2010

COOKBOOK

Row-Level Security

Table of Contents

Role vs. Row-Level Security

Let’s compare role and row-level security. Roles are controlling access to shared resources of your application, such as pages, data fields, and actions. Row-level security further restricts access to individual items (rows) presented in the application data views.

Role

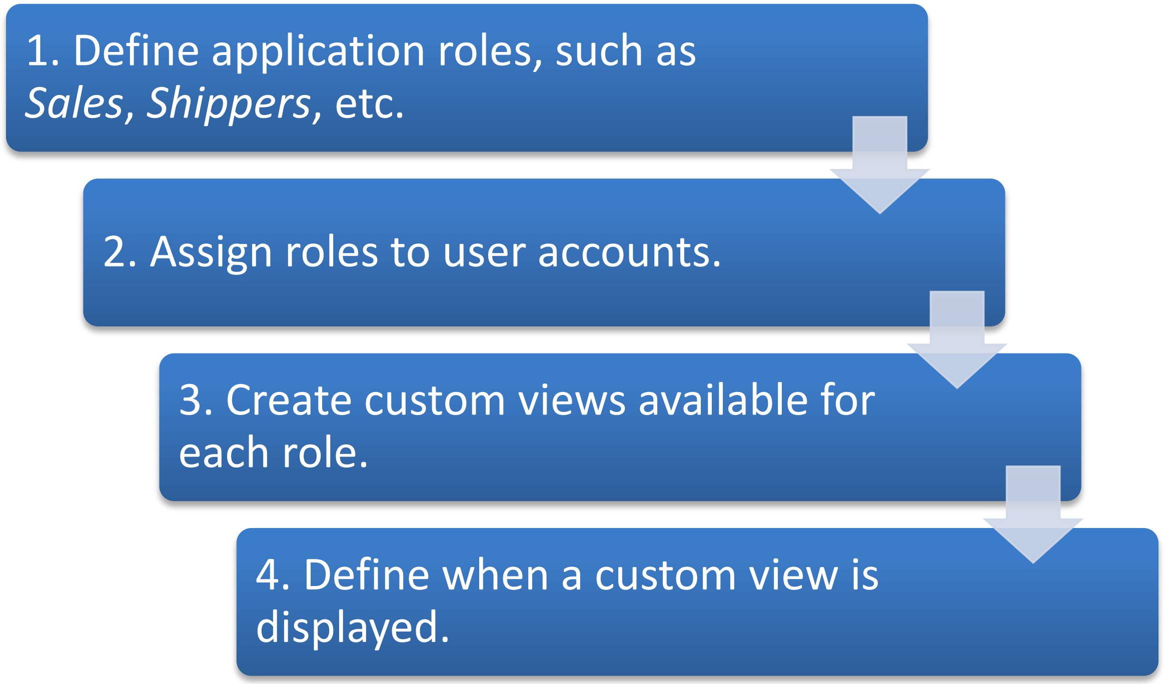
Roles are equivalent to vertical restrictions in your application. Let’s consider a few examples of vertical restrictions. Sales representatives are allowed to see a page with a list of *Orders*. Users from *Accounting* are allowed to see the *Commission* data field in the list of orders. Users with role *Admin* can access the *Membership Manager* and can create, edit, and delete any user.

Row-Level

Row-level security is equivalent to horizontal restrictions, applied on top of vertical restrictions. User *andrew.fuller* can only view his respective orders that are marked with his last name. User *federal.shipping* can see all orders that are “not shipped” or “shipped this or last week” if orders are assigned to shipper *Federal Shipping*.

Steps to Implementing Row-Level Security

How do you go about implementing row-level security?



1. Define application roles, such as *Sales*, *Shippers*, etc.

2. Assign roles to user accounts.

3. Create custom views available for each role.

4. Define when a custom view is displayed.

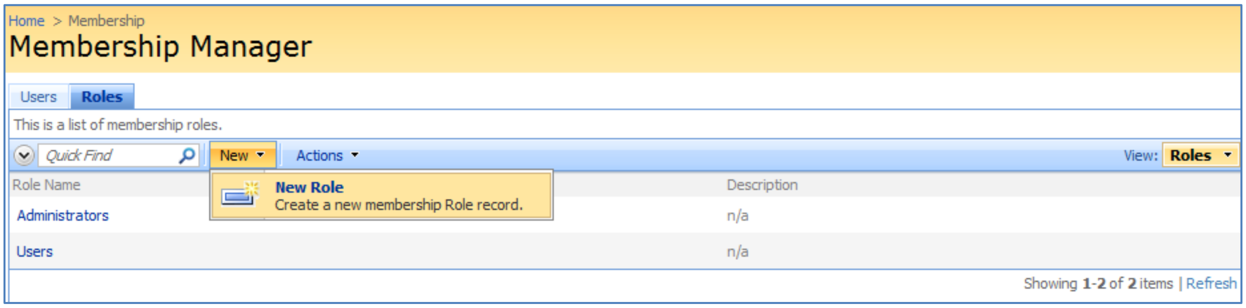
These four steps will provide row-level security. Step five will be to impose vertical security on the data by defining row-level filters for each custom view.

Setting up the Application

Let's consider a real world example. We'll use *Northwind* database. We'll set up row-level security with role *Sales*, *Shippers*, and *Customers*.

Creating Roles

Log in to the application as *Administrator*. Navigate to the *Membership Manager*, and switch to the roles tab. On the action bar, press the *New* button, and then *New Role*.


Home > Membership

Membership Manager

Users | Roles

This is a list of membership roles.

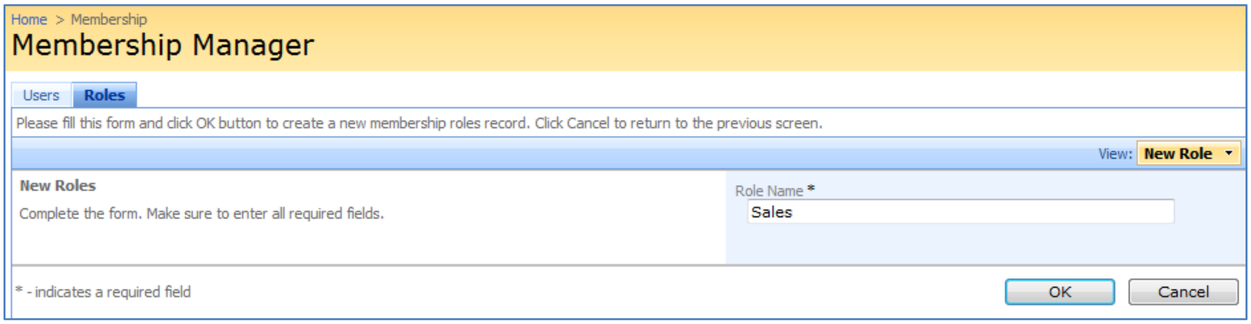
Quick Find | New ▾ | Actions ▾ | View: Roles ▾

New Role
Create a new membership Role record.

Role Name	Description
Administrators	n/a
Users	n/a

Showing 1-2 of 2 items | Refresh

You will be taken to the role creation screen. In the *Role Name* field, type *Sales*. Press *Ok* to save the new role.


Home > Membership

Membership Manager

Users | Roles

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

View: New Role ▾

New Roles

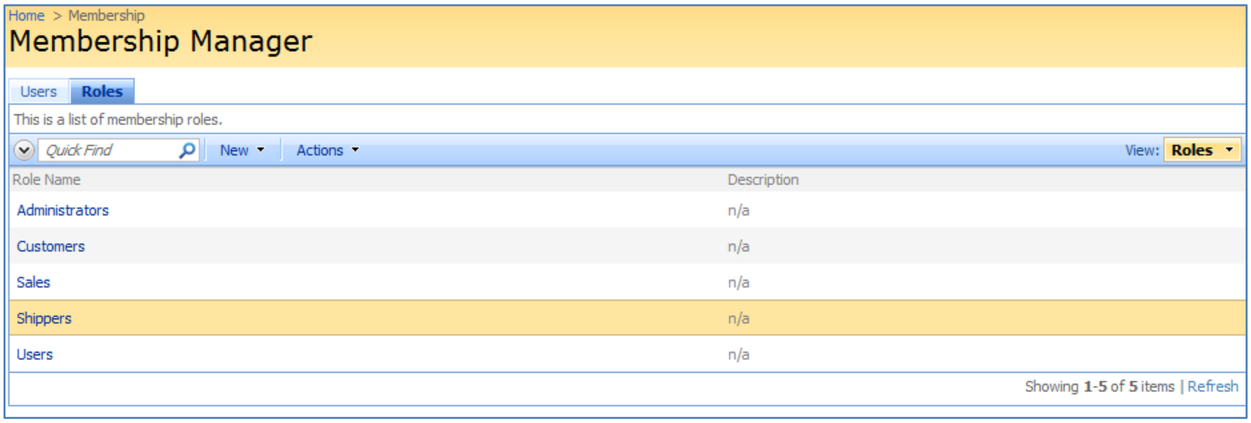
Complete the form. Make sure to enter all required fields.

Role Name *: Sales

* - indicates a required field

OK | Cancel

Create two more roles. One will have the name of *Shippers*, and the other will be called *Customers*. When finished, the list of roles should look like the image below.


Home > Membership

Membership Manager

Users | Roles

This is a list of membership roles.

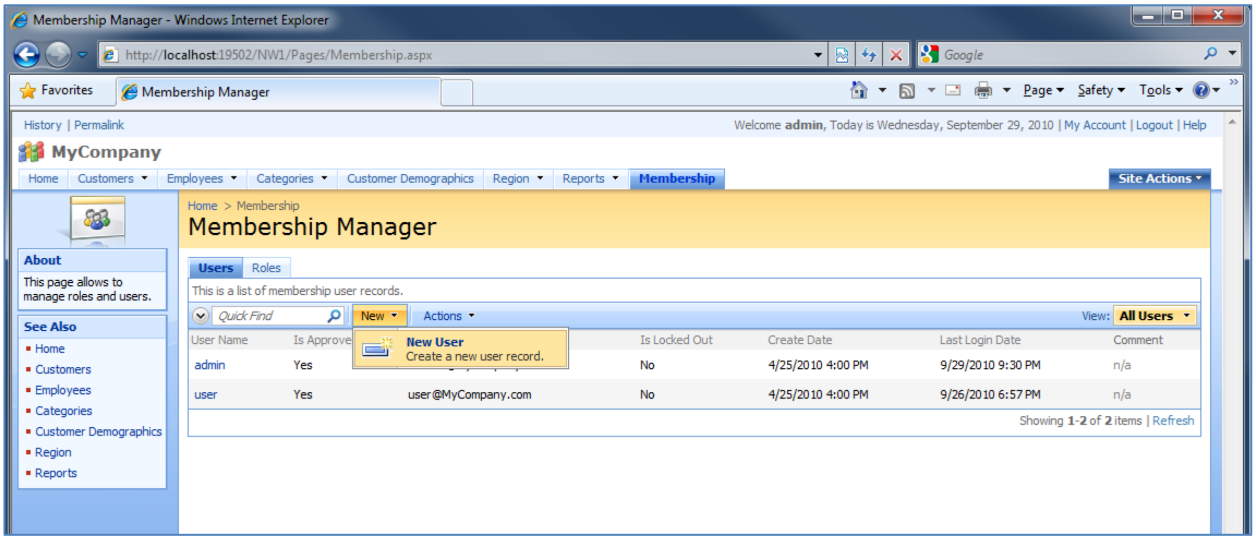
Quick Find | New ▾ | Actions ▾ | View: Roles ▾

Role Name	Description
Administrators	n/a
Customers	n/a
Sales	n/a
Shippers	n/a
Users	n/a

Showing 1-5 of 5 items | Refresh

Creating User Accounts

We need to set up several user accounts. In the *Membership Manager*, switch to *Users* tab. On the action bar, click on *New*, and press *New User*.



