

2011



USER GUIDE

Conditional Visibility of Fields and Categories

Wizards and Status Bars

Code On Time web applications offer powerful methods of presenting your data with very little effort. In this tutorial we will show you how to create a *wizard* form view with the context sensitive status bar.

Form views *editForm1* and *createForm1* are included with each data controller in a generated Code On Time web application. The first form, *editForm1*, is typically used to present a data row in “view” or “edit” mode. The second form, *createForm1*, provides user interface for “new” data rows. The picture below shows an example of *editForm1* rendering an employee record in “view” and “edit” modes.

Each form view must have at least one category that binds data controller fields to a view. A binding of a field to a view is called a “data field”. Only one binding of a field to a view can exist in any type of view.

For purposes of this tutorial, create your own *Northwind* sample application as explained at <http://blog.codeontime.com/2011/10/northwind-sample.html>.

Multiple Data Field Categories in Form Views

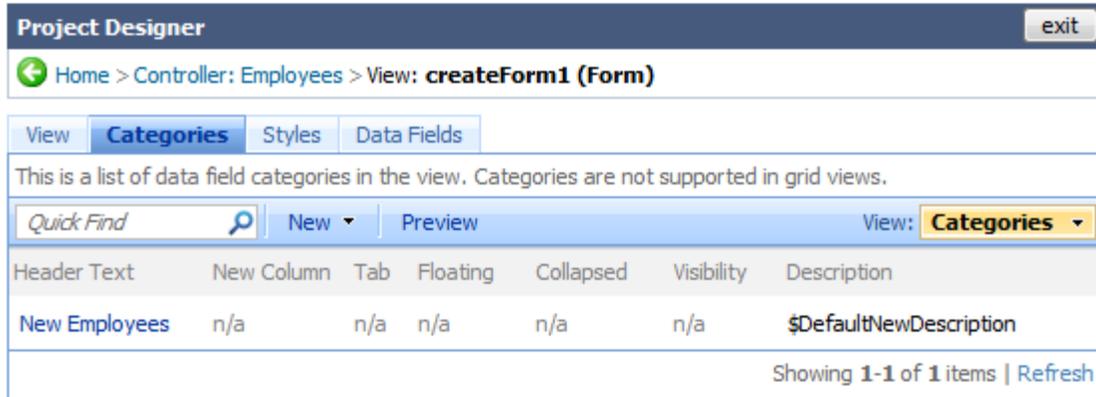
A simple list of business object data fields presented in a form view works well in many situations. On the other hand, business requirements might call for a grouping of related fields and even conditional display of field groups based on the user input.

Consider the following screenshot depicting automatically generated *createForm1* of *Employees* data controller.

The screenshot shows a web browser window with the URL <http://localhost:34198/Pages/Employees.aspx>. The page title is "Employees" and the user is logged in as "admin". The page contains a navigation menu with "Employees" selected. The main content area is titled "Employees" and contains a form for creating a new employee record. The form has a "Record" dropdown menu set to "New Employees" and "OK" and "Cancel" buttons. The form fields are: Last Name (required), First Name (required), Title, Title Of Courtesy, Birth Date (with a calendar icon), Hire Date (with a calendar icon), Address, City, Region, Postal Code, Country, Home Phone, Extension, Notes (text area), Reports To Last Name (dropdown menu), and Photo Path. A footer note states "© 2011 MyCompany. All rights reserved."

Let's break this form in four categories presenting "New Employee" fields, "Address", "Miscellaneous" information, and an overall "Summary" of a new employee record.

Start the web application generator, select the name of your project, and click the *Design* button. Select *Employees* data controller on *All Controllers* tab. Activate *Views* tab and select view *createForm1*. Activate *Categories* tab, shown below.



Rename *New Employees* category to *New Employee* and clear the description.

