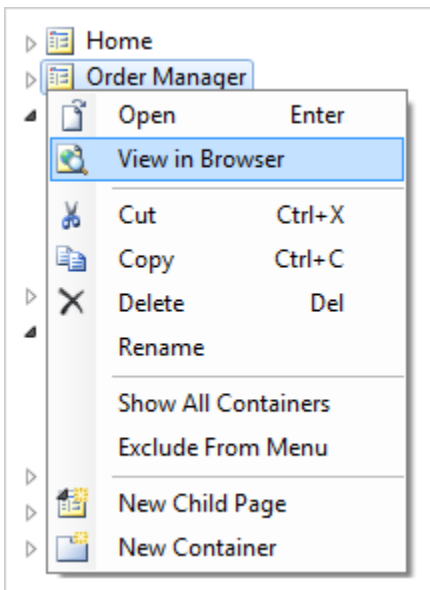


## Adding

Expand *Customers / Fields* node. Drag *Fax* field node onto *Customers / Views / grid1* node to create a data field from the *Fax* field.



Switch to the *Pages* tab in the *Project Explorer*. Right-click on *Order Manager* page node, and press *View in Browser*.



The *Customers* grid with new column layout will be displayed. The *Phone* field is more conveniently placed next to the *Contact Name*. The *Region* field has been removed. *Fax* has been added.

This is a list of customers.

Quick Find 		 New Customers		Actions ▾	Report ▾	View: <b>Customers</b> ▾			
Customer #	Company Name	Contact Name	Phone	Contact Title	Address	City	Postal Code	Country	Fax
ALFKI	Alfreds Futterkiste	Maria Anders	030-0074321	Sales Representative	Obere Str. 57	Berlin	12209	Germany	030-0076545
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	(5) 555-4729	Owner	Avda. de la Constitución 2222	México D.F.	05021	Mexico	(5) 555-3745
ANTON	Antonio Moreno Taquería	Antonio Moreno	(5) 555-3932	Owner	Mataderos 2312	México D.F.	05023	Mexico	n/a

## Designing Forms

Let's customize the forms on the *Order Manager* page.

### Floating Category Data Fields

When a customer is selected, the data item is displayed in the form view *editForm1*. The form data fields are rendered top-down.

Please review customers information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back.

Record ▾View: **Review Customers** ▾

⬆ ⬇ \* - indicates a required field

**Customers**

These are the fields of the customers record that can be edited.

Customer # *	ALFKI
Company Name *	Alfreds Futterkiste
Contact Name	Maria Anders
Contact Title	Sales Representative
Address	Obere Str. 57
City	Berlin
Region	
Postal Code	12209
Country	Germany
Phone	030-0074321
Fax	030-0076545

OKDeleteCancel

⬆ ⬇ \* - indicates a required field

**Orders**

This is a list of orders.

Quick Find 🔍

New Orders

Actions ▾

Report ▾

View: **Orders** ▾

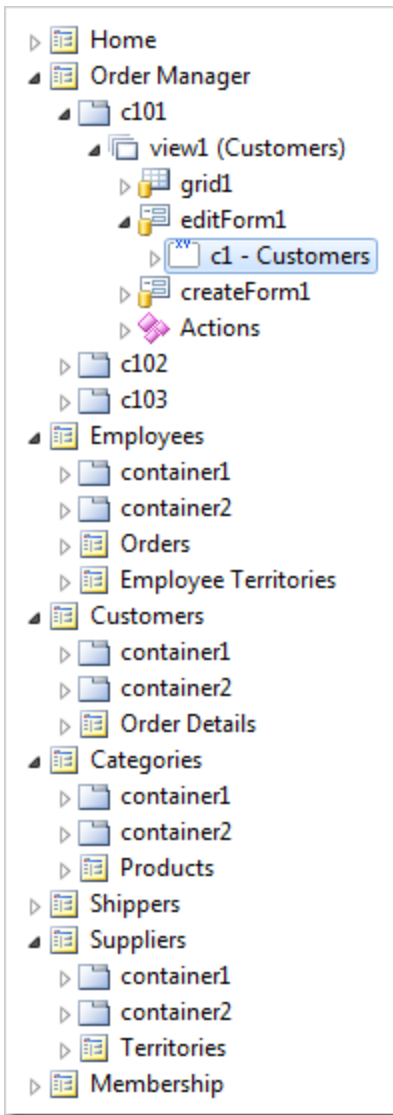
Employee Last Name	Order Date	Required Date	Shipped Date	Ship Via Company Name	Freight	Ship Name	Ship Address	Ship City
Suyama	8/25/1997	9/22/1997	9/2/1997	Speedy Express	\$29.46	Alfreds Futterkiste	Obere Str. 57	Berlin

OKDeleteCancel

Let's have the data fields displayed in a more compact fashion.

Switch back to the [Project Designer](#). On the *Pages* tab of *Project Explorer*, double-click on *Order Manager / c101 / view1 / editForm1 / c1 – Customers* category node.

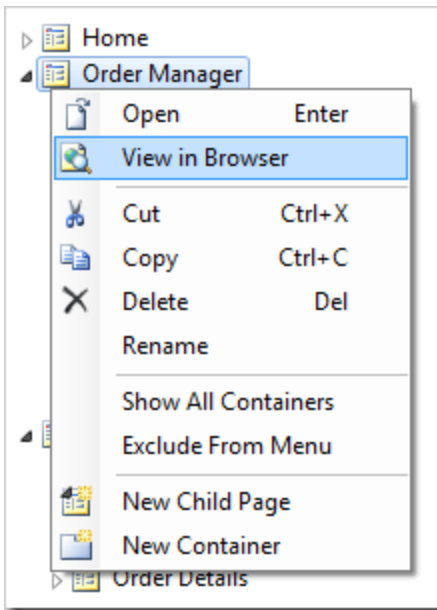
32 | Page



Change the *Float* property:

Property	Value
Float	Yes

Press OK to save the category. Right-click on *Order Manager* page node, and press *View in Browser*.



The data fields in *Customers* form view float from left to right and wrap when they reach the right side of the browser window which makes for a more compact presentation. If the browser window is resized, the form will resize as well.

Please review customers information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back.

New Customers
 Actions ▾
Report ▾
View: **Review Customers** ▾

Edit
Delete
Close

**Customers**

These are the fields of the customers record that can be edited.

Customer #	Company Name	Contact Name	Contact Title
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative
Address	City	Region	Postal Code
Obere Str. 57	Berlin	N/A	12209
Country	Phone	Fax	
Germany	030-0074321	030-0076545	

Edit
Delete
Close

**Orders**

This is a list of orders.

Quick Find 
New Orders
Actions ▾
Report ▾
View: **Orders** ▾

Employee Last Name	Order Date	Required Date	Shipped Date	Ship Via Company Name	Freight	Ship Name	Ship Address	Ship City
Suyama	8/25/1997	9/22/1997	9/2/1997	Speedy Express	\$29.46	Alfreds Futterkiste	Obere Str. 57	Berlin

## Configuring a Multi-Column Form

The picture below shows *editForm1* view of *Orders* data controller. Let's use categories to organize the data fields in multiple columns.

### Orders

Please review orders information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back.

New Orders

Actions ▾

Report ▾

View: **Review Orders** ▾

↑ ↓ 🖨

EditDeleteClose

### Orders

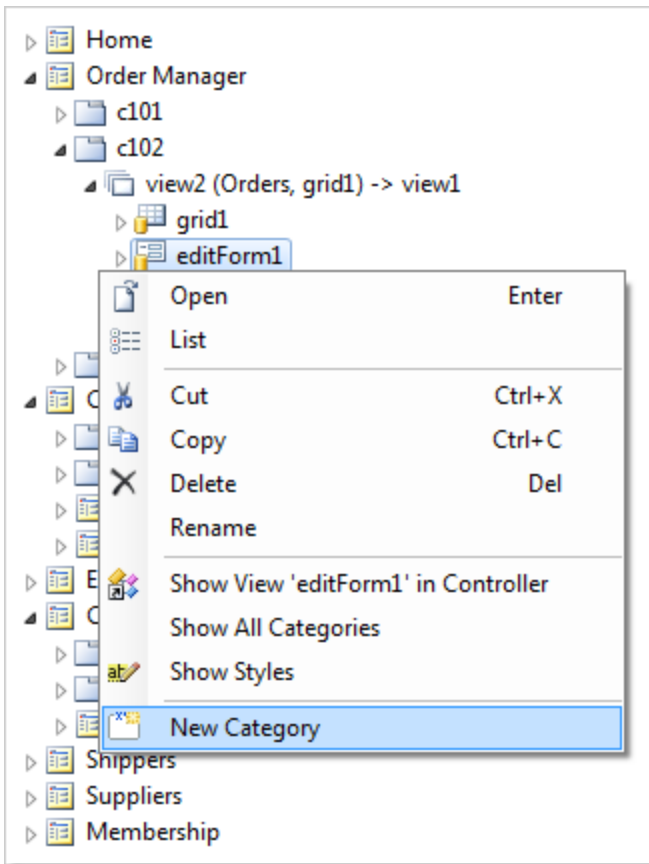
These are the fields of the orders record that can be edited.

Employee Last Name	Suyama ➕
Order Date	8/25/1997
Required Date	9/22/1997
Shipped Date	9/2/1997
Ship Via Company Name	Speedy Express ➕
Freight	\$29.46
Ship Name	Alfreds Futterkiste
Ship Address	Obere Str. 57
Ship City	Berlin
Ship Region	N/A
Ship Postal Code	12209
Ship Country	Germany

↑ ↓ 🖨

EditDeleteClose

In the *Project Explorer*, right-click on *Order Manager / c102 / view2 / editForm1* view node. Select *New Category* context menu option.

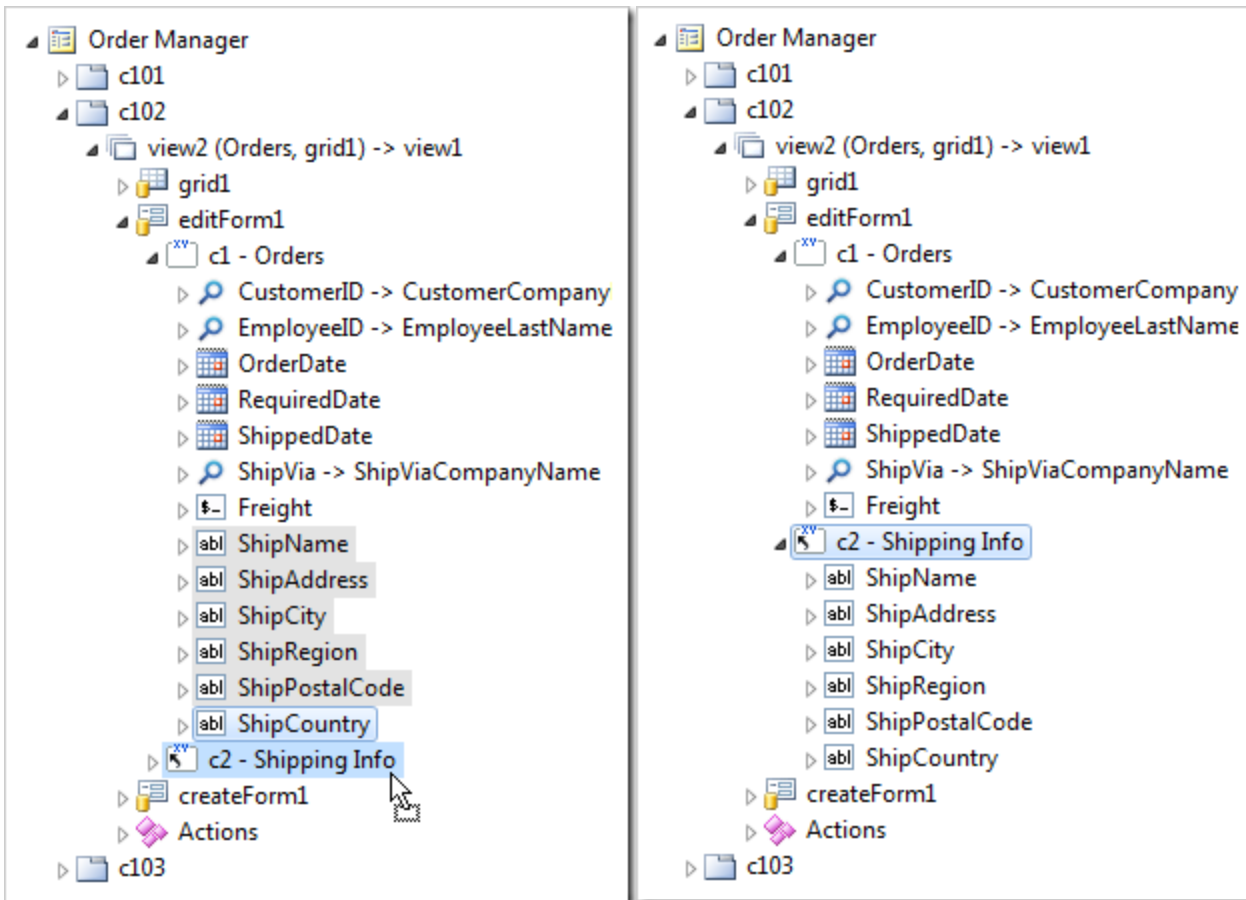


Assign the following properties:

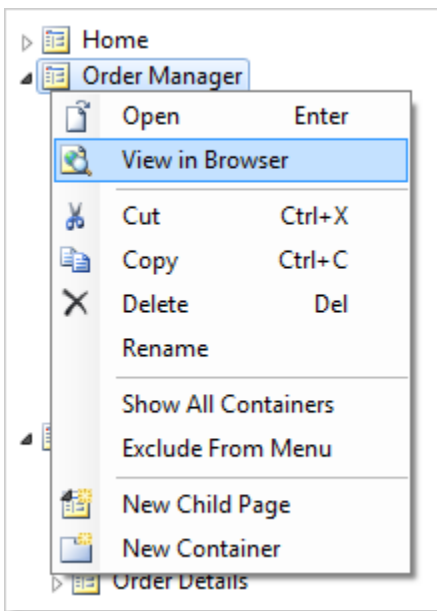
Property	Value
Header Text	Shipping Info
Description	This is the shipping information.
New Column	Yes

Press *OK* to save the category.