Home > Membership

Membership Manager

This page allows to manage roles and users.

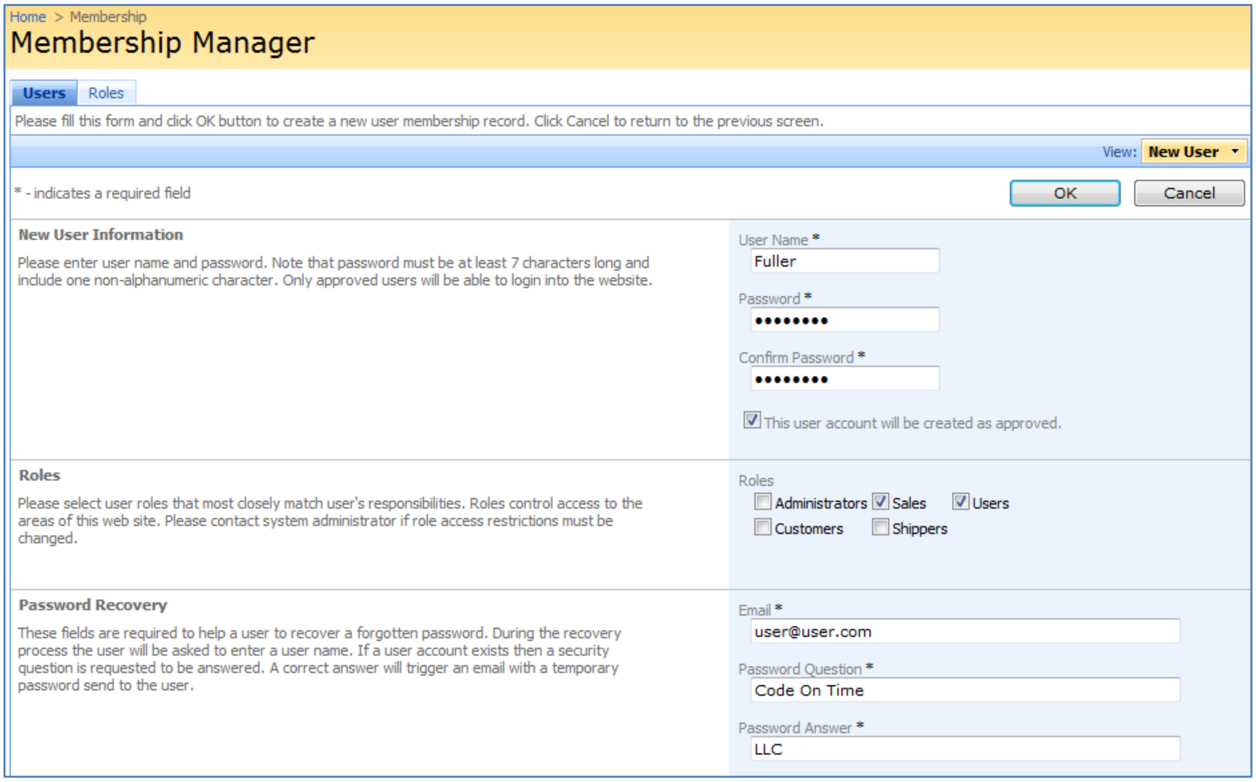
This is a list of membership user records.

New User — Create a new user record.

User Name	Is Approve		Is Locked Out	Create Date	Last Login Date	Comment
admin	Yes		No	4/25/2010 4:00 PM	9/29/2010 9:30 PM	n/a
user	Yes	user@MyCompany.com	No	4/25/2010 4:00 PM	9/26/2010 6:57 PM	n/a

Showing 1-2 of 2 items | Refresh

Create the user *Fuller*. The password will be “user123%”, which fulfills standard restrictions on the password imposed by ASP.NET. Give Mr. Fuller the roles of *Users* and *Sales*, and fill in the password recovery information.



Home > Membership

Membership Manager

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

* - indicates a required field

New User Information

Please enter user name and password. Note that password must be at least 7 characters long and include one non-alphanumeric character. Only approved users will be able to login into the website.

User Name *: Fuller

Password *: ••••••••

Confirm Password *: ••••••••

☑ This user account will be created as approved.

Roles

Please select user roles that most closely match user's responsibilities. Roles control access to the areas of this web site. Please contact system administrator if role access restrictions must be changed.

Roles: ☐ Administrators ☑ Sales ☑ Users ☐ Customers ☐ Shippers

Password Recovery

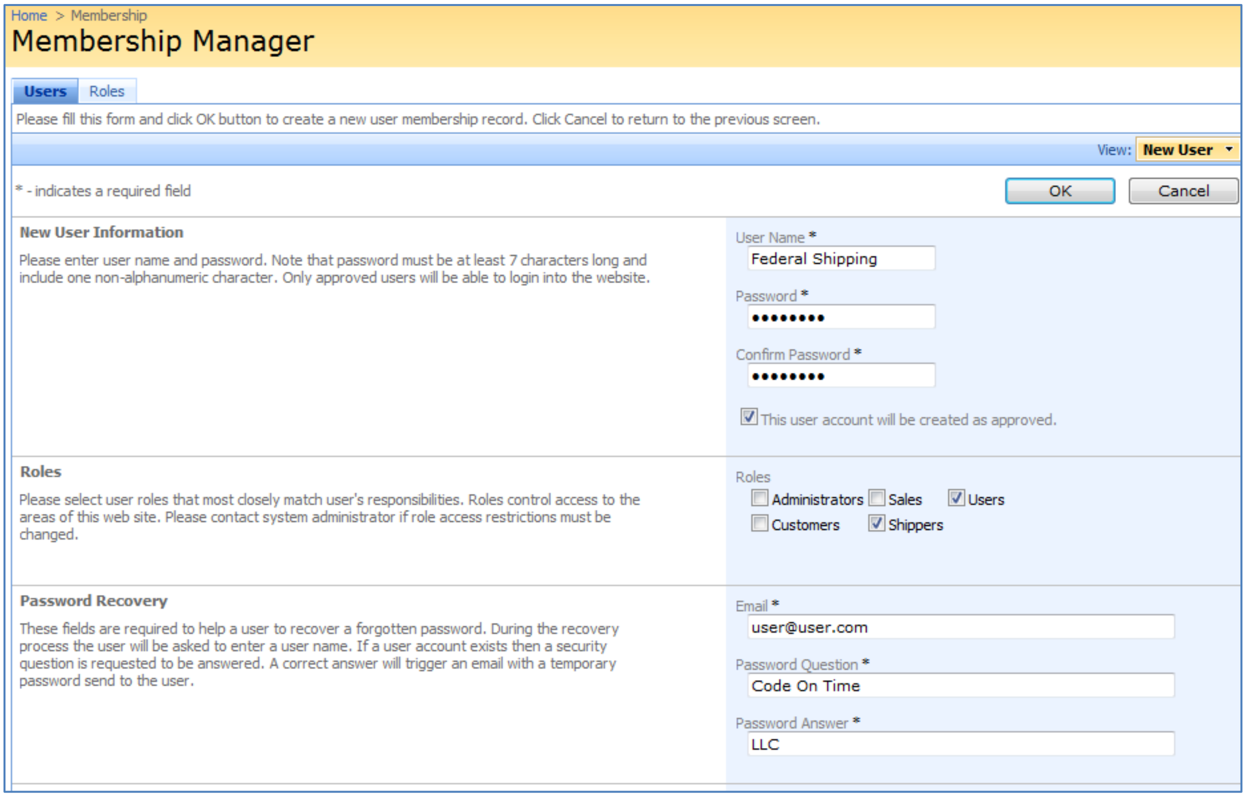
These fields are required to help a user to recover a forgotten password. During the recovery process the user will be asked to enter a user name. If a user account exists then a security question is requested to be answered. A correct answer will trigger an email with a temporary password send to the user.

Email *: user@user.com

Password Question *: Code On Time

Password Answer *: LLC

Next, create the account *Federal Shipping*. It will have the same password. This account will have the roles *Users* and *Shippers*. Don't forget to fill in the password recovery information.



Home > Membership

Membership Manager

Users | Roles

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

View: New User

* - indicates a required field

OK | Cancel

New User Information

Please enter user name and password. Note that password must be at least 7 characters long and include one non-alphanumeric character. Only approved users will be able to login into the website.

User Name *: Federal Shipping

Password *: ••••••••

Confirm Password *: ••••••••

☑ This user account will be created as approved.

Roles

Please select user roles that most closely match user's responsibilities. Roles control access to the areas of this web site. Please contact system administrator if role access restrictions must be changed.

Roles: □ Administrators □ Sales ☑ Users □ Customers ☑ Shippers

Password Recovery

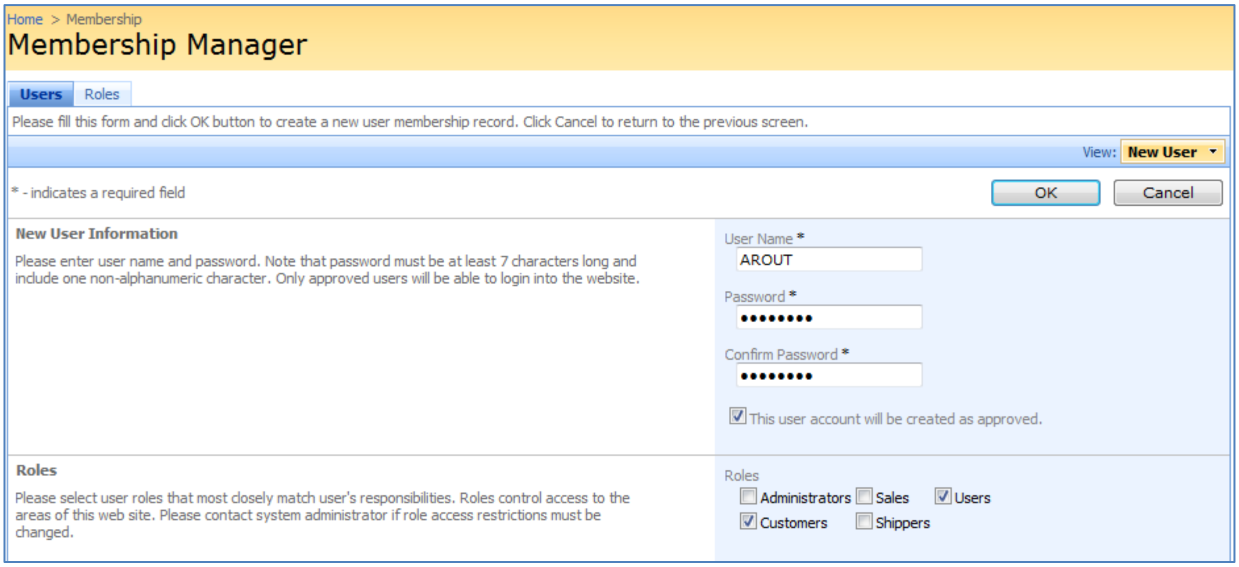
These fields are required to help a user to recover a forgotten password. During the recovery process the user will be asked to enter a user name. If a user account exists then a security question is requested to be answered. A correct answer will trigger an email with a temporary password send to the user.

Email *: user@user.com

Password Question *: Code On Time

Password Answer *: LLC

AROUT will be the last account created. *AROUT* is the *customer ID* of the company *Around The Horn*. This account will have the roles *Users* and *Customers*.



Home > Membership

Membership Manager

Users | Roles

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

View: New User

* - indicates a required field

OK | Cancel

New User Information

Please enter user name and password. Note that password must be at least 7 characters long and include one non-alphanumeric character. Only approved users will be able to login into the website.

User Name *: AROUT

Password *: ••••••••

Confirm Password *: ••••••••

☑ This user account will be created as approved.