Add three more categories with the following properties:

Header Text	Visibility	Description
Address	true	Enter address of {FirstName} {LastName}.
Miscellaneous	true	Enter {FirstName} {LastName}'s phone number, birthday, any relevant notes.
Summary	true	Please review the summary of the new record. <div style="margin:8px; padding:8px; height:220px; overflow:auto; border:solid 1px silver"> Last Name: {LastName}
 First Name: {FirstName}
 Title: {Title}
 Title Of Courtesy: {TitleOfCourtesy}
 Birth Date: {BirthDate}
 Hire Date: {HireDate}
 Address: {Address}
 City: {City}
 Region: {Region}
 Postal Code: {PostalCode}
 Country: {Country}
 Home Phone: {HomePhone}
 Extension: {Extension}
 Notes: {Notes}
 Reports To: {ReportsTo}
 Photo Path: {PhotoPath}
 </div> </div>

The list of categories in the Designer will look as follows.

Project Designer exit

Home > Controller: Employees > View: **createForm1 (Form)**

View **Categories** Styles Data Fields

This is a list of data field categories in the view. Categories are not supported in grid views.

Quick Find New ▾ Preview ▲ Up ▼ Down View: **Categories** ▾

Header Text	New Column	Tab	Floating	Collapsed	Visibility	Description
New Employee	n/a	n/a	n/a	n/a	n/a	n/a
Address	n/a	n/a	n/a	n/a	true	Enter address of {FirstName} {LastName}.
Miscellaneous	n/a	n/a	n/a	n/a	true	Enter {FirstName} {LastName}'s phone number, birthday, any relevant notes.
Summary	n/a	n/a	n/a	n/a	true	Please review the summary of the new record. <div style="margin:8px;padding:8px;height:220px;overflow:auto;border:solid 1px silver"> Last Name: {LastName} First Name: {FirstName}…

Showing 1-4 of 4 items | Refresh

All data fields are presently bound to the *New Employee* category.

Project Designer exit

Home > Controller: Employees > View: createForm1 (Form) > Category: **New Employee**

Category **Data Fields**

This is a list of data fields.

Quick Find New ▾ Preview View: **Data Fields** ▾

Field Name	Category	Read Only	Cols	Rows	Data Type	Format	Alias	Hidden	Text Mode	Search	Σ	Chart
LastName	New Employee	Default	20	n/a	String	n/a	n/a	No	Auto	Default	None	None
FirstName	New Employee	Default	10	n/a	String	n/a	n/a	No	Auto	Default	None	None
Title	New Employee	Default	30	n/a	String	n/a	n/a	No	Auto	Default	None	None
TitleOfCourtesy	New Employee	Default	25	n/a	String	n/a	n/a	No	Auto	Default	None	None
BirthDate	New Employee	Default	10	n/a	DateTime	n/a	n/a	No	Auto	Default	None	None
HireDate	New Employee	Default	10	n/a	DateTime	n/a	n/a	No	Auto	Default	None	None
Address	New Employee	Default	n/a	n/a	String	n/a	n/a	No	Auto	Default	None	None
City	New Employee	Default	15	n/a	String	n/a	n/a	No	Auto	Default	None	None
Region	New Employee	Default	15	n/a	String	n/a	n/a	No	Auto	Default	None	None
PostalCode	New Employee	Default	10	n/a	String	n/a	n/a	No	Auto	Default	None	None

« Previous | Page: 1 2 | Next » Items per page: 10, 15, 20, 25, 50, 100 | Showing 1-10 of 16 items | Refresh

Let's change that. Select view *createForm1* in the path at the top of the *Designer* page and activate *Categories* tab.

Select *Address* category, activate *Data Fields* tab and add new data fields referencing *Address*, *City*, *Region*, *PostalCode*, and *Country* fields of the controller.

You will notice that the *Designer* automatically copies properties of the fields from the *New Employee* category. The *Designer* also removes the fields from *New Employee* category to ensure that there are no duplicate field references.