In the *Project Explorer*, expand *Order Manager / c102 / view2 / editForm1 / c1 – Orders* category node. Select all data fields that start with “Ship-”. Drag the data fields onto *c2 – Shipping Info* category node.



Right-click on *Order Manager* page node and select *View in Browser*.



Data fields on the *Orders* form will be split in two columns.

## Orders

Please review orders information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back.

[New Orders](#)[Actions](#) ▾[Report](#) ▾View: **Review Orders** ▾

### Orders

These are the fields of the orders record that can be edited.

Employee Last Name	Suyama ↕
Order Date	11/15/1996
Required Date	12/13/1996
Shipped Date	11/20/1996
Ship Via Company Name	Speedy Express ↕
Freight	\$41.95



### Shipping Info

This is the shipping information.

Ship Name	Around the Horn
Ship Address	Brook Farm Stratford St. Mary
Ship City	Colchester
Ship Region	Essex
Ship Postal Code	CO7 6JX
Ship Country	UK

[Edit](#)[Delete](#)[Close](#)[Edit](#)[Delete](#)[Close](#)

## Custom Category Template

The *Order Details* layout of *editForm1* view created by the application generator is shown below.

Review Order Details

Please review order details information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back.

Order Details

These are the fields of the order details record that can be edited.

Product Name	Rössle Sauerkraut ↕
Unit Price	\$45.60
Quantity	15
Discount	0.25

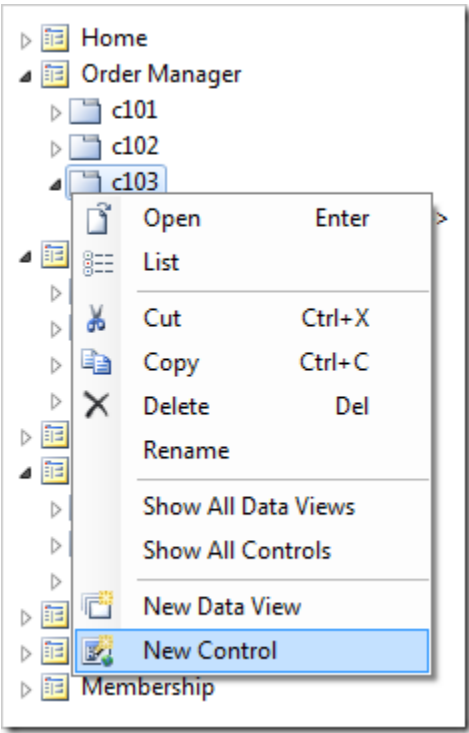
Reference Information

Additional details about order details are provided in the reference information section.

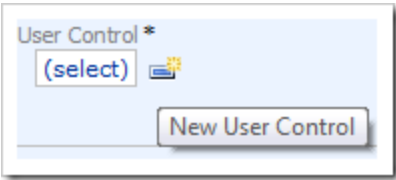
Order Customer Company Name	Alfreds Futterkiste
Order Employee Last Name	Suyama
Order Ship Via Company Name	Speedy Express
Product Category Name	Produce
Product Supplier Company Name	Plutzer Lebensmittelgroßmärkte AG

[Edit](#)[Delete](#)[Close](#)

Let's create a custom layout for the first category in *Order Details* form. In the *Project Explorer*, right-click on *Order Manager / c103* container node, and select *New Control*.



Next to the *User Control* property lookup, activate the *New User Control* icon.

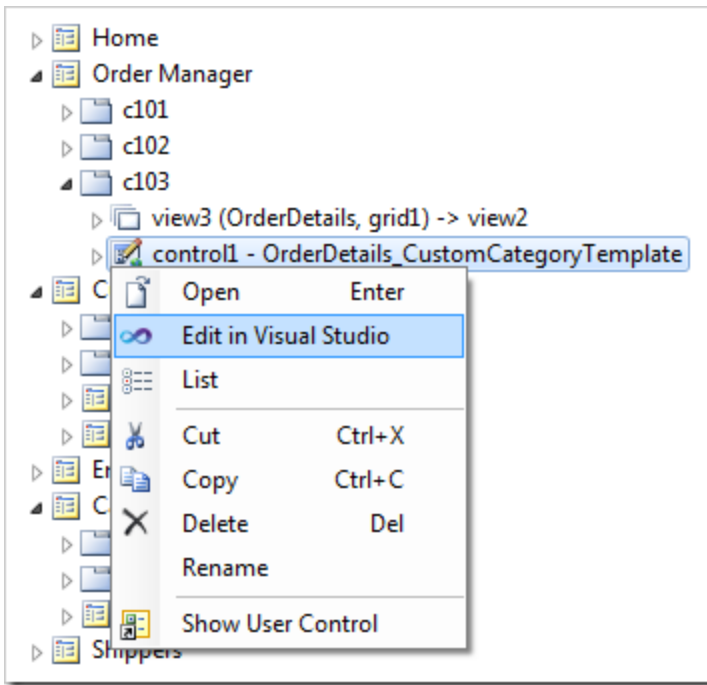


Assign a name to the user control.

Property	Value
Name	OrderDetails_CustomCategoryTemplate

Press *OK* to insert the new user control into the *User Control* property, and press *OK* again to save the control.

On the *Project Designer* toolbar, press *Generate*. When complete, right-click on *Order Manager / c103 / control1* node, and press *Edit in Visual Studio*.



The template file will open in *Visual Studio*. On the menu, select *Edit | Advanced | Format Document*. Replace everything after the `<%@ Control %>` element with the following:

```
<div style="display: none;">
  <div id="OrderDetails_editForm1_c1">
    <table>
      <tr>
        <td style="padding-right: 36px;font-weight:bold;">Product:</td>
        <td style="padding-right: 18px;">
          <span class="FieldPlaceholder DataOnly">{ProductID}</span>
        </td>
        <td style="padding-right: 18px;">Discount:</td>
        <td>
          <span class="FieldPlaceholder DataOnly">{Discount}</span>
        </td>
      </tr>
      <tr>
        <td>Unit Price:</td>
        <td>
          <span class="FieldPlaceholder DataOnly">{UnitPrice}</span>
        </td>
        <td>Quantity:</td>
        <td>
          <span class="FieldPlaceholder DataOnly">{Quantity}</span>
        </td>
      </tr>
    </table>
  </div>
</div>
```

Save the file.

Refresh the web browser page and select a customer, order, and order detail data rows. The first category of *editForm1* view of *Order Details* data controller will be rendered using the new template.

## Review Order Details



Please review order details information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back.

### Order Details

These are the fields of the order details record that can be edited.

<b>Product:</b>	<input type="text" value="Rössle Sauerkraut"/>		<b>Discount:</b>	<input type="text" value="0.25"/>
<b>Unit Price:</b>	<input type="text" value="\$45.60"/>		<b>Quantity:</b>	<input type="text" value="15"/>

### Reference Information

Additional details about order details are provided in the reference information section.

Order Customer Company Name	Alfreds Futterkiste
Order Employee Last Name	Suyama
Order Ship Via Company Name	Speedy Express
Product Category Name	Produce
Product Supplier Company Name	Plutzer Lebensmittelgroßmärkte AG

↑ ↓ \* - indicates a required field

OK

Delete

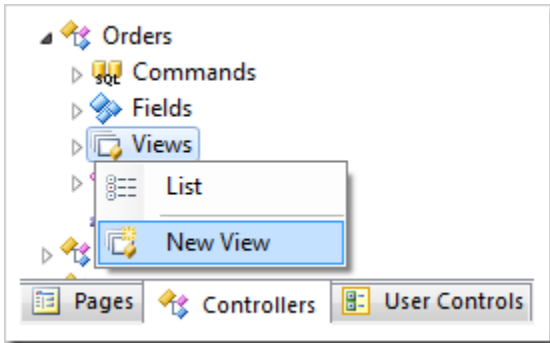
Cancel

## Configuring Charts

Let's create a chart that will graphically present orders broken down by quarter.

### Creating the View

Start the [Project Designer](#). In the *Project Explorer*, activate the *Controllers* tab. Right-click on *Orders / Views* node, and press *New View*.



Give this view the following properties:

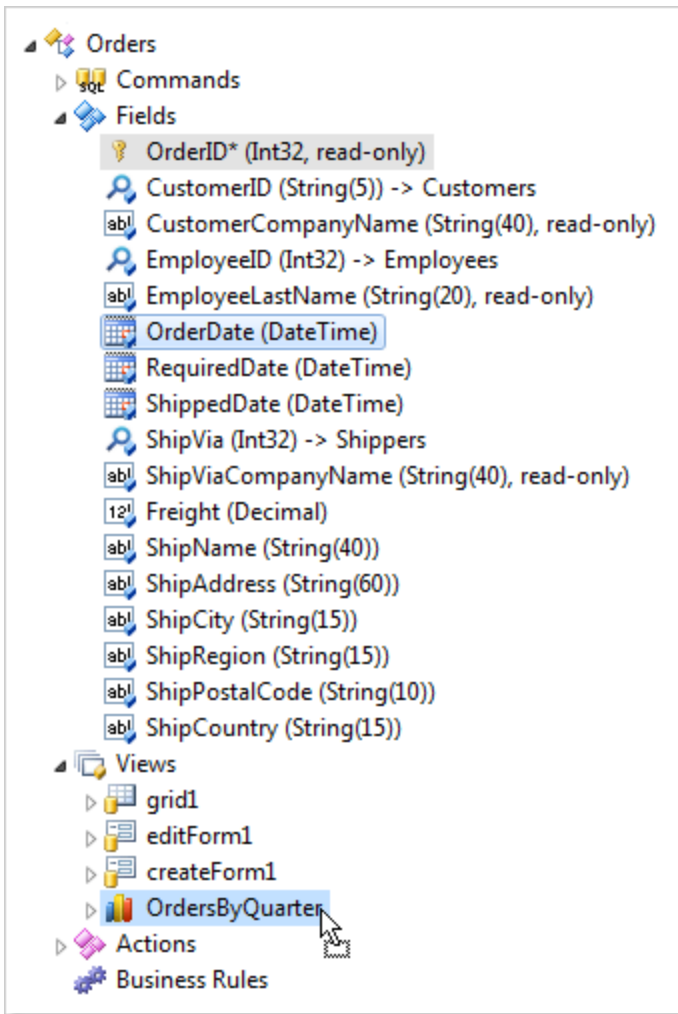
Property	Value
Id	OrdersByQuarter
Type	Chart
Label	Orders By Quarter

Press *OK* to save the view.

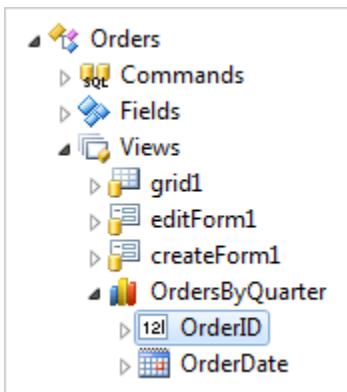
A chart requires at least two fields. One field will represent the values, and the other will form the *x*-axis of the chart that will group values by a certain criteria. A chart may have more than one field representing values.

For example, to display a chart of orders grouped by quarter, the *OrderID* field will represent values and *OrderDate* will represent the *x*-axis.

In the *Project Explorer*, expand the *Orders / Fields* node. Using *Ctrl* key, select *OrderID* and *OrderDate* fields, and drag them onto *Orders / Views / OrdersByQuarter* view node.



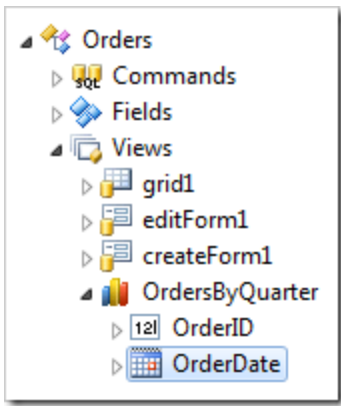
This will create two data fields. Double-click *Orders / Views / OrdersByQuarter / OrderID* data field node.



Assign the following values:

Property	Value
Aggregate	Count
Chart	Bar (Cylinder)

Press OK to save the data field. Double-click *Orders / Views / OrdersByQuarter / OrderDate* data field node.



Assign these values:

Property	Value
Data Format String	MMM yy
Chart	X, Quarter

Press *OK* to save.

### Viewing the Results

On the *Project Designer* toolbar, press *Browse*.

Navigate to the *Order Manager* page. Select a customer from the first data view. Use the *View Selector* in the top right corner of *Orders* data view to switch to *Orders By Quarter* view.

This is a list of customers.

*A filter has been applied. Company Name equals Alfreds Futterkiste.*

Customer #	Company Name	Contact Name	Phone	Contact Title	Address	City	Postal Code	Country	Fax
ALFKI	Alfreds Futterkiste	Maria Anders	030-0074321	Sales Representative	Obere Str. 57	Berlin	12209	Germany	030-0076545

Showing 1-1 of 1 items

### Orders

This is a list of orders.