Roles

Please select user roles that most closely match user's responsibilities. Roles control access to the areas of this web site. Please contact system administrator if role access restrictions must be changed.

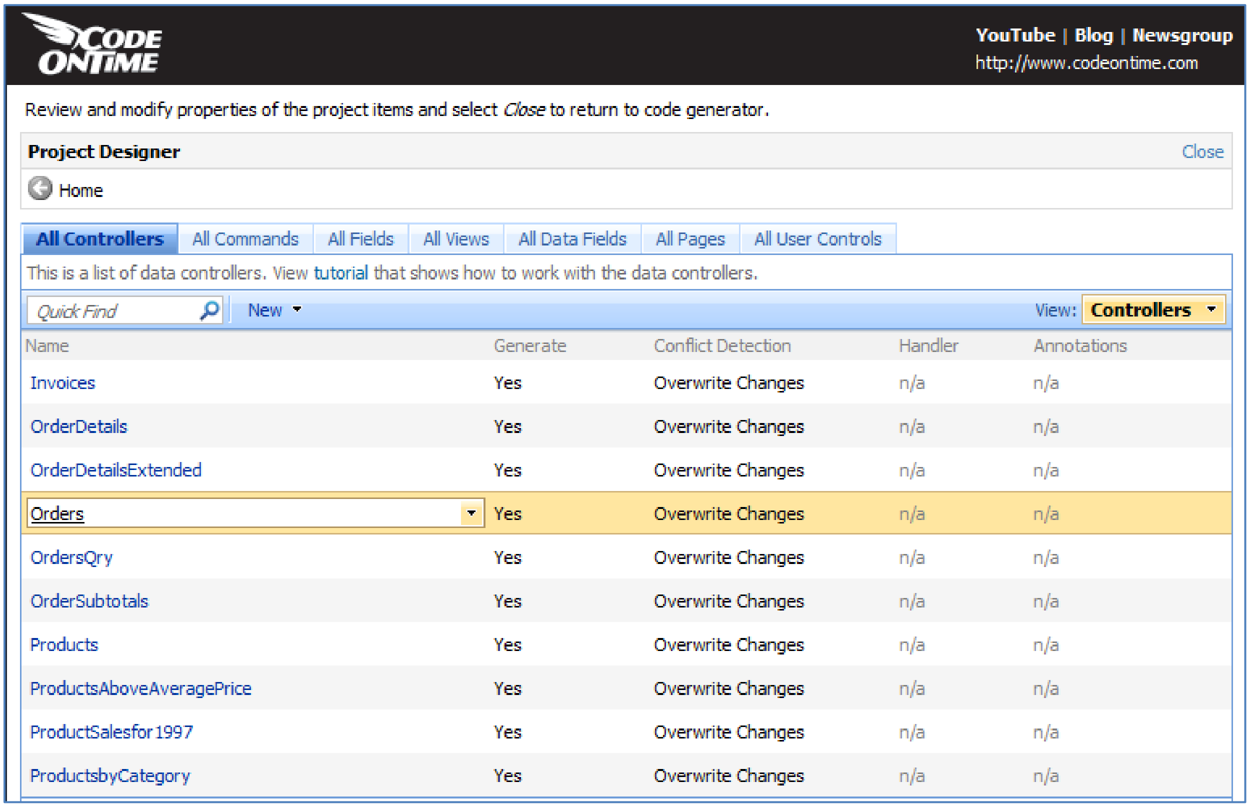
Roles: □ Administrators □ Sales ☑ Users ☑ Customers □ Shippers

You have two options on how you want to set up your row-level security. The next section will explain how to implement using the graphical interface of Code On Time Designer. The section after this explains how to implement using the Designer Spreadsheet (requires Microsoft Excel). Both methods will lead to the same result.

Row-Level Security Using the Designer

Define Role-Specific Views

We need to define role-specific views. Run *Code On Time Generator* and select the project from the project list. Click on the *Design* button. Select *Orders* data controller from the controller list.

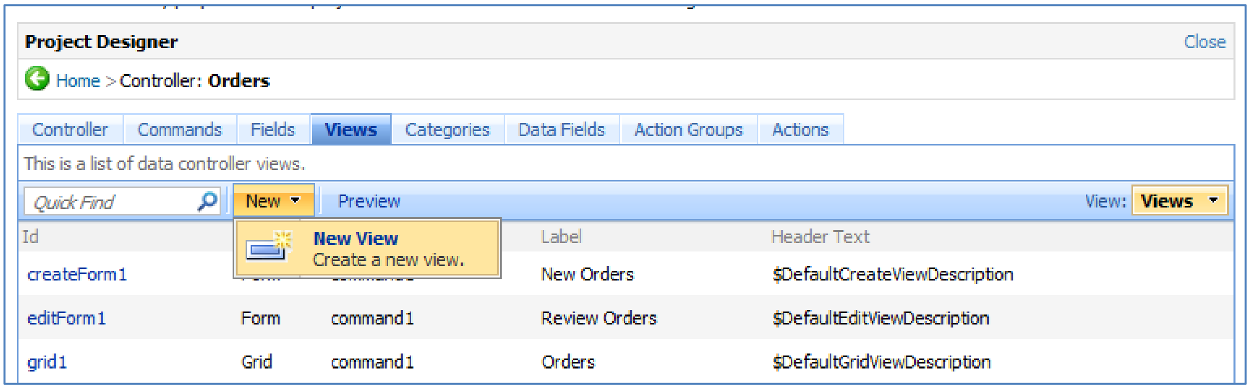


Review and modify properties of the project items and select *Close* to return to code generator.

This is a list of data controllers. View tutorial that shows how to work with the data controllers.

Name	Generate	Conflict Detection	Handler	Annotations
Invoices	Yes	Overwrite Changes	n/a	n/a
OrderDetails	Yes	Overwrite Changes	n/a	n/a
OrderDetailsExtended	Yes	Overwrite Changes	n/a	n/a
Orders	Yes	Overwrite Changes	n/a	n/a
OrdersQry	Yes	Overwrite Changes	n/a	n/a
OrderSubtotals	Yes	Overwrite Changes	n/a	n/a
Products	Yes	Overwrite Changes	n/a	n/a
ProductsAboveAveragePrice	Yes	Overwrite Changes	n/a	n/a
ProductSalesfor1997	Yes	Overwrite Changes	n/a	n/a
ProductsbyCategory	Yes	Overwrite Changes	n/a	n/a

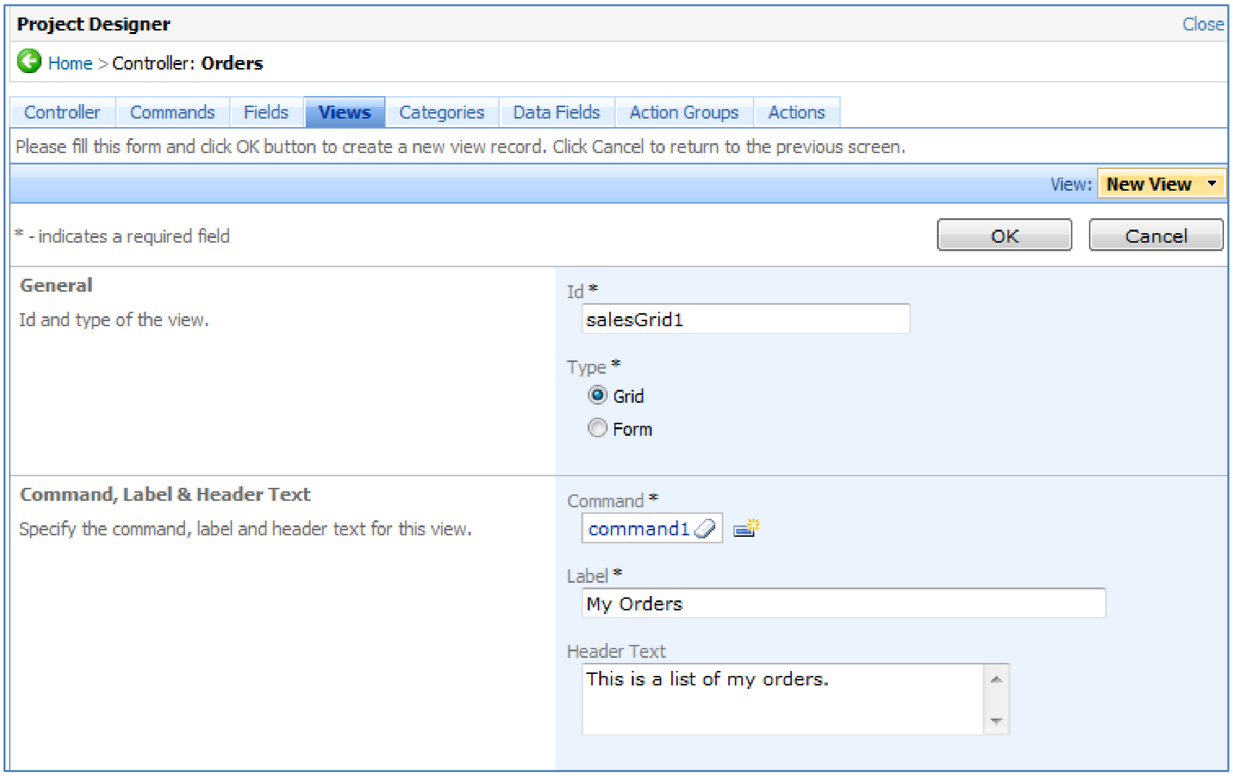
Click on the *Views* tab at the top of the page. You should be on the page as shown below. On the action bar, press *New*, and click on *New View*.



This is a list of data controller views.

Id			Label	Header Text
createForm1			New Orders	$DefaultCreateViewDescription
editForm1	Form	command1	Review Orders	$DefaultEditViewDescription
grid1	Grid	command1	Orders	$DefaultGridViewDescription

Create a new *View* with the *Id* of *salesGrid1*. In the *Command* field, click on *(select)* and choose *command1* from the list. For *Label* field, type "My Orders". The *Header Text* will be "This is a list of my orders".



Project Designer Close

Home > Controller: **Orders**

Controller | Commands | Fields | **Views** | Categories | Data Fields | Action Groups | Actions

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

View: **New View**

* - indicates a required field

General

Id and type of the view.

Id *: salesGrid1

Type *

● Grid

○ Form

Command, Label & Header Text

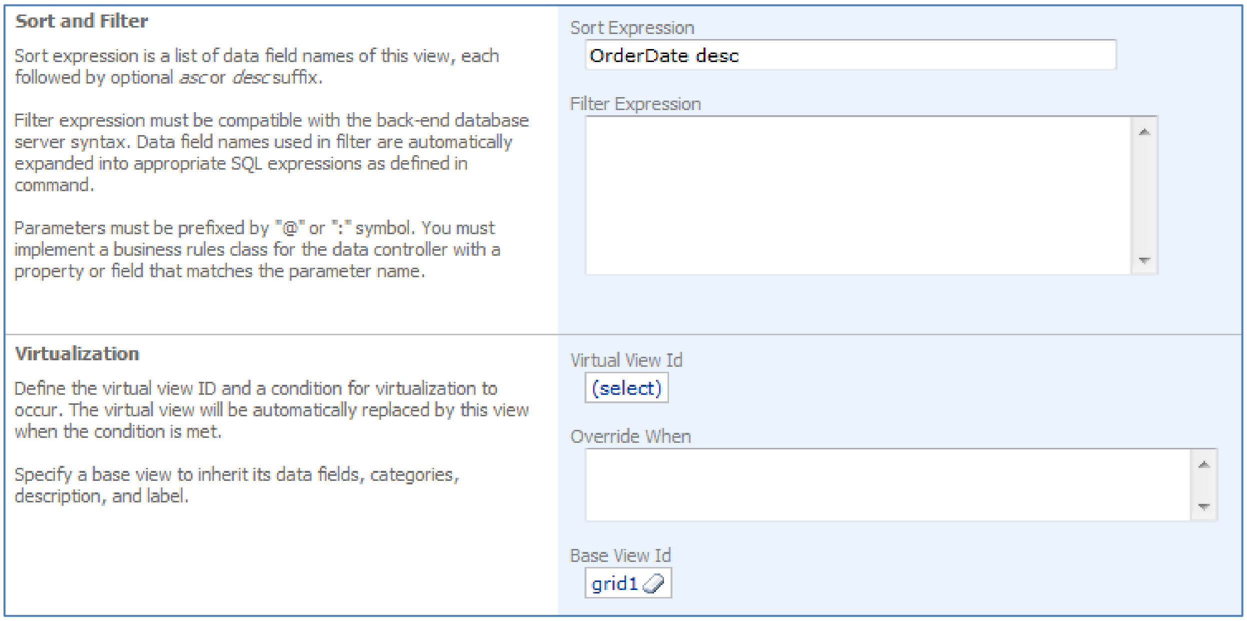
Specify the command, label and header text for this view.