Now follow the same routine and add *TitleOfCourtesy*, *BirthDate*, *HomePhone*, *Notes*, *ReportsTo*, and *PhotoPath* to *Miscellaneous* category of view *createForm1*. The new layout of fields of the view *createForm1* is presented in the screen shot.

Project Designer exit

Home > Controller: Employees > View: createForm1 (Form)

View Categories Styles **Data Fields**

This is a list of data fields.

Quick Find 🔍 New ▾ Preview View: Data Fields ▾

Field Name	Category	Read Only	Cols	Rows	Data Type	Format	Alias	Hidden	Text Mode	Search	Σ	Chart
LastName	New Employee	Default	20	n/a	String	n/a	n/a	No	Auto	Default	None	None
FirstName	New Employee	Default	10	n/a	String	n/a	n/a	No	Auto	Default	None	None
Title	New Employee	Default	30	n/a	String	n/a	n/a	No	Auto	Default	None	None
HireDate	New Employee	Default	10	n/a	DateTime	n/a	n/a	No	Auto	Default	None	None
Extension	New Employee	Default	4	n/a	String	n/a	n/a	No	Auto	Default	None	None
Address	Address	Default	n/a	n/a	String	n/a	n/a	No	Auto	Default	None	None
City	Address	Default	15	n/a	String	n/a	n/a	No	Auto	Default	None	None
Region	Address	Default	15	n/a	String	n/a	n/a	No	Auto	Default	None	None
PostalCode	Address	Default	10	n/a	String	n/a	n/a	No	Auto	Default	None	None
Country	Address	Default	15	n/a	String	n/a	n/a	No	Auto	Default	None	None
TitleOfCourtesy	Miscellaneous	Default	25	n/a	String	n/a	n/a	No	Auto	Default	None	None
BirthDate	Miscellaneous	Default	10	n/a	DateTime	n/a	n/a	No	Auto	Default	None	None
HomePhone	Miscellaneous	Default	24	n/a	String	n/a	n/a	No	Auto	Default	None	None
Notes	Miscellaneous	Default	n/a	5	String	n/a	n/a	No	Auto	Default	None	None
ReportsTo	Miscellaneous	Default	n/a	n/a	Int32	n/a	ReportsToLastName	No	Auto	Default	None	None
PhotoPath	Miscellaneous	Default	n/a	n/a	String	n/a	n/a	No	Auto	Default	None	None

Showing 1-16 of 16 items | Refresh

We also recommend that you select data fields *Title* and *TitleOfCourtesy* and set their *Auto Complete Prefix Length* property to "1". This will provide an auto complete option for both fields.

Generate web application and start creating a new *Employee* record. As you enter values in *FirstName* and *LastName* fields, the descriptions of categories will change. Notice that *John Doe* is displayed in the category descriptions in the picture below.

The screenshot shows a web browser window displaying a form for creating a new employee record. The browser's address bar shows the URL `http://localhost:34198/Pages/Employees.aspx`. The page title is "Employees" and the user is logged in as "admin". The form is titled "New Employee" and includes the following fields:

- Last Name ***: Doe
- First Name ***: John
- Title**: A dropdown menu with "Sales Manager" selected.
- Hire Date**: A date picker.
- Extension**: A text input field.
- Address**: A section titled "Enter address of John Doe" with fields for Address, City, Region, Postal Code, and Country.
- Miscellaneous**: A section titled "Enter John Doe's phone number, birthday, any relevant notes" with fields for Title Of Courtesy, Birth Date, Home Phone, Notes, Reports To Last Name (with a "(select)" button), and Photo Path.
- Summary**: A section titled "Please review the summary of the new record." with a text area containing:
 - Last Name: Doe
 - First Name: John
 - Title: N/A

Buttons for "OK" and "Cancel" are located at the top right of the form. A "View: New Employees" dropdown is also present.

Dynamic Category Descriptions

References to the names of fields enclosed in curly brackets are automatically replaced with field values as soon as the field value is changed, provided that the category visibility is dynamic.