Employee Last Name	Order Date	Required Date	Shipped Date	Ship Via Company Name	Freight	Ship Name		
Suyama	8/25/1997	9/22/1997	9/2/1997	Speedy Express	\$29.46	Alfreds Futterkiste	Obere Str. 57	Berlin
Peacock	10/3/1997	10/31/1997	10/13/1997	United Package	\$61.02	Alfred's Futterkiste	Obere Str. 57	Berlin
Peacock	10/13/1997	11/24/1997	10/21/1997	Speedy Express	\$23.94	Alfred's Futterkiste	Obere Str. 57	Berlin
Davolio	1/15/1998	2/12/1998	1/21/1998	Federal Shipping	\$69.53	Alfred's Futterkiste	Obere Str. 57	Berlin
Davolio	3/16/1998	4/27/1998	3/24/1998	Speedy Express	\$40.42	Alfred's Futterkiste	Obere Str. 57	Berlin

☒ Orders  
☐ Orders By Quarter

« Previous | Page: 1 2 | Next »

Items per page: 5, 10, 15, 20, 25 | Showing 1-5 of 6 items

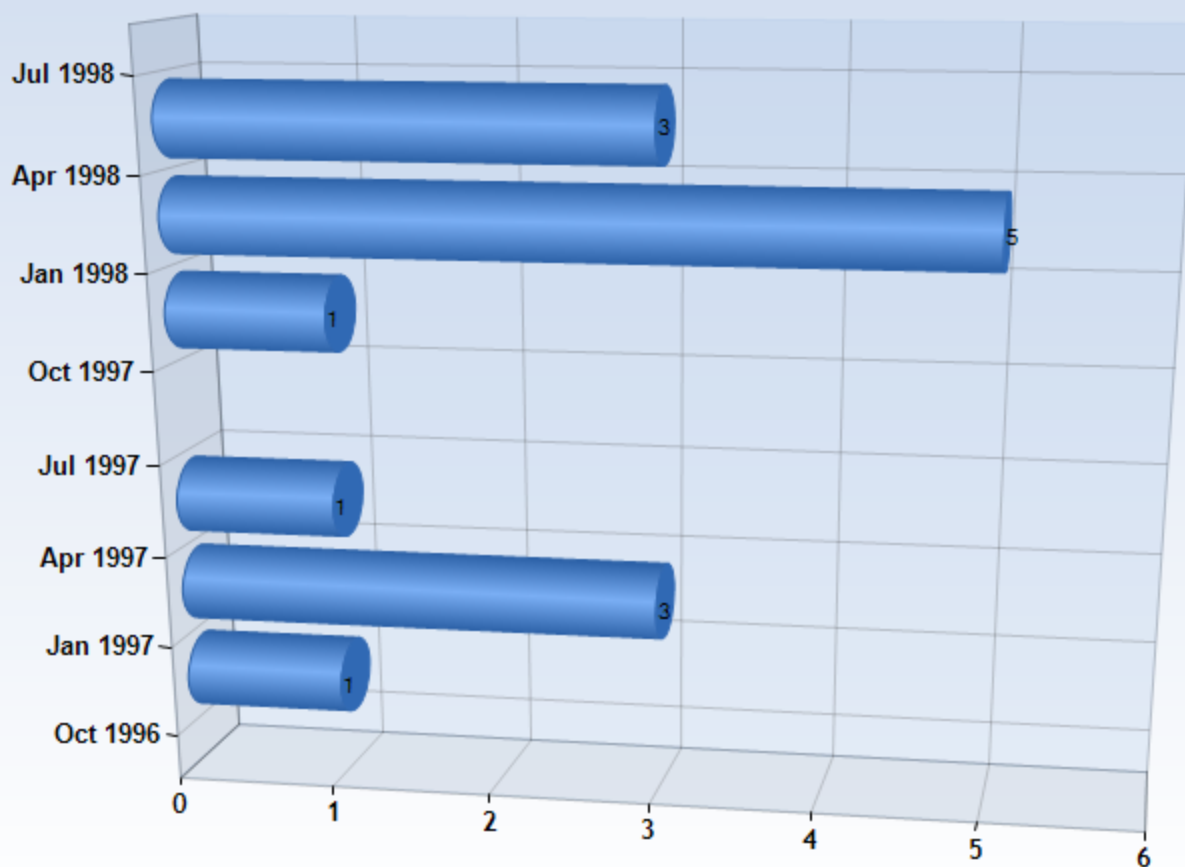
The chart will be displayed, showing the number of orders in each quarter filtered by the selected customer.



If you select another customer, the orders will change to reflect the new selection.

## Orders

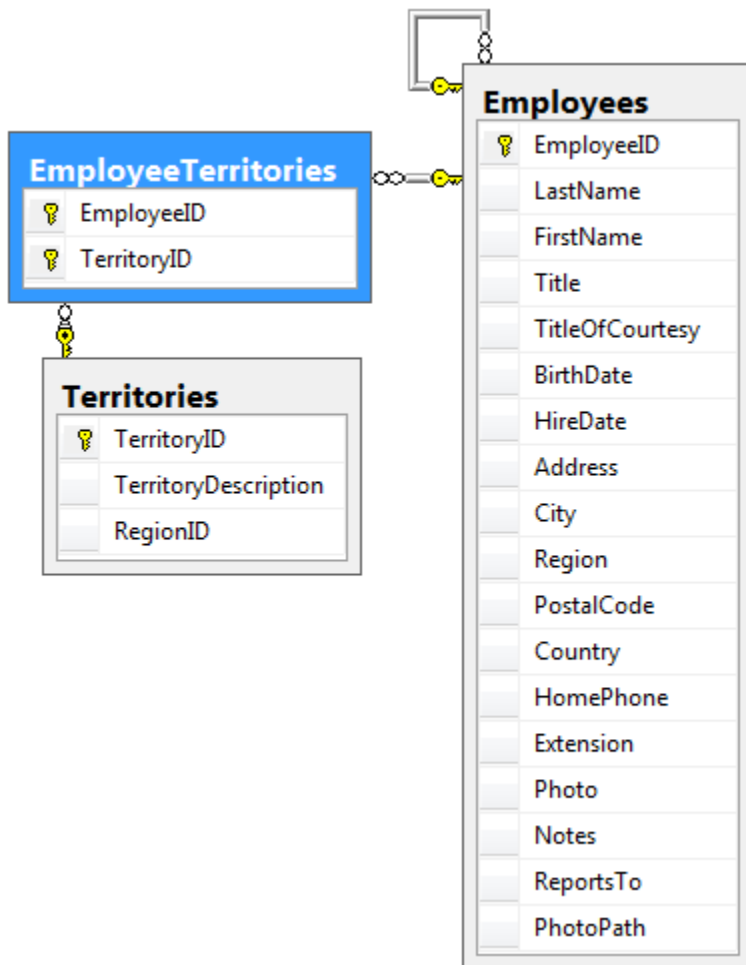
New Orders Actions Report View: **Orders By Quarter**



Showing 1-14 of 14 items |

## Creating Many-to-Many Fields

The [Northwind](#) database features a many-to-many relationship between *Territories* and *Employees*.



The app generator will create a page for *EmployeeTerritories* that allows editing records directly.

Home > Employees > Employee Territories

Employee Territories

This is a list of employee territories.

Quick Find

New

Actions ▾

Report ▾

View: Employee Territories ▾

Employee Last Name	Territory Description
Davolio	Wilton
Davolio	Neward
Fuller	Westboro

*EmployeeTerritories* is also available as a child data view on the *Employees* page. Selecting an employee will reveal a list of related territories.

This is a list of employees.

Quick Find New Employees Edit Delete Actions Report View: **Employees**

A filter has been applied. **First Name** equals **Nancy**.

Last Name	First Name	Title	Title Of Courtesy	Birth Date	Hire Date	Address	City	Region	Postal Code
Davolio	Nancy	Sales Representative	Ms.	12/8/1948	5/1/1992	507 - 20th Ave. E. Apt. 2A	Seattle	WA	98122

Showing 1-1 of 1 items |

**Employees** **Orders** **Employee Territories**

This is a list of employee territories.

Quick Find New Actions Report View: **Employee Territories**

Territory Description

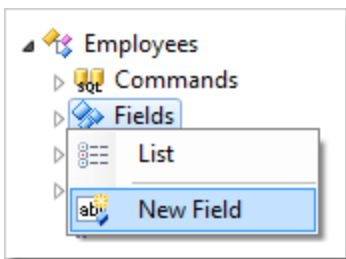
Wilton

Neward

Showing 1-2 of 2 items |

Let's create a many-to-many field on the employee edit form that will allow selecting territories in a check box list. Each option in the list will represent a territory from the *Territories* table. The checkbox next to the territory name will be checked if there is a record in the *EmployeeTerritories* table linking the selected employee with the territory. The relevant *EmployeeTerritories* records will be inserted or deleted when the user changes the selection of checkboxes and saves the employee record.

Start the [Project Designer](#). In the *Project Explorer*, switch to the *Controllers* tab. Right-click on *Employees / Fields* node, and press *New Field*.



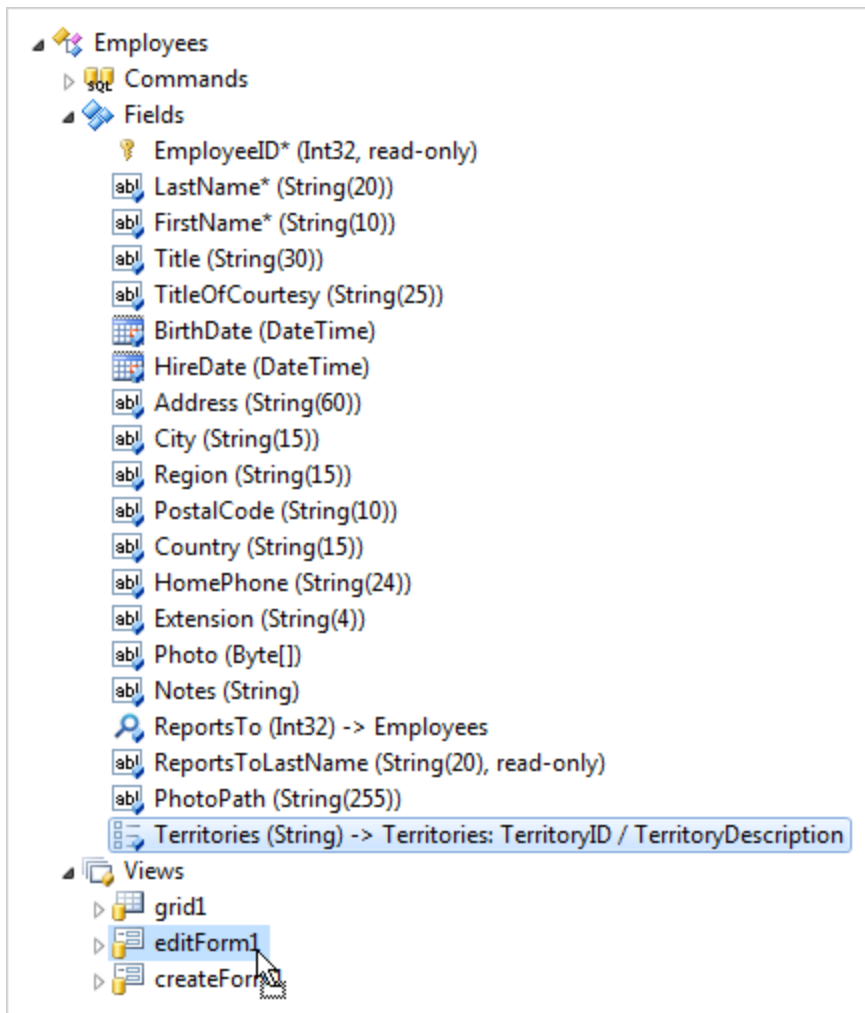
Give this field the following values:

Property	Value
Name	Territories
Allow null values	true
The value of this field is computed at run-time by SQL expression.	true
Label	Territories
Items Style	Check Box List

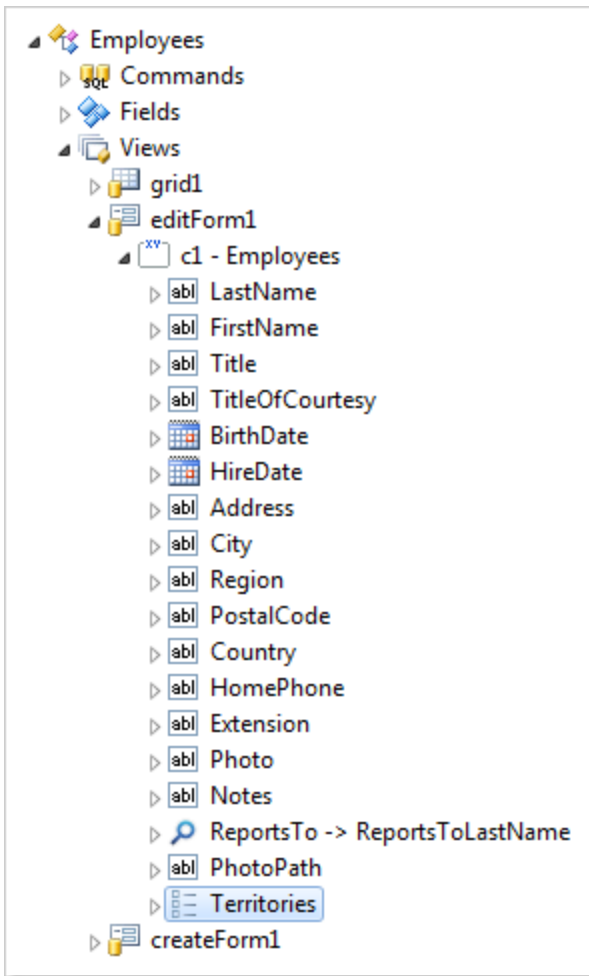
Items Data Controller	Territories
Data Value Field	TerritoryId
Data Text Field	TerritoryDescription
Target Controller	EmployeeTerritories

Press OK to save the field.