Command *: command1

Label *: My Orders

Header Text: This is a list of my orders.

Sort the orders by *OrderDate* in descending order by typing in "OrderDate desc" in the *Sort Expression* field. For *Base View ID*, select *grid1*. This will copy all data fields from *grid1*. Press Ok to save the view.



Sort and Filter

Sort expression is a list of data field names of this view, each followed by optional *asc* or *desc* suffix.

Filter expression must be compatible with the back-end database server syntax. Data field names used in filter are automatically expanded into appropriate SQL expressions as defined in command.

Parameters must be prefixed by "@" or ":" symbol. You must implement a business rules class for the data controller with a property or field that matches the parameter name.

Sort Expression: OrderDate desc

Filter Expression:

Virtualization

Define the virtual view ID and a condition for virtualization to occur. The virtual view will be automatically replaced by this view when the condition is met.

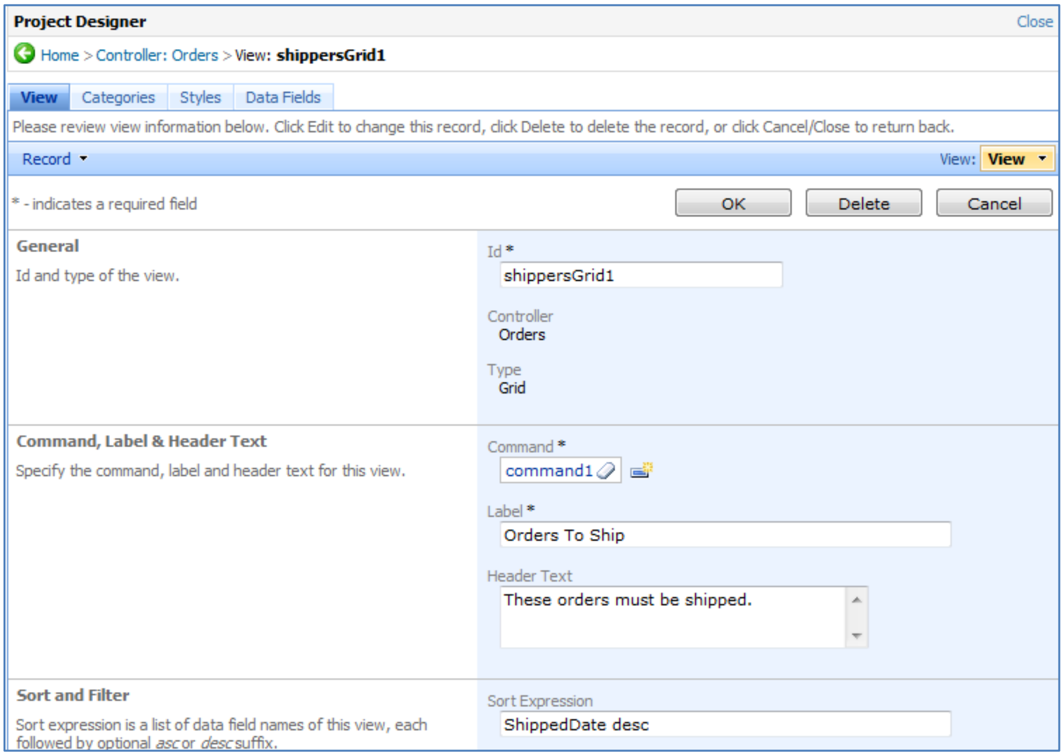
Specify a base view to inherit its data fields, categories, description, and label.

Virtual View Id: (select)

Override When:

Base View Id: grid1

Next, create a *View* with the *ID shippersGrid1*, with *Command* of *command1*. The *Label* will be "Orders To Ship", and the *Header Text* will be "These orders must be shipped". For *Sort Expression*, sort by *ShippedDate* in descending order. *Base View ID* will be *grid1*. Press Ok to save.



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

* - indicates a required field

General

Id and type of the view.

Id *: shippersGrid1

Controller: Orders

Type: Grid

Command, Label & Header Text

Specify the command, label and header text for this view.

Command *: command1

Label *: Orders To Ship

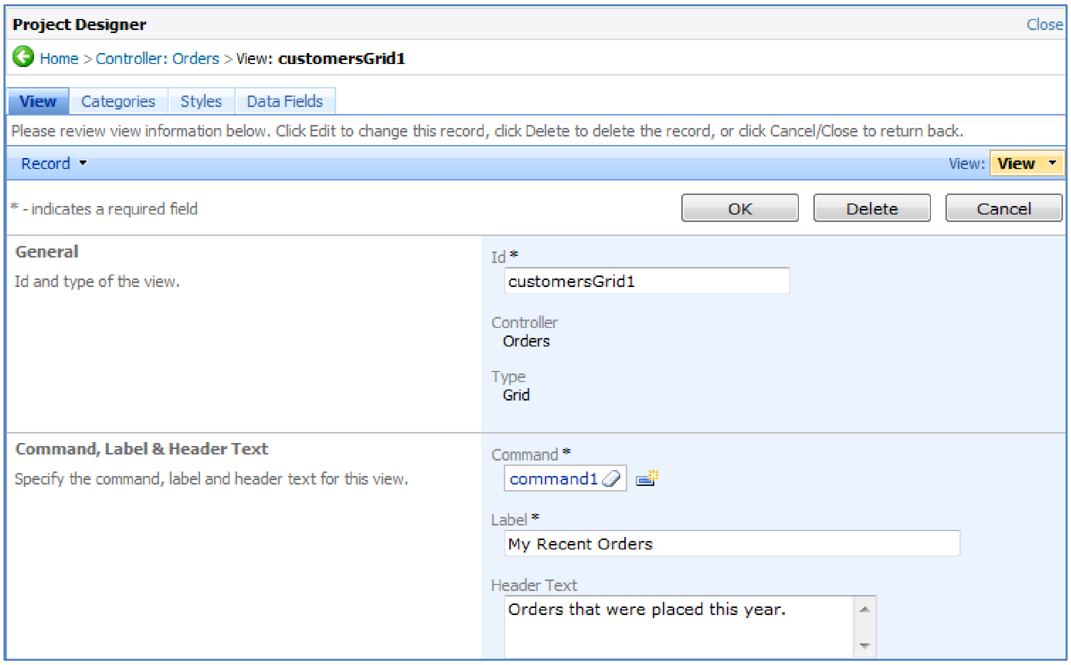
Header Text: These orders must be shipped.

Sort and Filter

Sort expression is a list of data field names of this view, each followed by optional *asc* or *desc* suffix.

Sort Expression: ShippedDate desc

The last view will go by the *ID* of *customersGrid1*. *Command* will be *command1*, *Label* will be "My Recent Orders", and *Header Text* will be "Orders that were placed this year".



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

* - indicates a required field

General

Id and type of the view.

Id *: customersGrid1

Controller: Orders

Type: Grid

Command, Label & Header Text

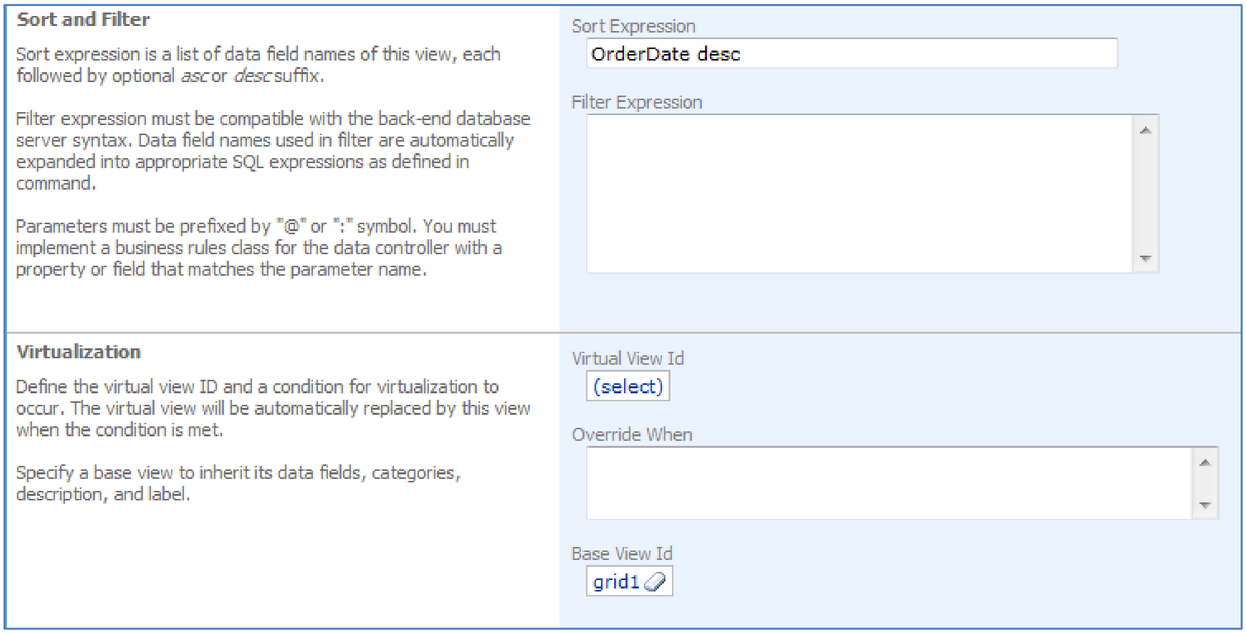
Specify the command, label and header text for this view.

Command *: command1

Label *: My Recent Orders

Header Text: Orders that were placed this year.

Sort all orders by *OrderDate* in descending order by typing in “OrderDate desc” in *Sort Expression*. *Base View ID* will be *grid1* as well.



Sort and Filter

Sort expression is a list of data field names of this view, each followed by optional *asc* or *desc* suffix.

Filter expression must be compatible with the back-end database server syntax. Data field names used in filter are automatically expanded into appropriate SQL expressions as defined in command.

Parameters must be prefixed by "@" or ":" symbol. You must implement a business rules class for the data controller with a property or field that matches the parameter name.

Sort Expression: OrderDate desc

Filter Expression:

Virtualization

Define the virtual view ID and a condition for virtualization to occur. The virtual view will be automatically replaced by this view when the condition is met.

Specify a base view to inherit its data fields, categories, description, and label.