Descriptions are automatically formatted with field value injection when you open a form view. If the category visibility is not dynamic then the values will not change even if the user is editing the record.

We have entered “true” expression in *Visibility* property of *Address*, *Miscellaneous*, and *Summary* categories. The expression is written in *JavaScript* and evaluated whenever data values are changed by user. This will ensure that the categories will be permanently visible, as “true” tends to evaluate to true.

The following picture shows collapsed *Address* and *Miscellaneous* categories and fully expanded *New Employee* and *Summary* categories. Note that *Summary* category has no fields bound to it but displays dynamic content thanks to the expressions embedded in the category description.

The screenshot shows a web browser window with the URL `http://localhost:34198/Pages/Employees.aspx`. The page title is "Employees" and the breadcrumb is "Home > Employees". The main heading is "Employees". Below the heading, there is a navigation bar with "Home", "Customers", "Employees", "Categories", "Customer Demographics", "Region", "Reports", and "Membership".

The form is titled "New Employee" and includes the following fields:

- Last Name * (Doe)
- First Name * (John)
- Title (Sales Representative)
- Hire Date (10/21/2011)
- Extension (8880)

The form also has sections for "Address", "Miscellaneous", and "Summary". The "Address" section is collapsed, and the "Miscellaneous" section is also collapsed. The "Summary" section is expanded and displays the following information:

Please review the summary of the new record.

Last Name: Doe
First Name: John
Title: Sales Representative
Title Of Courtesy: Mr.
Birth Date: 1/8/1973
Hire Date: 10/21/2011
Address: 507 - 20th Ave. E.Apt. 2A
City: Seattle
Region: WA
Postal Code: 92122
Country: USA
Home Phone: (206) 555-9857
Extension: 8880
Notes: Great skills and vast knowledge of our industry.
Reports To: Fuller
Photo Path: N/A

At the bottom of the form, there are "OK" and "Cancel" buttons. The footer of the page reads "© 2011 MyCompany. All rights reserved."

Converting Form View to a Wizard

Our form is quite lengthy and warrants some refinement and simplification.

We will change the form to present one category at any given time and move between categories upon request. This style of presentation is often referred as a wizard. Data field categories in view *createForm1* will become the “pages” of the wizard.

If a category is visible then two buttons *Back* and *Next* will be visible as well to allow advancement to the next step or return to previous step of data collection.

We will rename *OK* button to *Finish* and have it available on the *Summary* page of *New Employee* wizard only. Button *Cancel* will remain visible at all times.

Status Field

The upcoming release of EASE (Enterprise Application Services Engine) will be available in [Unlimited](#) edition of Code On Time. It has also brought some enhancements to the application framework for other editions.

Various elements of your application pay special attention to the presence of a field named “Status”.

The column named *Status* is frequently found in database tables of a typical line-of-business application. This column generally contains a short phrase or a number reflecting the business state of a data row representing a business object. For example, an *Order* can have a status of *Draft*, *Open*, *Cancelled*, or *Ready to Ship*.

The *Code On Time* application framework assumes that field *Status* exists in all data controllers. Client library will automatically add a virtual *Status* field to any business objects if the physical field is not found.