Drag *Employees / Fields / Territories* field node onto *Employees / Views / editForm1* view node.



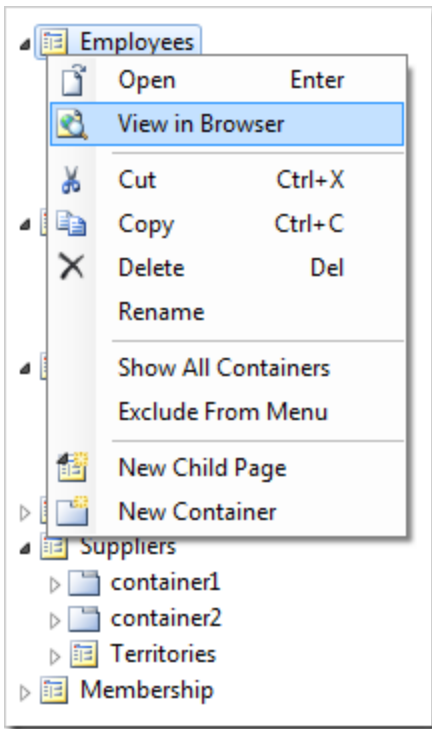
Double-click on *Employees / Views / editForm1 / Territories* data field node.



Change the *Columns* property:

Property	Value
Columns	5


Press *OK* to save. Switch to the *Pages* tab. Right-click on *Employees* page node, and press *View in Browser*.



View the details for an employee. *Territories* field will display a comma-separated list of values associated with the employee.


## Employees

These are the fields of the employees record that can be edited.

Last Name	Davolio
First Name	Nancy
Title	Sales Representative
Title Of Courtesy	Ms.
Birth Date	12/8/1948
Hire Date	5/1/1992
Address	507 - 20th Ave. E. Apt. 2A
City	Seattle
Region	WA
Postal Code	98122
Country	USA
Home Phone	(206) 555-9857
Extension	5467
Photo	
Notes	Education includes a BA in psychology from Colorado State University in 1970. She also completed "The Art of the Cold Call." Nancy is a member of Toastmasters International.
Reports To Last Name	Fuller ➡
Photo Path	<a href="http://accweb/emmployees/davolio.bmp">http://accweb/emmployees/davolio.bmp</a>
Territories	Neward , Wilton

[Edit](#)[Delete](#)[Close](#)

Press *Edit* button. All territories will be rendered as a check box list.

Photo 

Click [here](#) to upload or clear employees photo file.

Notes  
Education includes a BA in psychology from Colorado State University in 1970. She also completed "The Art of the Cold Call." Nancy is a member of Toastmasters International.


Reports To Last Name  
Fuller 

Photo Path  
<http://accweb/emmployees/davolio.bmp>

Territories

<input type="checkbox"/> Atlanta	<input type="checkbox"/> Cary	<input type="checkbox"/> Hoffman Estates	<input type="checkbox"/> Philadelphia	<input type="checkbox"/> Santa Monica
<input type="checkbox"/> Austin	<input type="checkbox"/> Chicago	<input type="checkbox"/> Hollis	<input type="checkbox"/> Phoenix	<input type="checkbox"/> Savannah
<input type="checkbox"/> Beachwood	<input type="checkbox"/> Colorado Springs	<input type="checkbox"/> Louisville	<input type="checkbox"/> Portsmouth	<input type="checkbox"/> Scottsdale
<input type="checkbox"/> Bedford	<input type="checkbox"/> Columbia	<input type="checkbox"/> Mellville	<input type="checkbox"/> Providence	<input type="checkbox"/> Seattle
<input type="checkbox"/> Bellevue	<input type="checkbox"/> Dallas	<input type="checkbox"/> Menlo Park	<input type="checkbox"/> Racine	<input type="checkbox"/> Southfield
<input type="checkbox"/> Bentonville	<input type="checkbox"/> Denver	<input type="checkbox"/> Minneapolis	<input type="checkbox"/> Redmond	<input type="checkbox"/> Tampa
<input type="checkbox"/> Bloomfield Hills	<input type="checkbox"/> Edison	<input type="checkbox"/> Morristown	<input type="checkbox"/> Rockville	<input type="checkbox"/> Troy
<input type="checkbox"/> Boston	<input type="checkbox"/> Fairport	<input type="checkbox"/> New York	<input type="checkbox"/> Roseville	<input type="checkbox"/> Westboro
<input type="checkbox"/> Braintree	<input type="checkbox"/> Findlay	<input type="checkbox"/> New York	<input type="checkbox"/> San Francisco	<input checked="" type="checkbox"/> Wilton
<input type="checkbox"/> Cambridge	<input type="checkbox"/> Georgetow	<input checked="" type="checkbox"/> Neward	<input type="checkbox"/> Santa Clara	
<input type="checkbox"/> Campbell	<input type="checkbox"/> Greensboro	<input type="checkbox"/> Orlando	<input type="checkbox"/> Santa Cruz	

↑ ↓ \* - indicates a required field

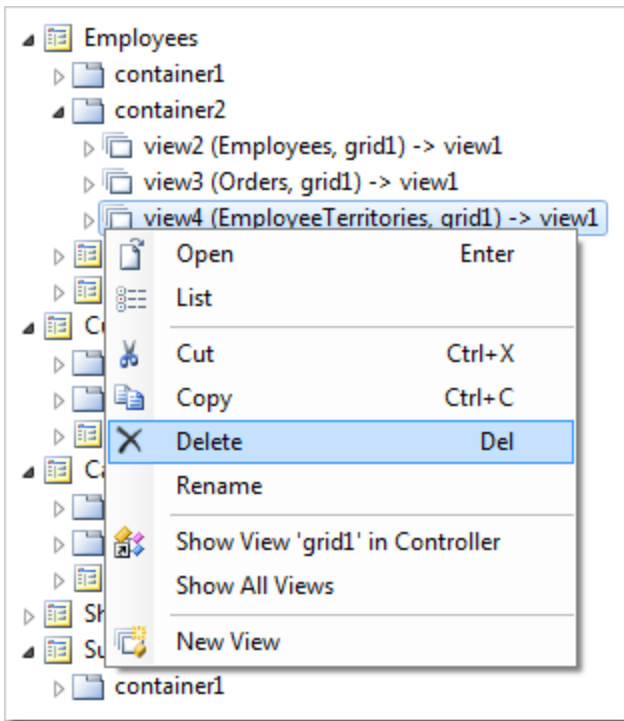
OK Delete Cancel

Check or uncheck several values, and press OK to save the record. Select the record to verify that the changes were persisted.

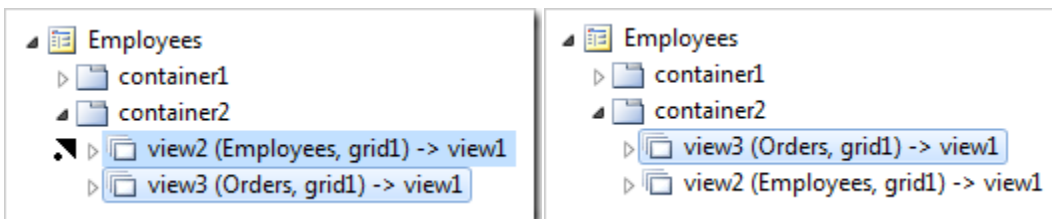
## Changing the Child Views

The *Employee Territories* child data view is now unnecessary.

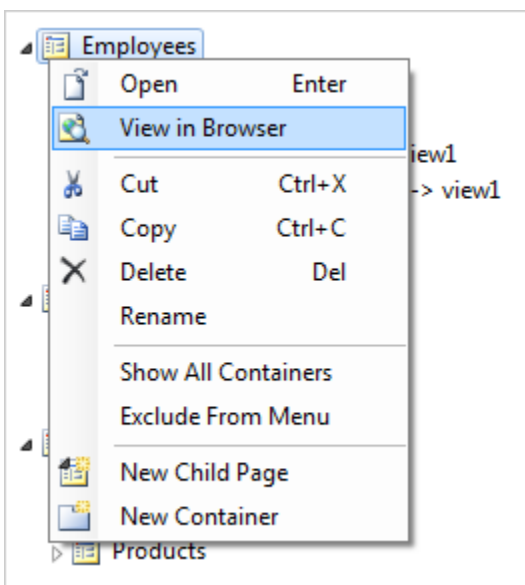
Switch back to the *Project Designer*. In the *Project Explorer*, right-click on *Employees / container2 / view4* node, and press *Delete*.



Drag *view3* node on the left side of *view2* to place it in the first position.



Right-click on *Employees* page node, and press *View in Browser*.



Select an employee from the list, and the *Orders* and *Employees* child data view tabs will appear below. The *Employee Territories* child data view tab is no longer present. The picture also shows the many-to-many field *Territories* in the master grid view.

This is a list of employees.

Quick Find

New Employees

Edit

Delete

Actions

Report

View: Employees

A filter has been applied. First Name equals Nancy.

Last Name	First Name	Title	Title Of Courtesy	Birth Date	Hire Date	Address	City	Region	Postal Code	Territories
Davolio	Nancy	Sales Representative	Ms.	12/8/1948	5/1/1992	507 - 20th Ave. E. Apt. 2A	Seattle	WA	98122	Neward , Wilton

Showing 1-1 of 1 items

## Orders Employees

This is a list of orders.

<div>Quick Find</div>		<div>New Orders</div>		<div>Actions</div>		<div>Report</div>		<div>View: Orders</div>	
Customer Company Name	Order Date	Required Date	Shipped Date	Ship Via Company Name	Freight	Ship Name	Ship Address	Ship City	
Ernst Handel	7/17/1996	8/14/1996	7/23/1996	Speedy Express	\$140.51	Ernst Handel	Kirchgasse 6	Graz	
Wartian Herkku	8/1/1996	8/29/1996	8/2/1996	Speedy Express	\$136.54	Wartian Herkku	Torikatu 38	Oulu	
Magazzini Alimentari Riuniti	8/7/1996	9/4/1996	8/9/1996	Speedy Express	\$26.93	Magazzini Alimentari Riuniti	Via Ludovico il Moro 22	Bergamo	
QUICK-Stop	8/20/1996	9/17/1996	8/26/1996	United Package	\$76.83	QUICK-Stop	Taucherstraße 10	Cunewalde	
Tradição Hipermercados	8/28/1996	9/25/1996	9/2/1996	United Package	\$1.35	Tradição Hipermercados	Av. Inês de Castro, 414	Sao Paulo	
« Previous   Page: 1 2 3 4 5 6 7 ... 25   Next »					Items per page: 5, 10, 15, 20, 25   Showing 1-5 of 123 items				

## Creating Calculated Fields

The picture below shows the *Order Details* form in edit mode.

**Review Order Details** [X]

Please review order details information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back.

**Order Details**

These are the fields of the order details record that can be edited.

<b>Product:</b>	<input type="text" value="Rössle Sauerkraut"/>	<b>Discount:</b>	<input type="text" value="0.25"/>
<b>Unit Price:</b>	<input type="text" value="\$45.60"/>	<b>Quantity:</b>	<input type="text" value="15"/>

**Reference Information**

Additional details about order details are provided in the reference information section.

Order Customer Company Name	Alfreds Futterkiste
Order Employee Last Name	Suyama
Order Ship Via Company Name	Speedy Express
Product Category Name	Produce
Product Supplier Company Name	Plutzer Lebensmittelgroßmärkte AG

↑ ↓ \* - indicates a required field

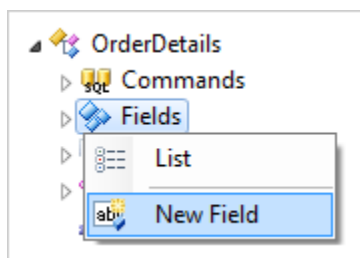
OK Delete Cancel

Let's create a calculated field called *Extended Price* that will display the extended price of the *Order Details* item.

Calculated fields are also known as *virtual* fields. The field value is not stored in the database - it is calculated at runtime based on values of other fields in the data row.

### Creating the virtual Field

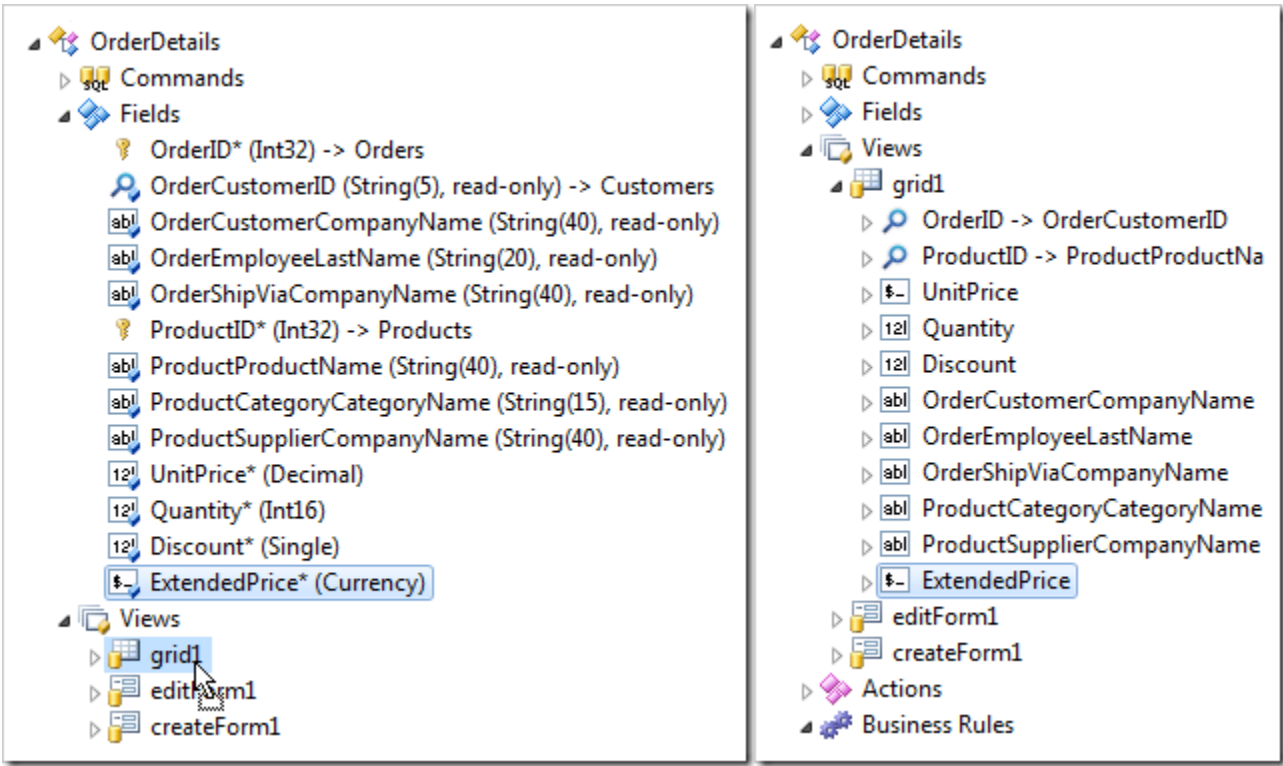
Activate the [Project Designer](#). In the *Project Explorer*, switch to the *Controllers* tab. Right-click on *OrderDetails / Fields* node, and press *New Field*.