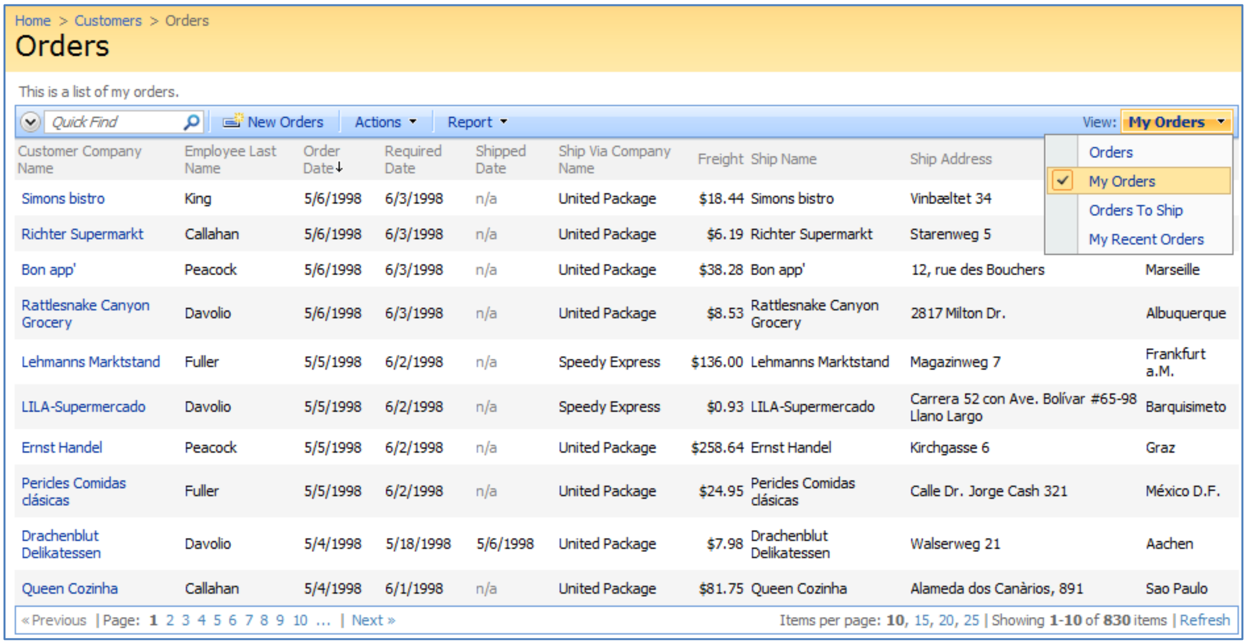
Virtual View Id: (select)

Override When:

Base View Id: grid1

Views in Action

Let’s take a look at these views in action. Regenerate the application using *Code On Time Generator*. When it finishes, navigate to the *Orders* page. If you select the *My Orders* view from the *View Selector*, you can see the custom description at the top, and that the *Order Date* field has been sorted in descending order. This list of orders is designed for salespeople.



Home > Customers > Orders

Orders

This is a list of my orders.

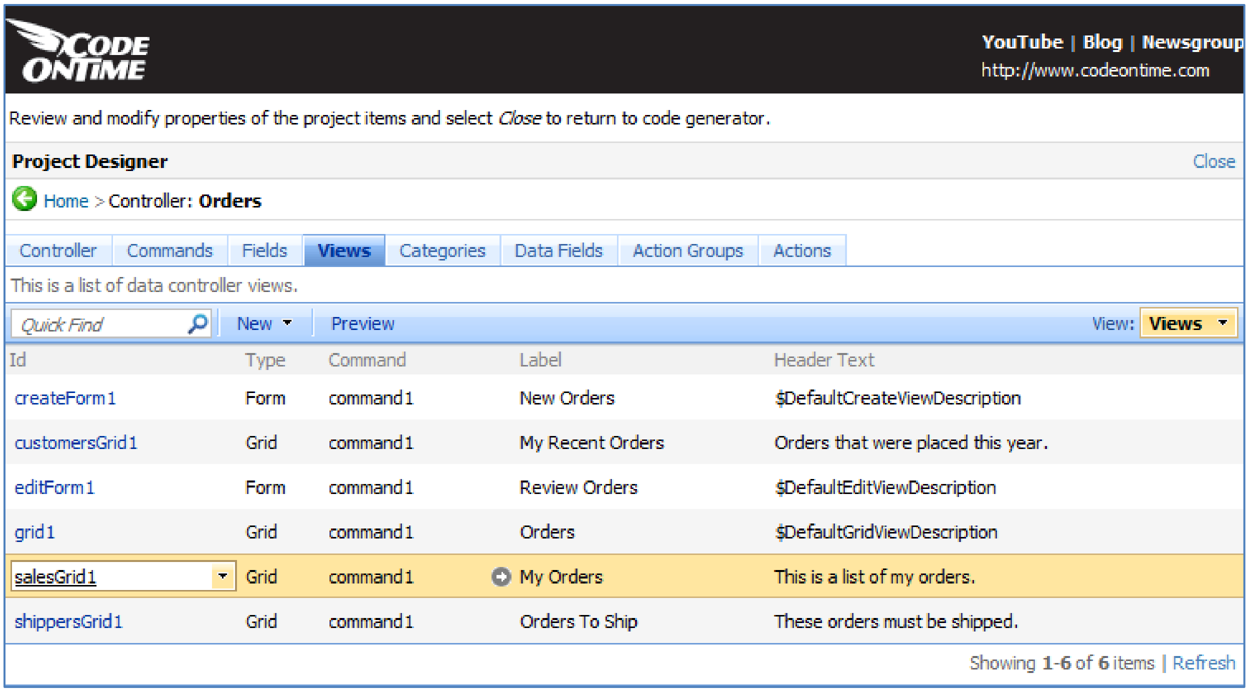
Customer Company Name	Employee Last Name	Order Date↓	Required Date	Shipped Date	Ship Via Company Name	Freight	Ship Name	Ship Address	
Simons bistro	King	5/6/1998	6/3/1998	n/a	United Package	$18.44	Simons bistro	Vinbæltet 34	
Richter Supermarkt	Callahan	5/6/1998	6/3/1998	n/a	United Package	$6.19	Richter Supermarkt	Starenweg 5	
Bon app'	Peacock	5/6/1998	6/3/1998	n/a	United Package	$38.28	Bon app'	12, rue des Bouchers	Marseille
Rattlesnake Canyon Grocery	Davolio	5/6/1998	6/3/1998	n/a	United Package	$8.53	Rattlesnake Canyon Grocery	2817 Milton Dr.	Albuquerque
Lehmanns Marktstand	Fuller	5/5/1998	6/2/1998	n/a	Speedy Express	$136.00	Lehmanns Marktstand	Magazinweg 7	Frankfurt a.M.
LILA-Supermercado	Davolio	5/5/1998	6/2/1998	n/a	Speedy Express	$0.93	LILA-Supermercado	Carrera 52 con Ave. Bolívar #65-98 Llano Largo	Barquisimeto
Ernst Handel	Peacock	5/5/1998	6/2/1998	n/a	United Package	$258.64	Ernst Handel	Kirchgasse 6	Graz
Pericles Comidas clásicas	Fuller	5/5/1998	6/2/1998	n/a	United Package	$24.95	Pericles Comidas clásicas	Calle Dr. Jorge Cash 321	México D.F.
Drachenblut Delikatessen	Davolio	5/4/1998	5/18/1998	5/6/1998	United Package	$7.98	Drachenblut Delikatessen	Walserweg 21	Aachen
Queen Cozinha	Callahan	5/4/1998	6/1/1998	n/a	United Package	$81.75	Queen Cozinha	Alameda dos Canàrios, 891	Sao Paulo

«Previous | Page: 1 2 3 4 5 6 7 8 9 10 … | Next » Items per page: 10, 15, 20, 25 | Showing 1-10 of 830 items | Refresh

The view *Orders To Ship* lists all of the most recent orders in descending order. The view *My Recent Orders* is designed for customers. We do not expect end users to sign in, go to the *Orders* page, and select the correct view from the selector, so we need to indicate when custom views will override *grid1* and present themselves to the end users. Custom views will replace *grid1* at runtime. We'll use *Virtual View ID* to configure replacement, and use *Override When* to configure condition for replacement to take place.

Define "Override When" and "Virtual View Id"

In *Code On Time Generator*, click on the project name in the project list and press the *Design* button. Select the *Orders* data controller from the controller list, and switch to the *Views* tab.



Review and modify properties of the project items and select *Close* to return to code generator.

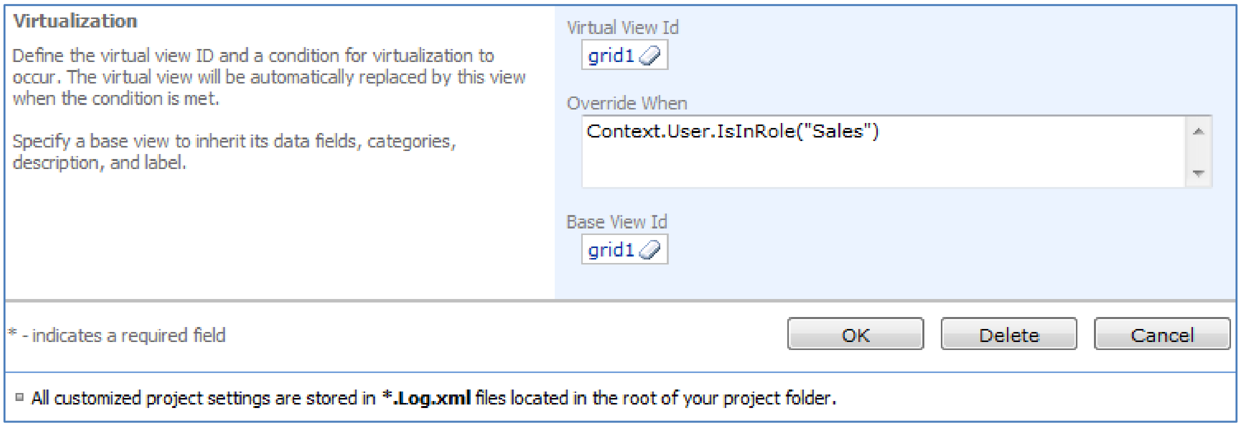
Project Designer

This is a list of data controller views.

Id	Type	Command	Label	Header Text
createForm1	Form	command1	New Orders	$DefaultCreateViewDescription
customersGrid1	Grid	command1	My Recent Orders	Orders that were placed this year.
editForm1	Form	command1	Review Orders	$DefaultEditViewDescription
grid1	Grid	command1	Orders	$DefaultGridViewDescription
salesGrid1	Grid	command1	My Orders	This is a list of my orders.
shippersGrid1	Grid	command1	Orders To Ship	These orders must be shipped.

Showing **1-6** of **6** items | Refresh

Click on *salesGrid1*, and press *Edit*. Scroll down to the *Virtualization* section. Change *Virtual View ID* to grid1. In the *OverrideWhen* field, type in "Context.User.IsInRole("Sales")". Save the view.



Virtualization

Define the virtual view ID and a condition for virtualization to occur. The virtual view will be automatically replaced by this view when the condition is met.

Specify a base view to inherit its data fields, categories, description, and label.

Virtual View Id: grid1

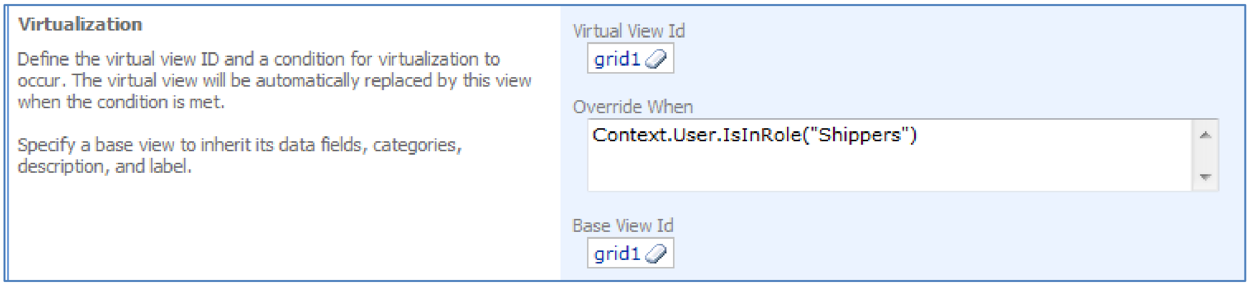
Override When: Context.User.IsInRole("Sales")

Base View Id: grid1

* - indicates a required field

All customized project settings are stored in ***.Log.xml** files located in the root of your project folder.