Visibility of Categories

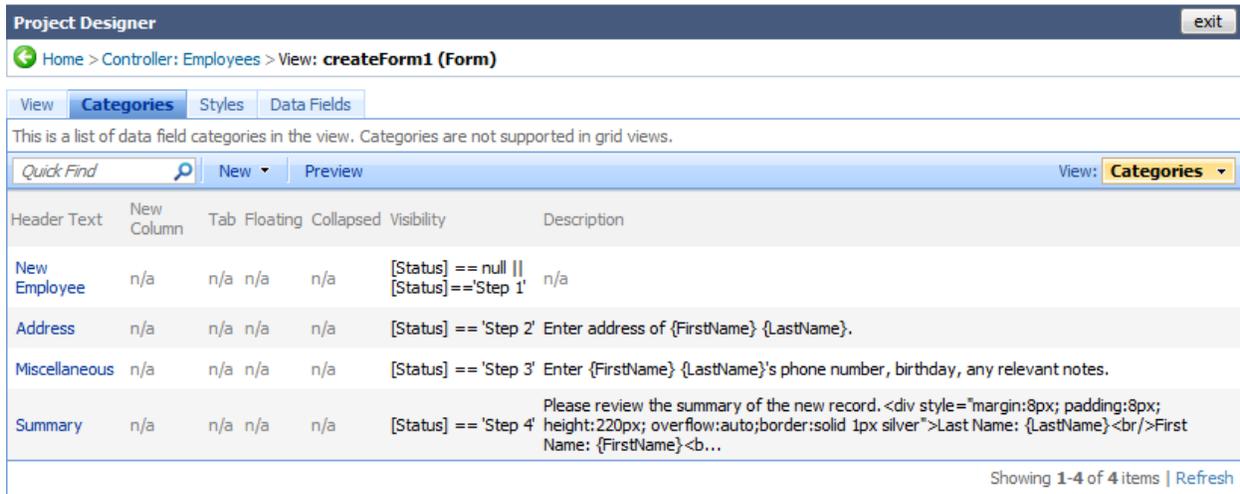
Implementation of a wizard calls for some sort of status that can be used to determine the active “page” of the wizard.

We will use the virtual “Status” field to determine the visibility of a category and have the value of the field change when a user moves from one “page” of the wizard to another.

Change the *Visibility expression* of categories in *createForm1* view as follows.

Category	Visibility Expression (JavaScript)
New Employee	[Status] == null [Status] == 'Step 1'
Address	[Status] == 'Step 2'
Miscellaneous	[Status] == 'Step 3'
Summary	[Status] == 'Step 4'

The screenshot of *Project Designer* shows categories with modified Visibility expression.



The initial value of virtual field “Status” is null, which will guarantee that only *New Employee* category is visible when a user starts creating a new record.

“Status” Action

If you were to implement an order management system with each order having a certain status then it is possible that you would have used a drop down list of available statuses to present the current status value.

Most of the time, change of the internal status of a business object results in a change to the user interface.

A status change is better implemented as a user interface action. For example, a buyer will set the order status to *Submitted* by pushing a *Submit* button. An employee in the shipping department will select a menu option *Ship* to change the order status to *Shipped*.

Code On Time web applications offer different action scopes that result in clickable links and button rendering in forms, on action bars, in action column, and grid context menu. We have introduced a new standard action Status that will change the value of a virtual or physical field with the name “Status” to the argument of the action.

Click *Employees* controller in the path at top of the project designer and activate *Action Groups* tab.

Select action group *ag2* (that has a scope of *Form*).

Activate *Actions* tab of the group.

Filter available actions by *When Last Command Name* property to the value “New”. You will see definitions of two actions with command names *Insert* and *Cancel*. These actions are rendered as push buttons with captions *OK* and *Cancel* in the previous picture.

Command Name	Command Argument	Header Text	When Last Command Name	When Last Command Argument	When Client Script (JavaScript)	When Key Selected	When HRef (Regex)	When View (Regex)	Roles	Scope
Insert	n/a	n/a	New	n/a	n/a	n/a	n/a	n/a	n/a	Form
Cancel	n/a	n/a	New	n/a	n/a	n/a	n/a	n/a	n/a	Form

Showing 1-2 of 2 items | Refresh

Add seven actions, defined below.

#	Command Name	Command Argument	Header Text	Causes Validation	When Last Command Name	When Client Script
1.	None		Back	No	New	[Status] == null [Status] == 'Step 1'
2.	Status	Step 2	Next	Yes	New	[Status] == null [Status] == 'Step 1'
3.	Status	Step 1	Back	No	New	[Status] == 'Step 2'
4.	Status	Step 3	Next	Yes	New	[Status] == 'Step 2'
5.	Status	Step 2	Back	No	New	[Status] == 'Step 3'
6.	Status	Step 4	Next	No	New	[Status] == 'Step 3'
7.	Status	Step 3	Back	No	New	[Status] == 'Step 4'

Open the context menu of *Insert* action and choose *Make Last*.