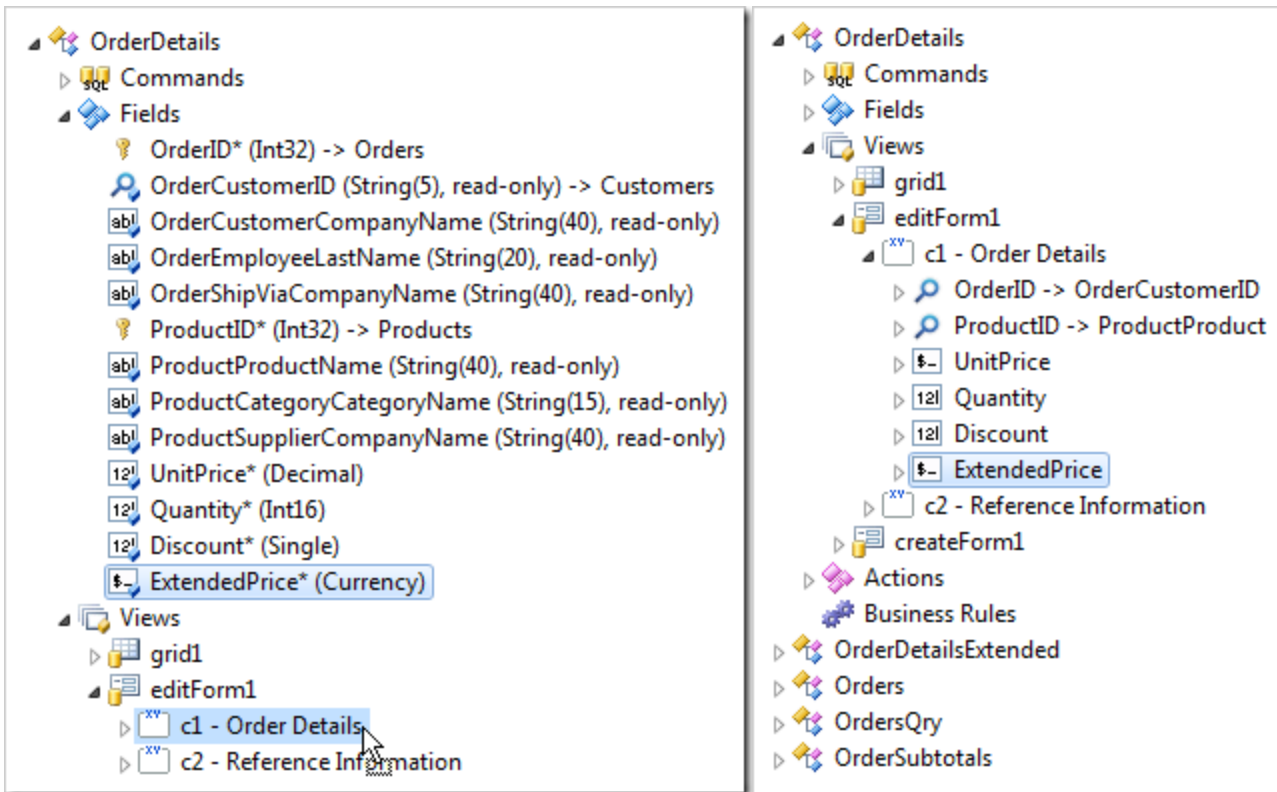
Assign the following values:

Property	Value
Name	ExtendedPrice
Type	Currency
Label	Extended Price
Allow Query-by-Example	true
Allow Sorting	true

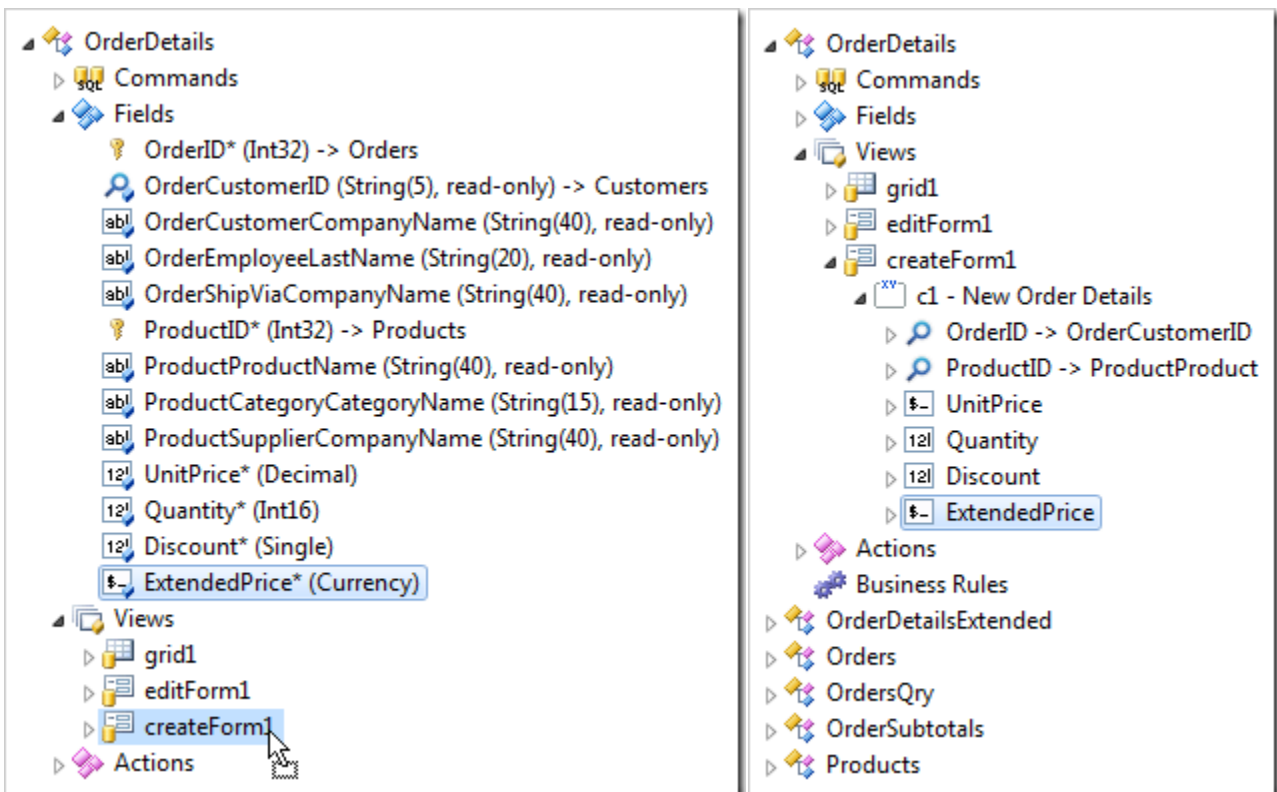
Press OK to save. Drag *OrderDetails / Fields / ExtendedPrice* node onto *OrderDetails / Views / grid1* node to bind the field to the view *grid1*.



Drag *OrderDetails / Fields / ExtendedPrice* node onto *OrderDetails / Views / editForm1 / c1 – Order Details* node to bind the field to view *editForm1*.



Finally, drag *OrderDetails / Fields / ExtendedPrice* node onto *OrderDetails / Views / createForm1* node.



A binding of *field* to a *view* is called a *data field*.

On the toolbar, press *Browse* and navigate to the *Order Manager* page. The *Extended Price* data field is visible in the grid and in the form views of *Order Details*. However, the field is rendered as blank.

**Order Details**  
 This is a list of order details.

View: **Order Details**

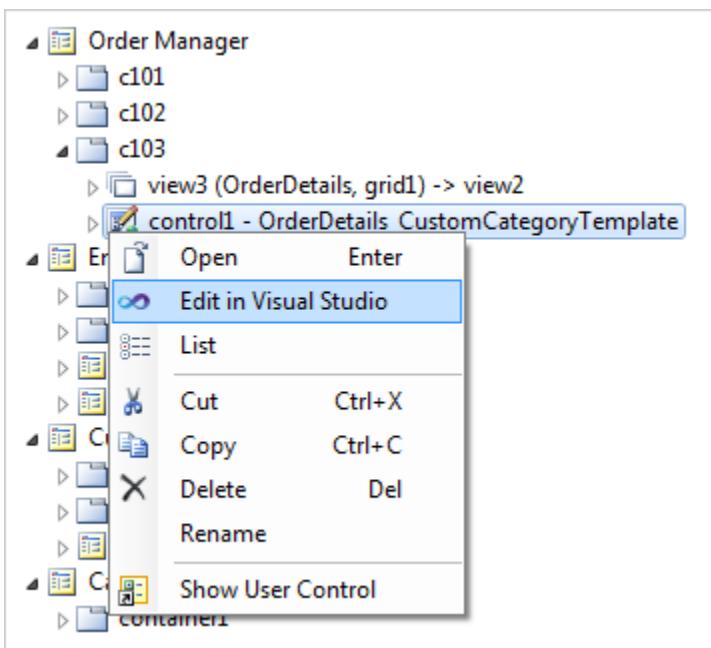
Product Name	Unit Price	Quantity	Discount	Order Customer Company Name	Order Employee Last Name	Order Ship Via Company Name	Product Category Name	Product Supplier Company Name	Extended Price
<div> <div>Rössle Sauerkraut</div> <div></div> </div>	\$45.60	15	0.25	Alfreds Futterkiste	Suyama	Speedy Express	Produce	Plutzer Lebensmittelgroßmärkte AG	

Note that if you have a [custom category template](#) associated with the form view, then the field will not be visible automatically. In that case, the custom category template for `editForm1` needs to be updated to display the data field.

Read the next section to learn how to include the new virtual data field in the template, or skip to the following section discussing how to provide an [SQL Formula](#) for the field.

### Updating the Custom Category Template

Switch to the *Project Designer*. In the *Project Explorer*, switch to the *Pages* tab. Right-click on *Order Manager / c103 / control1* node and press *Edit in Visual Studio*.



The template file will be opened in *Visual Studio*. Replace the existing code after the `<%@ Control %>` element with the following:

```
<div style="display: none;">
  <div id="OrderDetails_editForm1_c1">
    <table>
      <tr>
        <td style="padding-right: 36px;font-weight:bold;">Product:</td>
```

```

<td style="padding-right: 18px;">
    <span class="FieldPlaceholder DataOnly">{ProductID}</span>
</td>
<td style="padding-right: 18px;">Discount:</td>
<td>
    <span class="FieldPlaceholder DataOnly">{Discount}</span>
</td>
</tr>
<tr>
<td>Unit Price:</td>
<td>
    <span class="FieldPlaceholder DataOnly">{UnitPrice}</span>
</td>
<td>Quantity:</td>
<td>
    <span class="FieldPlaceholder DataOnly">{Quantity}</span>
</td>
<td style="padding-left:18px;"><i>Extended Price:</i></td>
<td>
    <span class="FieldPlaceholder DataOnly">{ExtendedPrice}</span>
</td>
</tr>
</table>
</div>
</div>

```


Save the file, and refresh the webpage. The *Extended Price* data field will be displayed in edit form.

Review Order Details
X

Please review order details information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back.

**Order Details**  
These are the fields of the order details record that can be edited.

**Product:**

Rössle Sauerkraut


Discount:

0.25

Unit Price:

\$45.60

Quantity:

15

*Extended Price:*

**Reference Information**  
Additional details about order details are provided in the reference information section.



Order Customer Company Name
Alfreds Futterkiste

Order Employee Last Name
Suyama

Order Ship Via Company Name
Speedy Express

Product Category Name
Produce

Product Supplier Company Name
Plutzer Lebensmittelgroßmärkte AG

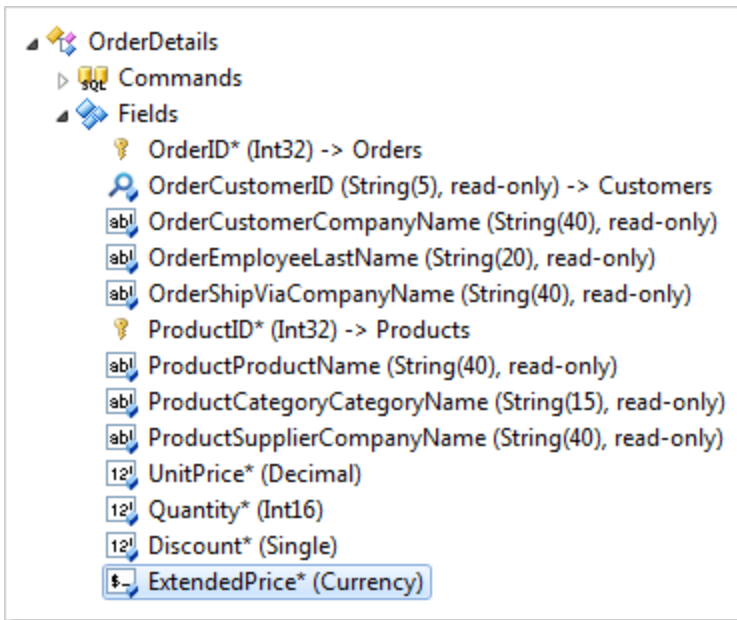


\* - indicates a required field

OK
Delete
Cancel

## SQL Formula

Let's provide an SQL expression evaluated when the data rows of order details are selected from the database.