The view *shippersGrid1* will have a similar expression. Change *Virtual View ID* to *grid1*, and for *OverrideWhen*, type in “Context.User.IsInRole(“Shippers”)”.



Virtualization

Define the virtual view ID and a condition for virtualization to occur. The virtual view will be automatically replaced by this view when the condition is met.

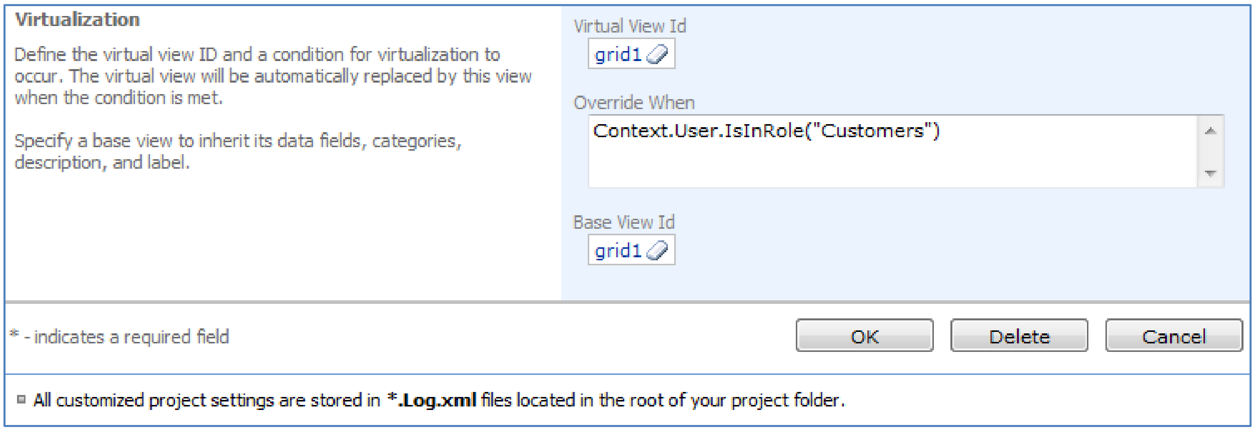
Specify a base view to inherit its data fields, categories, description, and label.

Virtual View Id: grid1

Override When: Context.User.IsInRole("Shippers")

Base View Id: grid1

Perform the same operation for *customersGrid1*. Change *Virtual View ID* to *grid1*, and *OverrideWhen* to “Context.User.IsInRole(“Customers”)”.



Virtualization

Define the virtual view ID and a condition for virtualization to occur. The virtual view will be automatically replaced by this view when the condition is met.

Specify a base view to inherit its data fields, categories, description, and label.

Virtual View Id: grid1

Override When: Context.User.IsInRole("Customers")

Base View Id: grid1

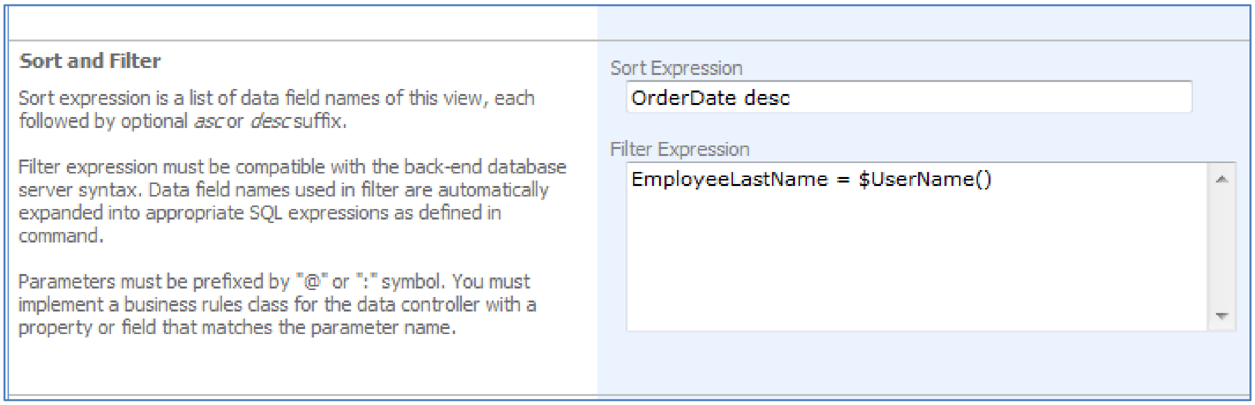
* - indicates a required field

All customized project settings are stored in ***.Log.xml** files located in the root of your project folder.

Define View Filters

Now we need to define row-level view filters. These will insure that sales and customers will see their respective orders, and that shippers will see orders that were not shipped or shipped this or last week.

For *salesGrid1*, type in the *Filter Expression* of “EmployeeLastName = $UserName()”.



Sort and Filter

Sort expression is a list of data field names of this view, each followed by optional *asc* or *desc* suffix.

Filter expression must be compatible with the back-end database server syntax. Data field names used in filter are automatically expanded into appropriate SQL expressions as defined in command.

Parameters must be prefixed by "@" or ":" symbol. You must implement a business rules class for the data controller with a property or field that matches the parameter name.

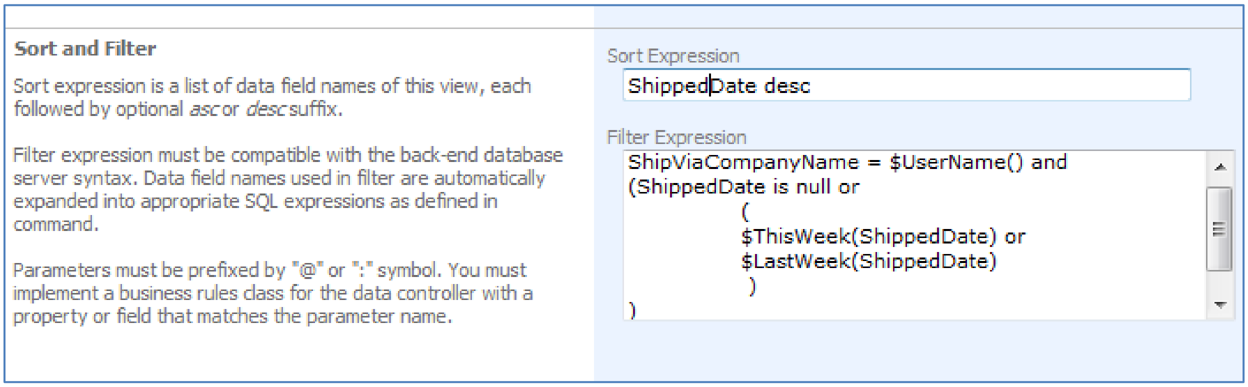
Sort Expression: OrderDate desc

Filter Expression: EmployeeLastName = $UserName()

For *shippersGrid1*, type in the *Filter Expression* of:

```
ShipViaCompanyName = $UserName() and
(ShippedDate is null or
            (
            $ThisWeek(ShippedDate) or
            $LastWeek(ShippedDate)
            )
)
```

Type it in the field as shown below.



Sort and Filter

Sort expression is a list of data field names of this view, each followed by optional *asc* or *desc* suffix.

Filter expression must be compatible with the back-end database server syntax. Data field names used in filter are automatically expanded into appropriate SQL expressions as defined in command.

Parameters must be prefixed by "@" or ":" symbol. You must implement a business rules class for the data controller with a property or field that matches the parameter name.

Sort Expression: ShippedDate desc

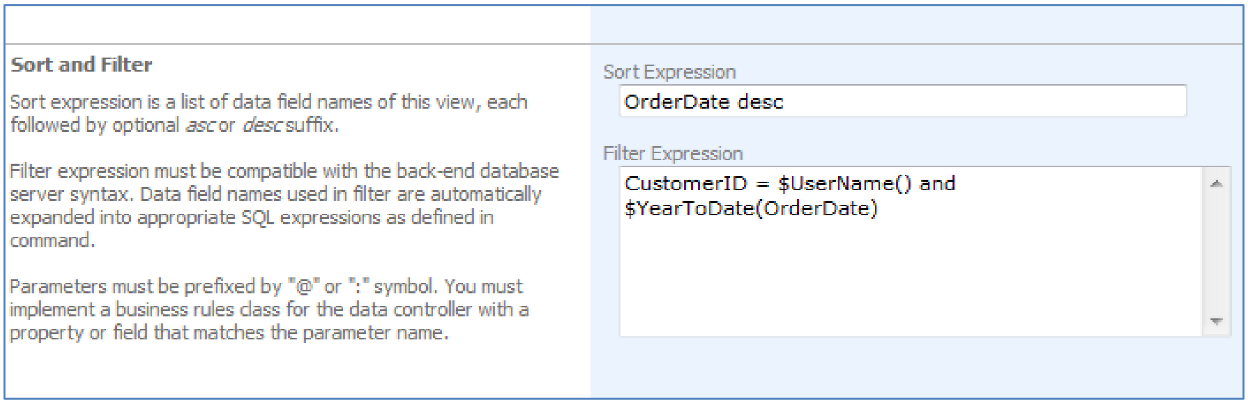
Filter Expression:

```
ShipViaCompanyName = $UserName() and
(ShippedDate is null or
         (
         $ThisWeek(ShippedDate) or
         $LastWeek(ShippedDate)
          )
)
```

Lastly, set *the Filter Expression* of *customersGrid1* to

```
CustomerID = $UserName() and
$YearToDate(OrderDate)
```

Type it in the field below.



Sort and Filter

Sort expression is a list of data field names of this view, each followed by optional *asc* or *desc* suffix.

Filter expression must be compatible with the back-end database server syntax. Data field names used in filter are automatically expanded into appropriate SQL expressions as defined in command.

Parameters must be prefixed by "@" or ":" symbol. You must implement a business rules class for the data controller with a property or field that matches the parameter name.