In the *Project Explorer*, switch to the *Controllers* tab and double-click on *OrderDetails / Fields / ExtendedPrice* node.



Change the following:

Property	New Value
The value of this field is computed at run-time by SQL expression.	true
SQL Formula	<code>OrderDetails.UnitPrice * OrderDetails.Quantity * (1 - OrderDetails.Discount)</code>
Values of this field cannot be edited	true
Data Format String	c

Press OK to save the field. On the toolbar, press *Browse*. The *Extended Price* field will not be editable, and the value will be presented when the form is rendered.

**Order Details**  
These are the fields of the order details record that can be edited.

<b>Product:</b>	<input type="text" value="Rössle Sauerkraut"/>	<b>Discount:</b>	<input type="text" value="0.25"/>
<b>Unit Price:</b>	<input type="text" value="\$45.60"/>	<b>Quantity:</b>	<input type="text" value="15"/>

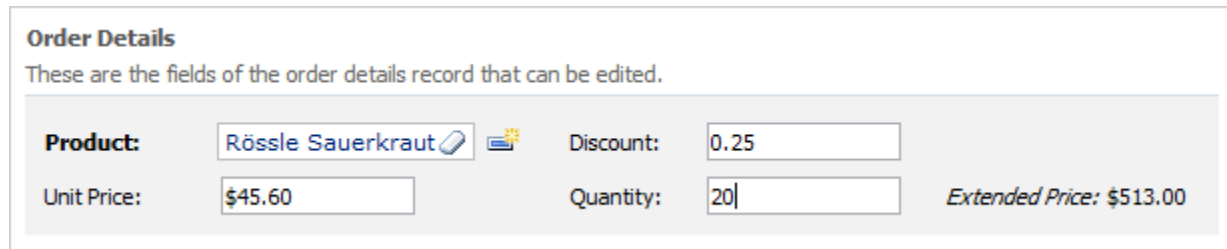
*Extended Price: \$513.00*

*SQL Formula* provides the highest possible performance for the calculation since it is being evaluated by the database engine. The complexity of calculated fields depends on the capabilities of the database engine. There are numerous built-in functions that the developer can take advantage of.

Fields based on *SQL Formula* can be [sorted](#) and [filtered](#) with the highest possible performance as well.

### Calculating Field Values Just-in-Time

*Extended Price* is not recalculated when a user changes values of *Unit Price*, *Discount*, or *Quantity* fields.



The screenshot shows a web form titled "Order Details" with a subtitle "These are the fields of the order details record that can be edited." The form contains several input fields: "Product" with a dropdown menu showing "Rössle Sauerkraut", "Unit Price" with a text box containing "\$45.60", "Discount" with a text box containing "0.25", and "Quantity" with a text box containing "20". To the right of the "Quantity" field, the calculated "Extended Price" is displayed as "\$513.00". There are also small icons next to the "Product" and "Unit Price" fields.

The *SQL Formula* of the *Extended Price* field is evaluated only when the data is selected from the database. If the user makes changes in the browser window, the values will not be submitted to the server until the record is saved. Developers can implement a server-side or client-side calculation that will be performed just-in-time as users change the values of the formula's base fields.

The application framework includes a business rule engine that allows implementing rules in SQL, C#/Visual Basic, or JavaScript.

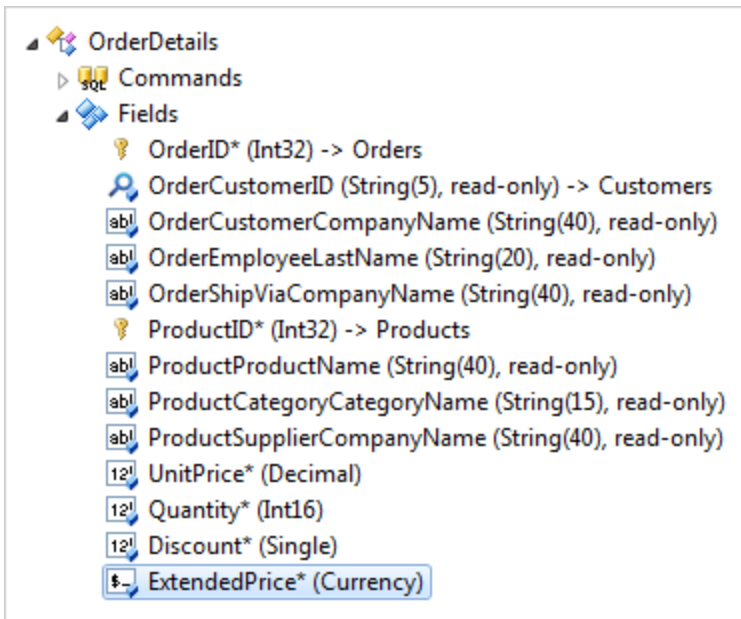
SQL and C#/Visual Basic business rules require a round-trip between the web browser and the web server. They can look up database information and interact with external systems when necessary.

JavaScript business rules are executed in the browser and provide the highest possible performance when server-side data is not required for calculation.

Let's consider implementing the calculation using all three flavors of business rules. Note that the developer needs only one of them to accomplish just-in-time calculation of *Extended Price*.

### SQL Business Rule

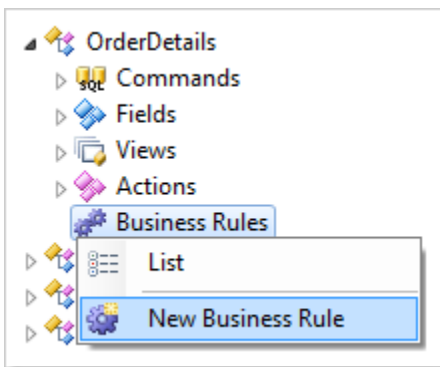
In the *Project Explorer*, double-click on *OrderDetails / Fields / ExtendedPrice* node.



Make the following changes:

Property	New Value
The value of this field is calculated by a business rule expression.	true
Context Fields	UnitPrice,Quantity,Discount

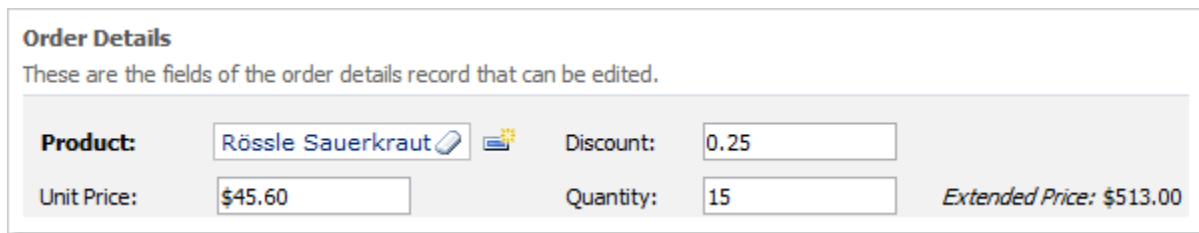
Press OK to save. Right-click on *OrderDetails / Business Rules* node, and press *New Business Rule*.



Assign these values:

Property	Value
Type	SQL
Command Name	Calculate
Phase	Execute
Script	<code>set @ExtendedPrice = @UnitPrice * @Quantity * (1 - @Discount)</code>

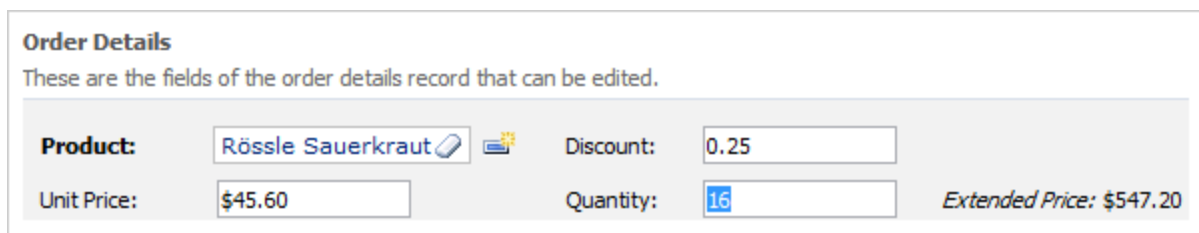
Press **OK** to save. On the toolbar, press *Browse*. Select and start editing an *Order Details* record.



**Order Details**  
These are the fields of the order details record that can be edited.

<b>Product:</b>	Rössle Sauerkraut	<b>Discount:</b>	0.25
<b>Unit Price:</b>	\$45.60	<b>Quantity:</b>	15
			<i>Extended Price:</i> \$513.00

Change the value in *Quantity* field. Press *Enter* or shift the focus to a different field. The *Extended Price* will be updated.



**Order Details**  
These are the fields of the order details record that can be edited.

<b>Product:</b>	Rössle Sauerkraut	<b>Discount:</b>	0.25
<b>Unit Price:</b>	\$45.60	<b>Quantity:</b>	16
			<i>Extended Price:</i> \$547.20

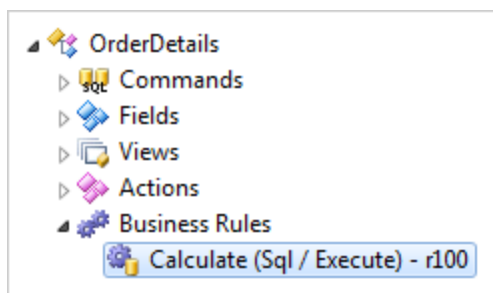
The client library executes *Calculate* action, which causes transfer of field values to the server. The application framework will pass the business rule script along with the parameter values to the database engine for execution. Then, it evaluates parameters and returns changed values to the client web browser.

SQL business rules involve client, application server, and database engine tiers. The advantage of SQL business rules is the ability to access any database information when necessary.

## Code Business Rule

The business rule can also be implemented using C# or Visual Basic.

If you have the SQL business rule created in the previous section, then you will need to delete or rename the business rule. Double-click on *OrderDetails / Business Rules / Calculate* node.

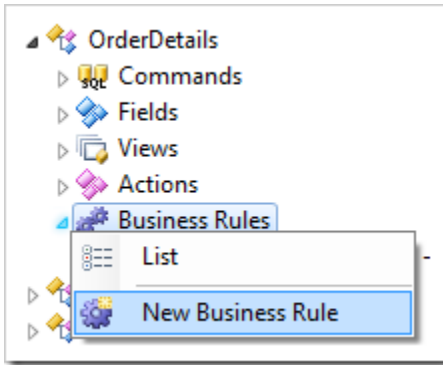


Change the *Command Name* property:

Property	Value
Command Name	DoNotRun

Press **OK** to save. Alternatively, business rule *r100* can be deleted.

Right-click on *OrderDetails / Business Rules* node, and press *New Business Rule*.



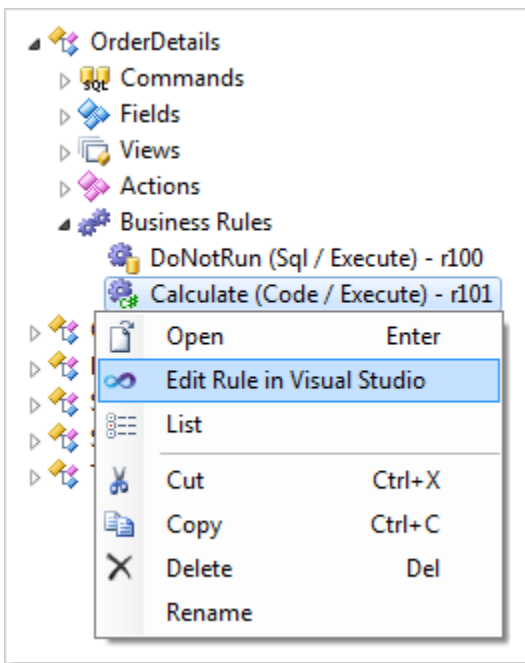
Assign these values:

Property	Value
Type	C# / Visual Basic
Command Name	Calculate
Phase	Execute

Press *OK* to save.

Code business rule files do not exist until the application generator has created them. On *Project Designer* toolbar, press *Browse*.

When complete, right-click *OrderDetails / Business Rules / Calculate* business rule node, and press *Edit Rule in Visual Studio*.



The file will be opened in *Visual Studio*. The entire class definition and parameters of the business rule method are already defined.

Replace the body of the rule with the call of *UpdateFieldValue* method:

```
C#:  
  
using System;  
using MyCompany.Data;  
  
namespace MyCompany.Rules  
{  
    public partial class OrderDetailsBusinessRules : MyCompany.Data.BusinessRules  
    {  
        [Rule("r101")]  
        public void r101Implementation(int? orderID,  
            string orderCustomerID, string orderCustomerCompanyName,  
            string orderEmployeeLastName, string orderShipViaCompanyName,  
            int? productID, string productProductName, string productCategoryCategoryName,  
            string productSupplierCompanyName, decimal? unitPrice, short? quantity,  
            float? discount, decimal? extendedPrice)  
        {  
            UpdateFieldValue("ExtendedPrice",  
                Convert.ToDouble(unitPrice.Value) * quantity.Value * (1 - discount.Value));  
        }  
    }  
}
```

*Visual Basic:*

```
Imports MyCompany.Data  
Imports System  
  
Namespace MyCompany.Rules  
  
    Partial Public Class OrderDetailsBusinessRules  
        Inherits MyCompany.Data.BusinessRules  
  
        <Rule("r101")> _  
        Public Sub r101Implementation(  
            ByVal orderID As Nullable(Of Integer),  
            ByVal orderCustomerID As String,  
            ByVal orderCustomerCompanyName As String,  
            ByVal orderEmployeeLastName As String,  
            ByVal orderShipViaCompanyName As String,  
            ByVal productID As Nullable(Of Integer),  
            ByVal productProductName As String,  
            ByVal productCategoryCategoryName As String,  
            ByVal productSupplierCompanyName As String,  
            ByVal unitPrice As Nullable(Of Decimal),  
            ByVal quantity As Nullable(Of Short),  
            ByVal discount As Nullable(Of Single),  
            ByVal extendedPrice As Nullable(Of Decimal)  
        )  
            UpdateFieldValue("ExtendedPrice",  
                Convert.ToDouble(unitPrice.Value) * quantity.Value * (1 - discount.Value))  
        End Sub  
    End Class  
End Namespace
```

End Class  
End Namespace

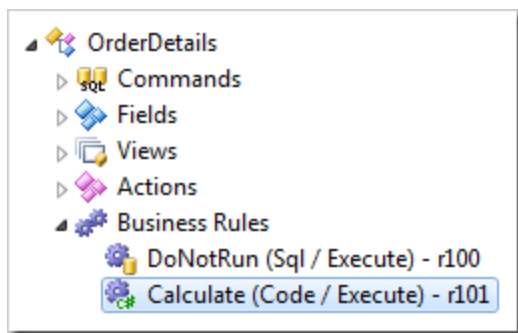
Save the file, and refresh the webpage.

The same behavior as with the SQL business rule will be exhibited every time a context field is changed in the form. The client library will make a short trip to the web server to perform the calculation. The application framework will call the business rules class method which is linked to the business rule defined in *OrderDetails* data controller.

### JavaScript Business Rule

If the values of the base fields are known on the client at the time when a calculation needs to be performed, then the web server round-trip is redundant. JavaScript business rules offer an option to implement complex logic executed by the web browser.

First, disable execution of the previously created C#/Visual Basic business rule. Double-click on *OrderDetails / Business Rules / Calculate* node.

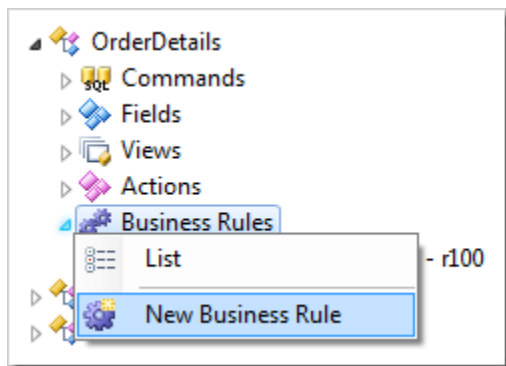


Change the *Command Name* property:

Property	Value
Command Name	DoNotRun

Press *OK* to save. Alternatively, business rule *r101* can be deleted.

Right-click on *OrderDetails / Business Rules* node, and press *New Business Rule*.



Assign these values:

Property	Value
Type	JavaScript
Command Name	Calculate
Phase	Execute
Script	[ExtendedPrice] = [UnitPrice] * [Quantity] * (1 - [Discount]); <code>this.preventDefault();</code>

At runtime, the client library will automatically translate the script into the following JavaScript code:

```
this.updateFieldValue('ExtendedPrice',
    this.selectFieldValue('UnitPrice') *
    this.selectFieldValue('Quantity') *
    (1 - this.selectFieldValue('Discount')));
this.preventDefault();
```

Press *OK* to save. On the toolbar, press *Browse*. Open the form view of an order detail.

Order Details

These are the fields of the order details record that can be edited.

Product:

Rössle Sauerkraut

Discount:

0.25

Unit Price:

\$45.60

Quantity:

15

Extended Price:

\$513.00

Change a value in one of the fields. Press *Tab* to shift focus away from the field. The *Extended Price* will automatically be updated without performing a server request.

Order Details

These are the fields of the order details record that can be edited.

Product:

Rössle Sauerkraut

Discount:

0.3

Unit Price:

\$45.60

Quantity:

15

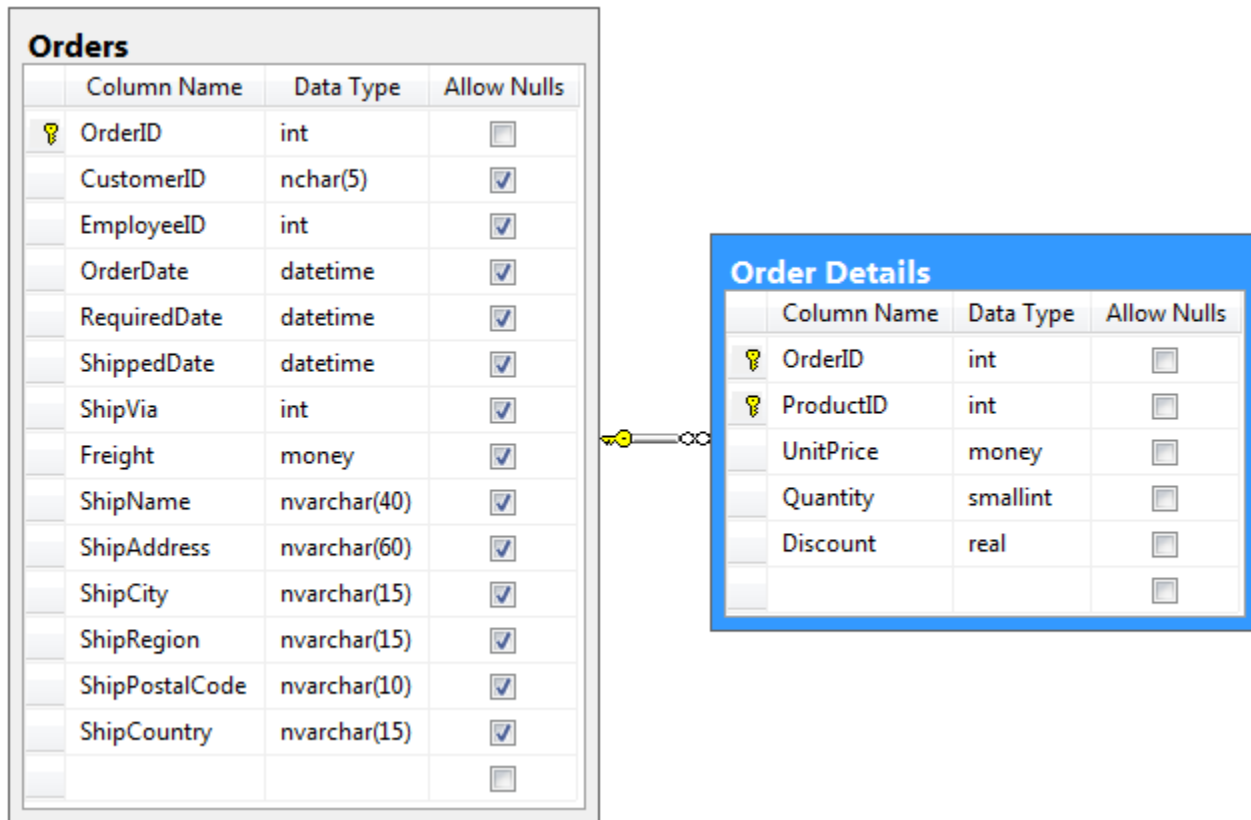
Extended Price:

\$478.80

The call of the method *preventDefault()* will prevent the client library from processing the *Calculate* action on the server.

## Implementing a Custom Action

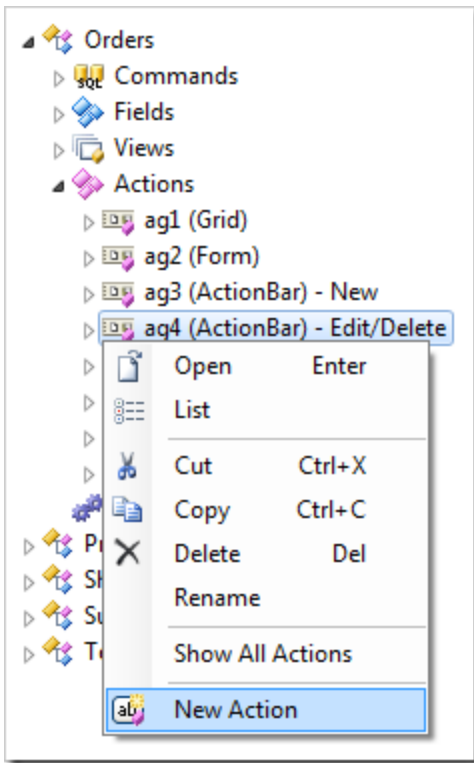
Let's implement a custom action that will apply a discount to all line items of a specific order. Orders are stored in the table *Orders*, and line items are stored in *Order Details*.



### Adding a Custom Action

First, an action needs to be defined in the action group of a data controller.

Start the [Project Designer](#). In the *Project Explorer*, switch to the *Controllers* tab. Right-click on *Orders / Actions / ag4 (ActionBar)* – *Edit/Delete* action group node, and press *New Action*.



Assign the following values:

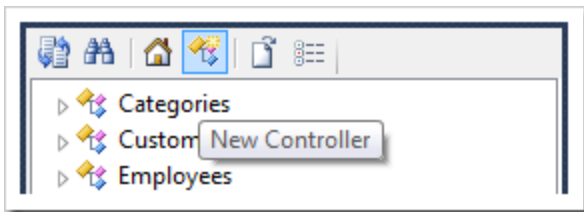
Property	Value
Command Name	Custom
Command Argument	ApplyDiscount
Header Text	Apply a Discount
When Key Selected	Yes
Confirmation	_controller=OrderDiscount _title=Specify a Discount _width=500

Press OK to save the action.

The [Confirmation](#) property references a data controller called "OrderDiscount". This data controller does not exist in the project – we will create this confirmation data controller from scratch and have it configured to collect the *Discount* value from the user. The data controller will not be based on any database table or view.

### Creating the Confirmation Controller

On the *Project Explorer* toolbar, press the *New Controller* icon.

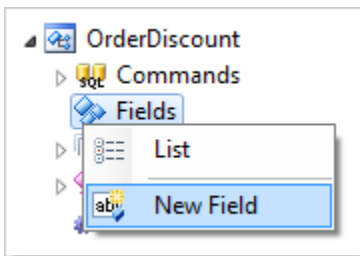


Give this controller a name:

Property	Value
Name	OrderDiscount

Press *OK* to save.

Right-click on *OrderDiscount / Fields* node, and press *New Field*.



Assign the following values:

Property	Value
Name	CustomerCompanyName
Type	String
Length	50
The value of this field is calculated by a business rule expression.	True
Label	Customer Company Name
Values of this field cannot be edited	True

Press *OK* to save.

Add another field with these values:

Property	Value
Name	Discount
Type	Single
Label	Discount

Save the field.

Generate the app and select an order. On the action bar, press the *Apply a Discount* action.

This is a list of orders.

Quick Find

New Orders

Edit

Delete

Apply a Discount

Actions

Report

View: Orders

Customer Company Name	Employee Last Name	Order Date	Required Date	Shipped Date	Ship Via Company Name	Freight	Ship Name	Ship Address	Ship Ci
Vins et alcools Chevalier	Buchanan	7/4/1996	8/1/1996	7/16/1996	Federal Shipping	\$32.38	Vins et alcools Chevalier	59 rue de l'Abbaye	Reims

A modal form with two empty fields, *Customer Company Name* and *Discount*, will be displayed.

Specify a Discount

Customer Company Name

N/A

Discount \*

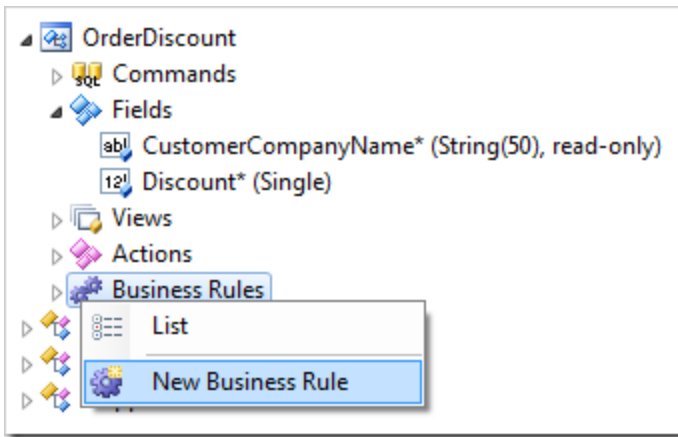
\* - indicates a required field

OK

Cancel

Let's populate *Customer Company Name* with the name of the company associated with the selected order and initialize *Discount* with the average discount of the order details.

In the *Project Explorer*, right-click on *OrderDiscount / Business Rules* node, and press *New Business Rule*.