Sort Expression: OrderDate desc

Filter Expression:

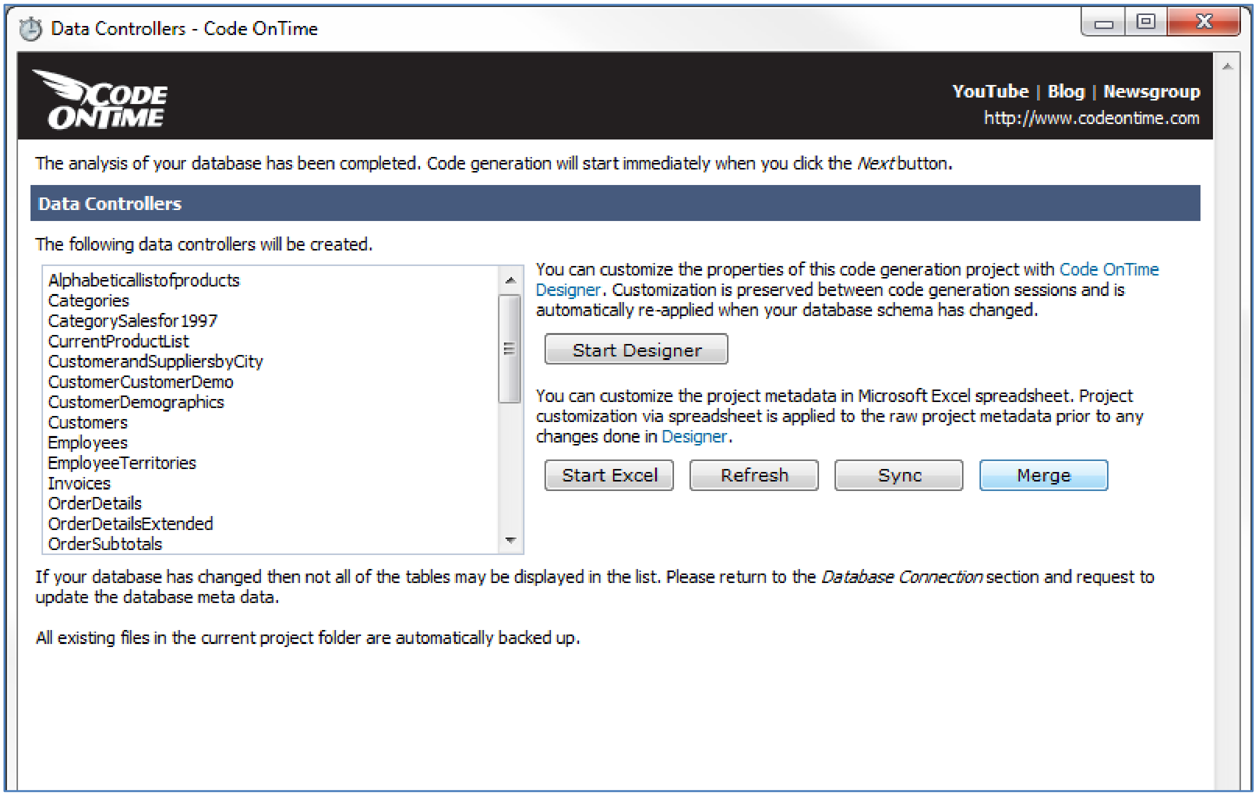
```
CustomerID = $UserName() and
$YearToDate(OrderDate)
```

Close the designer, and regenerate the application. Once the *Generator* finishes, a web page will appear with the freshly generated application.

Now we can use the user accounts created earlier to test the modifications. Skip to Viewing the Results section to learn how to use your new row-level security.

Row-Level Security Using the Designer Spreadsheet

Select your project in *Code On Time Generator*, and press the *Skip* button. You will arrive on the *Data Controllers* page. Click on the *Merge* button and confirm to create a *Designer Spreadsheet*.



Data Controllers - Code OnTime
CODE ONTIME
YouTube | Blog | Newsgroup
http://www.codeontime.com

The analysis of your database has been completed. Code generation will start immediately when you click the *Next* button.

Data Controllers

The following data controllers will be created.

Alphabeticallistofproducts
Categories
CategorySalesfor1997
CurrentProductList
CustomerandSuppliersbyCity
CustomerCustomerDemo
CustomerDemographics
Customers
Employees
EmployeeTerritories
Invoices
OrderDetails
OrderDetailsExtended
OrderSubtotals

You can customize the properties of this code generation project with Code OnTime Designer. Customization is preserved between code generation sessions and is automatically re-applied when your database schema has changed.

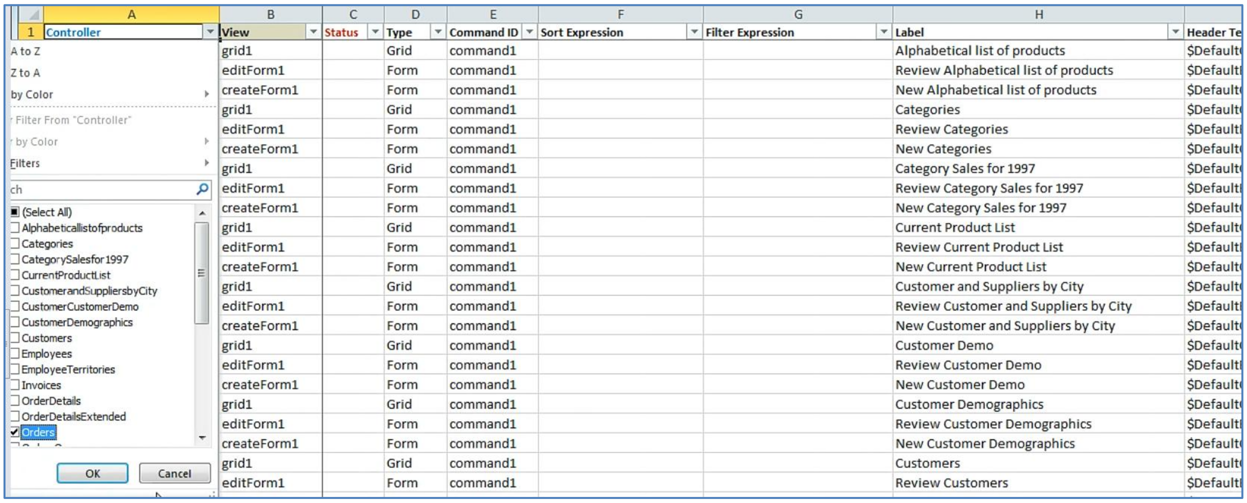
You can customize the project metadata in Microsoft Excel spreadsheet. Project customization via spreadsheet is applied to the raw project metadata prior to any changes done in Designer.

If your database has changed then not all of the tables may be displayed in the list. Please return to the *Database Connection* section and request to update the database meta data.

All existing files in the current project folder are automatically backed up.

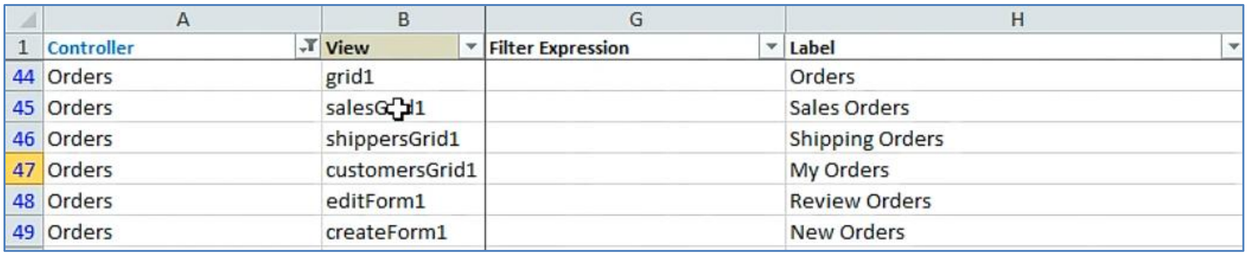
Define Role-Specific Views

When the spreadsheet loads, switch to the *Views* tab. Filter the *Controller* column to see only the *Views* that belong to *Orders* data controller.



View	Status	Type	Command ID	Sort Expression	Filter Expression	Label	Header Te
grid1		Grid	command1			Alphabetical list of products	$Default
editForm1		Form	command1			Review Alphabetical list of products	$Default
createForm1		Form	command1			New Alphabetical list of products	$Default
grid1		Grid	command1			Categories	$Default
editForm1		Form	command1			Review Categories	$Default
createForm1		Form	command1			New Categories	$Default
grid1		Grid	command1			Category Sales for 1997	$Default
editForm1		Form	command1			Review Category Sales for 1997	$Default
createForm1		Form	command1			New Category Sales for 1997	$Default
grid1		Grid	command1			Current Product List	$Default
editForm1		Form	command1			Review Current Product List	$Default
createForm1		Form	command1			New Current Product List	$Default
grid1		Grid	command1			Customer and Suppliers by City	$Default
editForm1		Form	command1			Review Customer and Suppliers by City	$Default
createForm1		Form	command1			New Customer and Suppliers by City	$Default
grid1		Grid	command1			Customer Demo	$Default
editForm1		Form	command1			Review Customer Demo	$Default
createForm1		Form	command1			New Customer Demo	$Default
grid1		Grid	command1			Customer Demographics	$Default
editForm1		Form	command1			Review Customer Demographics	$Default
createForm1		Form	command1			New Customer Demographics	$Default
grid1		Grid	command1			Customers	$Default
editForm1		Form	command1			Review Customers	$Default

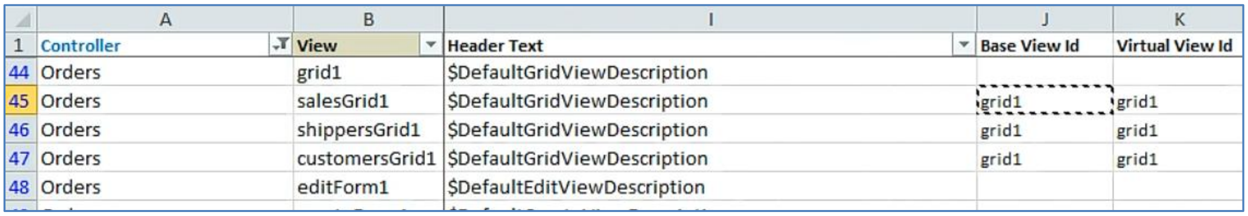
Select *grid1*, and make three copies of this row. Give the new rows the names *salesGrid1*, *shippersGrid1*, and *customersGrid1*, with the labels *Sales Orders*, *Shipping Orders*, and *My Orders*, respectively.



	A	B	G	H
1	Controller	View	Filter Expression	Label
44	Orders	grid1		Orders
45	Orders	salesGrid1		Sales Orders
46	Orders	shippersGrid1		Shipping Orders
47	Orders	customersGrid1		My Orders
48	Orders	editForm1		Review Orders
49	Orders	createForm1		New Orders

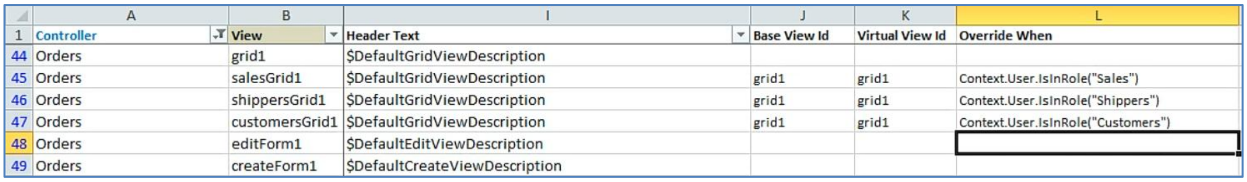
Define "Override When" and "Virtual View Id"

These three views will have the value *grid1* for *Base View Id* and for *Virtual View Id*.



	A	B	I	J	K
1	Controller	View	Header Text	Base View Id	Virtual View Id
44	Orders	grid1	$DefaultGridViewDescription		
45	Orders	salesGrid1	$DefaultGridViewDescription	grid1	grid1
46	Orders	shippersGrid1	$DefaultGridViewDescription	grid1	grid1
47	Orders	customersGrid1	$DefaultGridViewDescription	grid1	grid1
48	Orders	editForm1	$DefaultEditViewDescription		

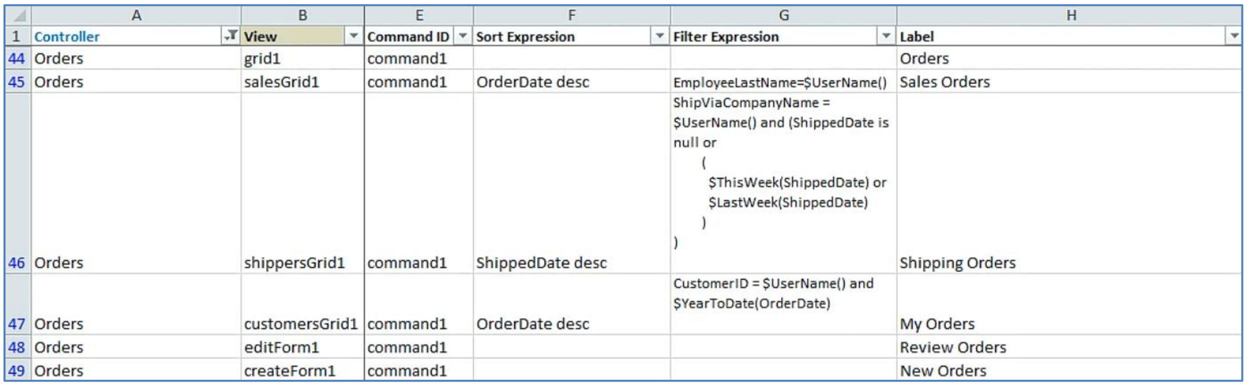
For the *Override When* field, write in the values as shown below.