Assign the following values:

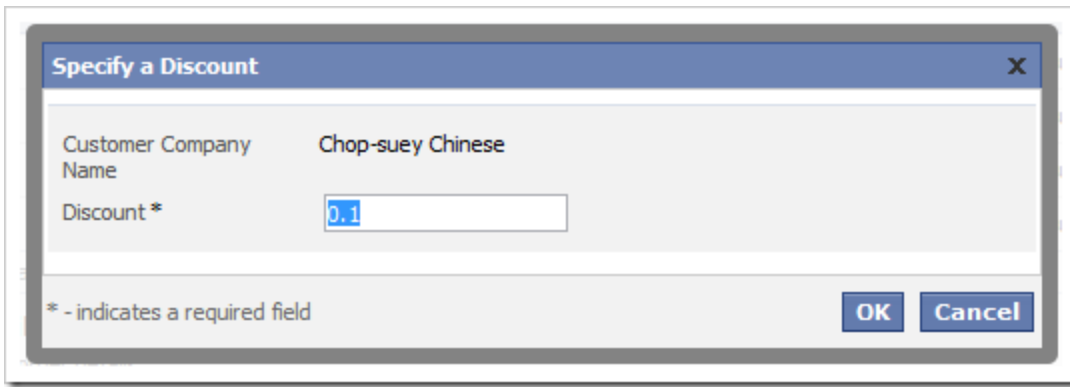
Property	Value
Type	SQL
Command Name	New
Phase	Execute
Script	<pre>set @CustomerCompanyName = @Context_CustomerCompanyName  select @Discount = avg(Discount) from "Order Details" where OrderID = @Context_OrderID</pre>

Press OK to save the business rule.

The parameter `@CustomerCompanyName` refers to the *CustomerCompanyName* field of the confirmation data controller.

The parameter `@Context_CustomerCompanyName` refers to the *CustomerCompanyName* field of the *Orders* data controller.

The application framework will pass the script for execution to the database engine when the action *Custom / ApplyDiscount* is activated by the user. The result is shown in the next screenshot.



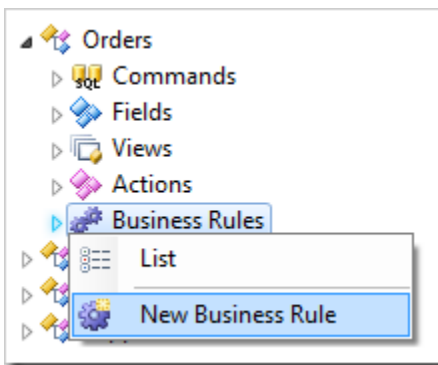
A dialog box titled "Specify a Discount" with a close button (X) in the top right corner. It contains two input fields: "Customer Company Name" with the value "Chop-suey Chinese" and "Discount \*" with the value "0.1". The "Discount \*" field has a blue selection highlight. At the bottom left, there is a note: "\* - indicates a required field". At the bottom right, there are "OK" and "Cancel" buttons.

If the user clicks *OK*, nothing will happen. The application does not know what to do with a “Custom” action with argument of “ApplyDiscount”.

### Creating an SQL Business Rule to Handle the Action

Now that an action and controller are in place to capture the user input, a business rule needs to be created to apply the specified discount to all *OrderDetails* data rows associated with the selected order.

Right-click on *Orders / Business Rules* node, and press *New Business Rule*.



Use these values:

Property	Value
Type	SQL
Command Name	Custom
Command Argument	ApplyDiscount
Phase	Execute
Script	<pre>-- apply discount to order details update "Order Details" set Discount = @Parameters_Discount where OrderID = @OrderID</pre>

```
-- force refresh of child views
set @Result_RefreshChildren = 1
```

Press *OK* to save.

The first statement in the SQL script will update [Order Details].[Discount] column where the *OrderID* matches the selected order. The value of the discount is referenced by *@Parameters\_Discount* parameter.

The second statement instructs the client library to refresh the child data views of the master data view *Orders*. This will cause *Order Details* data view to reflect the updated discount.

On the *Project Designer* toolbar, press *Browse*. Navigate to the *Orders* page and select an order. A list of related order details will be displayed below. Take note of the discounts of the order details.

This is a list of orders.

Quick Find New Orders Edit Delete Apply a Discount Actions Report View: **Orders**

A filter has been applied. Order Date equals 7/11/1996.

Customer Company Name	Employee Last Name	Order Date	Required Date	Shipped Date	Ship Via Company Name	Freight	Ship Name	Ship Address	Ship City
Chop-suey Chinese	Buchanan	7/11/1996	8/8/1996	7/23/1996	United Package	\$22.98	Chop-suey Chinese	Hauptstr. 31	Bern

Showing 1-1 of 1 items

**Order Details**

This is a list of order details.

Quick Find New Order Details Actions Report View: **Order Details**

Product Name	Unit Price	Quantity	Discount	Order Customer Company Name	Order Employee Last Name	Order Ship Via Company Name	Product Category Name	Product Supplier Company Name	Extended Price
Guaraná Fantástica	\$3.60	15	0.15	Chop-suey Chinese	Buchanan	United Package	Beverages	Refrescos Americanas LTDA	\$45.90
Pâté chinois	\$19.20	21	0.15	Chop-suey Chinese	Buchanan	United Package	Meat/Poultry	Ma Maison	\$342.72
Longlife Tofu	\$8.00	21	0	Chop-suey Chinese	Buchanan	United Package	Produce	Tokyo Traders	\$168.00

Showing 1-3 of 3 items

On the action bar of *Orders* grid view, press *Apply a Discount*. The *confirmation modal popup* will appear, displaying the current *Customer Company Name* and the average discount. Enter a discount of ".25".

Specify a Discount

Customer Company Name

Chop-suey Chinese

Discount \*

.25

\* - indicates a required field

OK

Cancel

Press OK, and the specified discount will be applied to all records in *Order Details* table that belong to the selected order.

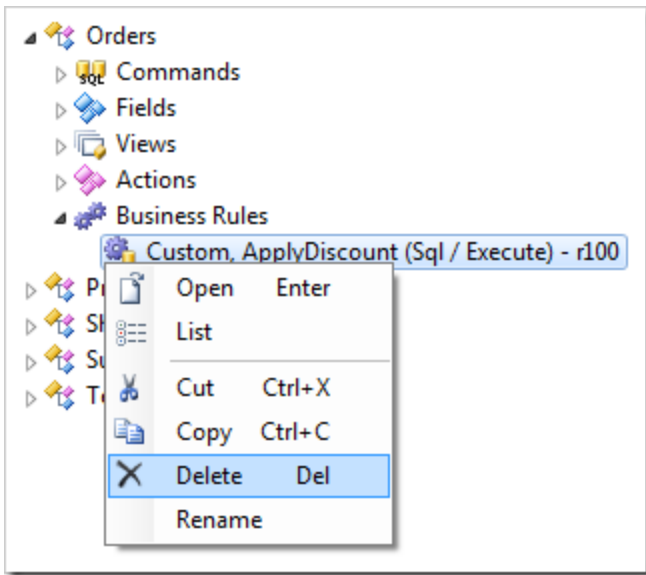
Order Details									
This is a list of order details.									
Quick Find		New Order Details	Actions	Report	View: Order Details				
Product Name	Unit Price	Quantity	Discount	Order Customer Company Name	Order Employee Last Name	Order Ship Via Company Name	Product Category Name	Product Supplier Company Name	Extended Price
Guaraná Fantástica	\$3.60	15	0.25	Chop-suey Chinese	Buchanan	United Package	Beverages	Refrescos Americanas LTDA	\$40.50
Pâté chinois	\$19.20	21	0.25	Chop-suey Chinese	Buchanan	United Package	Meat/Poultry	Ma Maison	\$302.40
Longlife Tofu	\$8.00	21	0.25	Chop-suey Chinese	Buchanan	United Package	Produce	Tokyo Traders	\$126.00
Showing 1-3 of 3 items									

## Creating “Code” Business Rule to Handle the Action

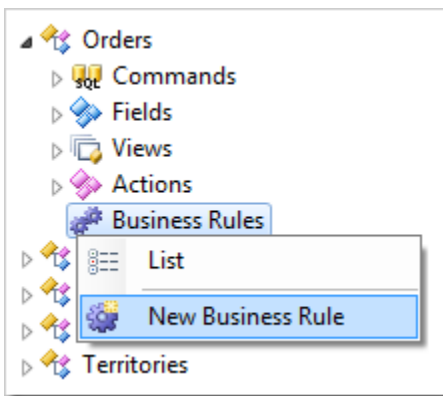
Instead of using SQL, you may also create a code business rule written in C# or Visual Basic to handle the calculation.

If you have implemented the previous SQL business rule, you will need to delete it.

Right-click on *Orders / Business Rules / Custom, ApplyDiscount (Sql / Execute)* - r100 business rule node and press *Delete*.



Right-click on *Orders / Business Rules* node and press *New Business Rule*.



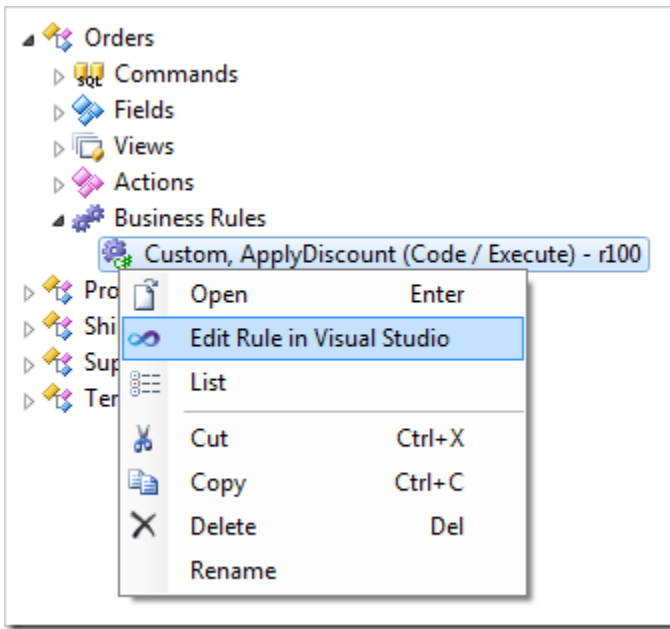
Assign these values:

Property	Value
Type	C# / Visual Basic
Command Name	Custom
Command Argument	ApplyDiscount
Phase	Execute

Save the rule. "Code" business rules do not have a script stored in the data controller definition file. A code file must be created in the project. The application generator will create an initial "empty" business rule code file as soon as the project is generated.

On the *Project Designer* toolbar, press *Generate*.

When complete, right-click on *Orders / Business Rules / Custom, ApplyDiscount (Code / Execute) – r100* node and press *Edit Rule in Visual Studio*.



The file will be opened in *Visual Studio*. The generator has created a template for the business rule. Replace the existing code with the following:

C#:

```
using System;
using MyCompany.Data;

namespace MyCompany.Rules
{
    public partial class OrdersBusinessRules : MyCompany.Data.BusinessRules
    {
        /// <summary>
        /// This method will execute in any view for an action
        /// with a command name that matches "Custom" and argument that matches "ApplyDiscount".
        /// </summary>
        [Rule("r100")]
        public void r100Implementation(
            int? orderID,
            string customerID,
            string customerCompanyName,
            int? employeeID,
            string employeeLastName,
            DateTime? orderDate,
            DateTime? requiredDate,
            DateTime? shippedDate,
            int? shipVia,
            string shipViaCompanyName,
            decimal? freight,
            string shipName,
            string shipAddress,
            string shipCity,
            string shipRegion,
            string shipPostalCode,
```

```

        string shipCountry)
    {
        // This is the placeholder for method implementation.
        using (SqlText applyDiscount = new SqlText(
            "update [Order Details] " +
            "set Discount = @Discount " +
            "where OrderID = @OrderID"))
        {
            applyDiscount.AddParameter("@Discount", SelectFieldValue("Parameters_Discount"));
            applyDiscount.AddParameter("@OrderID", orderID);
            applyDiscount.ExecuteNonQuery();
        }
        Result.RefreshChildren();
    }
}
}

```

Visual Basic:

```

Imports MyCompany.Data
Imports System
Imports MyCompany.Rules

```

```

Namespace MyCompany.Rules

```

```

    Partial Public Class OrdersBusinessRules
        Inherits MyCompany.Data.BusinessRules

```

```

        ''' <summary>
        ''' This method will execute in any view for an action
        ''' with a command name that matches "Custom" and argument that matches "ApplyDiscount".
        ''' </summary>
        <Rule("r100")> _
        Public Sub r100Implementation( _
            ByVal orderID As Nullable(Of Integer), _
            ByVal customerID As String, _
            ByVal customerCompanyName As String, _
            ByVal employeeID As Nullable(Of Integer), _
            ByVal employeeLastName As String, _
            ByVal orderDate As Nullable(Of DateTime), _
            ByVal requiredDate As Nullable(Of DateTime), _
            ByVal shippedDate As Nullable(Of DateTime), _
            ByVal shipVia As Nullable(Of Integer), _
            ByVal shipViaCompanyName As String, _
            ByVal freight As Nullable(Of Decimal), _
            ByVal shipName As String, _
            ByVal shipAddress As String, _
            ByVal shipCity As String, _
            ByVal shipRegion As String, _
            ByVal shipPostalCode As String, _
            ByVal shipCountry As String)
            'This is the placeholder for method implementation.
            Using applyDiscount As SqlText = New SqlText(
                "update [Order Details] " +
                "set Discount = @Discount " +
                "where OrderID = @OrderID"
            )
                applyDiscount.AddParameter("@Discount", SelectFieldValue("Parameters_Discount"))
                applyDiscount.AddParameter("@OrderID", orderID)
                applyDiscount.ExecuteNonQuery()
            End Using
        End Sub
    End Class

```

```
        End Using
        Result.RefreshChildren()
    End Sub
End Class
End Namespace
```

Save the file, and refresh the web page. The *Assign a Discount* action will function in exactly the same way as the version with the SQL business rule.