	A	B	I	J	K	L
1	Controller	View	Header Text	Base View Id	Virtual View Id	Override When
44	Orders	grid1	$DefaultGridViewDescription			
45	Orders	salesGrid1	$DefaultGridViewDescription	grid1	grid1	Context.User.IsInRole("Sales")
46	Orders	shippersGrid1	$DefaultGridViewDescription	grid1	grid1	Context.User.IsInRole("Shippers")
47	Orders	customersGrid1	$DefaultGridViewDescription	grid1	grid1	Context.User.IsInRole("Customers")
48	Orders	editForm1	$DefaultEditViewDescription			
49	Orders	createForm1	$DefaultCreateViewDescription			

Define View Filters

There are two more fields that need to be modified for this example, *Sort* and *Filter Expression*. *Sales* and *Customers* grid will be sorted in descending order by *Order* Date. *Shippers* grid will be sorted by descending *Shipped Date*. The filtering will be done as displayed in the picture below. The Filter Expressions are listed in the *Code On Time Designer* section, Define View Filters.

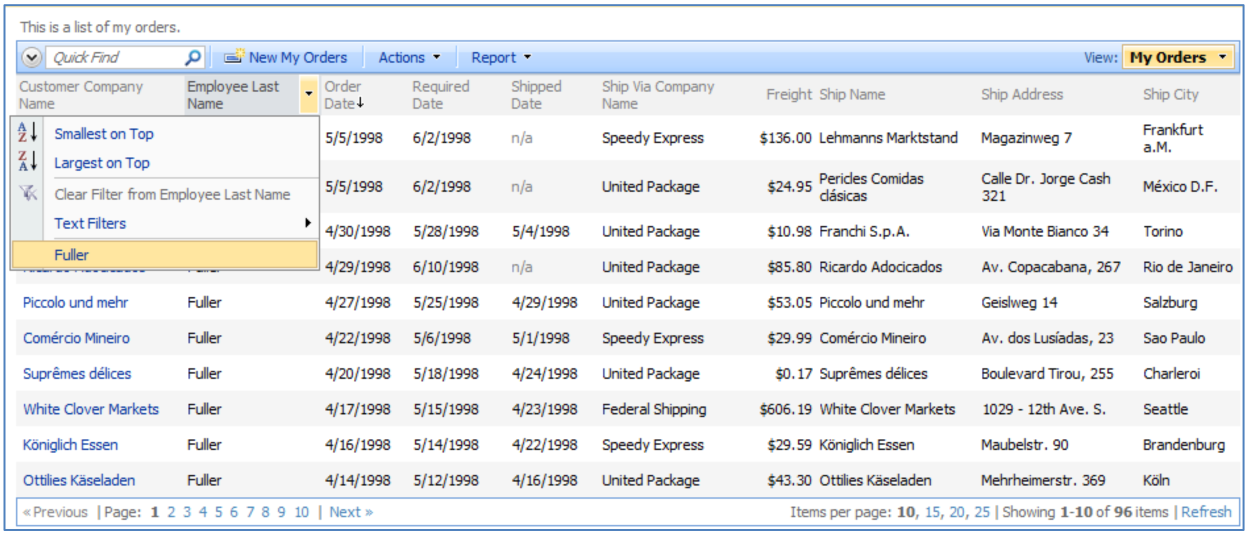


	A	B	E	F	G	H
1	Controller	View	Command ID	Sort Expression	Filter Expression	Label
44	Orders	grid1	command1			Orders
45	Orders	salesGrid1	command1	OrderDate desc	EmployeeLastName=$UserName()	Sales Orders
46	Orders	shippersGrid1	command1	ShippedDate desc	ShipViaCompanyName = $UserName() and (ShippedDate is null or ($ThisWeek(ShippedDate) or $LastWeek(ShippedDate)))	Shipping Orders
47	Orders	customersGrid1	command1	OrderDate desc	CustomerID = $UserName() and $YearToDate(OrderDate)	My Orders
48	Orders	editForm1	command1			Review Orders
49	Orders	createForm1	command1			New Orders

Viewing the Results

First, we sign in as Fuller. The password is *user123%*. Click *Login*, and navigate to the *Orders* page.

You can see that all the orders listed are those with the *Employee Last Name* of *Fuller*, and there are no other filtering options available. The orders are sorted according to *OrderDate*.

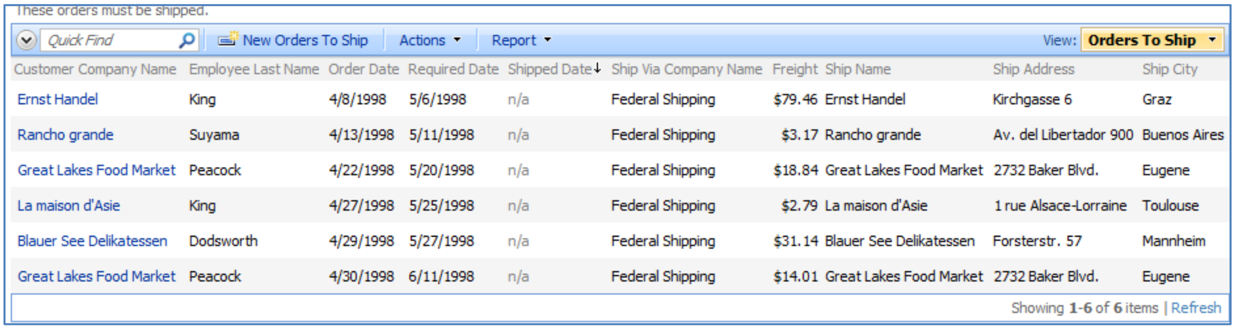


This is a list of my orders.

Customer Company Name	Employee Last Name	Order Date↓	Required Date	Shipped Date	Ship Via Company Name	Freight	Ship Name	Ship Address	Ship City
		5/5/1998	6/2/1998	n/a	Speedy Express	$136.00	Lehmanns Marktstand	Magazinweg 7	Frankfurt a.M.
		5/5/1998	6/2/1998	n/a	United Package	$24.95	Pericles Comidas clásicas	Calle Dr. Jorge Cash 321	México D.F.
		4/30/1998	5/28/1998	5/4/1998	United Package	$10.98	Franchi S.p.A.	Via Monte Bianco 34	Torino
		4/29/1998	6/10/1998	n/a	United Package	$85.80	Ricardo Adocicados	Av. Copacabana, 267	Rio de Janeiro
Piccolo und mehr	Fuller	4/27/1998	5/25/1998	4/29/1998	United Package	$53.05	Piccolo und mehr	Geislweg 14	Salzburg
Comércio Mineiro	Fuller	4/22/1998	5/6/1998	5/1/1998	Speedy Express	$29.99	Comércio Mineiro	Av. dos Lusíadas, 23	Sao Paulo
Suprêmes délices	Fuller	4/20/1998	5/18/1998	4/24/1998	United Package	$0.17	Suprêmes délices	Boulevard Tirou, 255	Charleroi
White Clover Markets	Fuller	4/17/1998	5/15/1998	4/23/1998	Federal Shipping	$606.19	White Clover Markets	1029 - 12th Ave. S.	Seattle
Königlich Essen	Fuller	4/16/1998	5/14/1998	4/22/1998	Speedy Express	$29.59	Königlich Essen	Maubelstr. 90	Brandenburg
Ottilies Käseladen	Fuller	4/14/1998	5/12/1998	4/16/1998	United Package	$43.30	Ottilies Käseladen	Mehrheimerstr. 369	Köln

« Previous | Page: 1 2 3 4 5 6 7 8 9 10 | Next » Items per page: 10, 15, 20, 25 | Showing 1-10 of 96 items | Refresh

Let's sign out, and sign in again as *Federal Shipping*, with the password of *user123%*. On the *Orders* page, you can see that all of the orders have *Federal Shipping* as *Ship Via Company Name*, and no other filtering options are available. It is sorted in descending order by *Shipped Date*. You can see that the orders listed are those shipped this week, last week, or were never shipped.

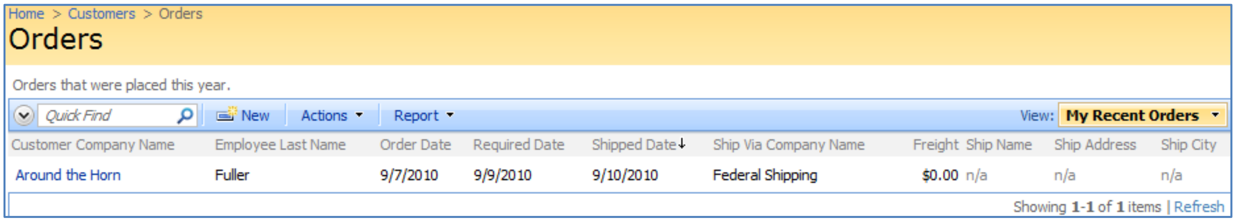


These orders must be shipped.

Customer Company Name	Employee Last Name	Order Date	Required Date	Shipped Date↓	Ship Via Company Name	Freight	Ship Name	Ship Address	Ship City
Ernst Handel	King	4/8/1998	5/6/1998	n/a	Federal Shipping	$79.46	Ernst Handel	Kirchgasse 6	Graz
Rancho grande	Suyama	4/13/1998	5/11/1998	n/a	Federal Shipping	$3.17	Rancho grande	Av. del Libertador 900	Buenos Aires
Great Lakes Food Market	Peacock	4/22/1998	5/20/1998	n/a	Federal Shipping	$18.84	Great Lakes Food Market	2732 Baker Blvd.	Eugene
La maison d'Asie	King	4/27/1998	5/25/1998	n/a	Federal Shipping	$2.79	La maison d'Asie	1 rue Alsace-Lorraine	Toulouse
Blauer See Delikatessen	Dodsworth	4/29/1998	5/27/1998	n/a	Federal Shipping	$31.14	Blauer See Delikatessen	Forsterstr. 57	Mannheim
Great Lakes Food Market	Peacock	4/30/1998	6/11/1998	n/a	Federal Shipping	$14.01	Great Lakes Food Market	2732 Baker Blvd.	Eugene

Showing 1-6 of 6 items | Refresh

This time, sign in as *AROUT*, with the password *user123%.* You can see that the list of orders has only one record in it, which belongs to the company of *AROUT*, and it is sorted in descending order of *Order Date*.



Orders

Orders that were placed this year.

Customer Company Name	Employee Last Name	Order Date	Required Date	Shipped Date↓	Ship Via Company Name	Freight	Ship Name	Ship Address	Ship City
Around the Horn	Fuller	9/7/2010	9/9/2010	9/10/2010	Federal Shipping	$0.00	n/a	n/a	n/a

Showing 1-1 of 1 items | Refresh

Ideal Implementation

The implementation that we have created is not ideal. *Northwind* database was not designed for row-level security. Ideally, *UserName* or *UserId* column would be available in the tables *Employees*, *Shippers*, and *Customers*. Filters with the *$UserName()* or *$UserId()* functions would be matched to the corresponding data fields.

If you were to have the *UserName* field in the specified tables, then a de-normalization field map will need to be set up to automatically include *UserName* in the respective data controllers.

You would also need to change your filters by replacing the data fields *EmployeeLastName*, *ShipViaCompanyName*, and *CustomerCompanyName*.