Command Name	Command Argument	Header Text	When Last Command Name	When Last Command Argument
Insert	n/a	n/a	New	n/a
Status	Step 2	Back	New	n/a
Status	Step 4	Next	New	n/a
Status	Step 3	Back	New	n/a

Change its *Header Text* to “Finish” and enter the following expression in *When Client Script* property.

[Status] == 'Step 4'

The list of actions will look as follows in the project designer.

Project Designer
exit

← Home > Controller: Employees > Action Group: **ag2 (Form)**

Action Group
Actions

This is a list of data controller actions. Actions are available in the specified UI scope of the group if all "when..." conditions are met.

Quick Find
🔍
New ▾
Preview
⬆️ Up
Down ▾
View: Actions ▾

📘 A filter has been applied. When Last Command Name equals **New**. ✕

Command Name	Command Argument	Header Text	When Last Command Name	When Last Command Argument	When Client Script (JavaScript)	When Key Selected	When HRef (Regex)	When View (Regex)	Roles	Scope
Cancel	n/a	n/a	New	n/a	n/a	n/a	n/a	n/a	n/a	Form
None	n/a	Back	New	n/a	[Status] == null [Status] == 'Step 1'	n/a	n/a	n/a	n/a	Form
Status	Step 2	Next	New	n/a	[Status] == null [Status] == 'Step 1'	n/a	n/a	n/a	n/a	Form
Status	Step 1	Back	New	n/a	[Status] == 'Step 2'	n/a	n/a	n/a	n/a	Form
Status	Step 3	Next	New	n/a	[Status] == 'Step 2'	n/a	n/a	n/a	n/a	Form
Status	Step 2	Back	New	n/a	[Status] == 'Step 3'	n/a	n/a	n/a	n/a	Form
Status	Step 4	Next	New	n/a	[Status] == 'Step 3'	n/a	n/a	n/a	n/a	Form
Status	Step 3	Back	New	n/a	[Status] == 'Step 4'	n/a	n/a	n/a	n/a	Form
Insert	n/a	Finish	New	n/a	[Status] == 'Step 4'	n/a	n/a	n/a	n/a	Form

Showing 1-9 of 9 items | Refresh

Wizard in Action

Generate the application and start entering a new *Employee* record.

User can cancel creation of a new record at any time by pushing *Cancel* button. The first “page” of the wizard displays a disabled “Back” button, representing the action with a command name of “None”. Action *None* is always displayed as disabled and provides a useful placeholder that gives the user interface consistency.

On this page, *Status* equals “Step 1”.

The screenshot shows the 'Employees' wizard at Step 1. The breadcrumb is 'Home > Employees'. The title is 'Employees'. Below the title is a message: 'Please fill this form and click OK button to create a new employees record. Click Cancel to return to the previous screen.' The form has a 'Record' dropdown and a 'View: New Employees' dropdown. A legend indicates that an asterisk (*) denotes a required field. The form fields are: 'Last Name *' with value 'Doe', 'First Name *' with value 'John', 'Title' with a red error message 'This field is required. X', 'Hire Date' with value '10/21/2011' and a calendar icon, and 'Extension' with value '8880'. At the bottom, there are 'Cancel', 'Back', and 'Next' buttons.

Press *Next*, and *Status* will change to “Step 2”.

The screenshot shows the 'Employees' wizard at Step 2. The breadcrumb is 'Home > Employees'. The title is 'Employees'. Below the title is the same message as in Step 1. The form has the same 'Record' and 'View: New Employees' dropdowns. The legend is the same. The form fields are: 'Address' with value '507-20th Ave. E.Apt. 2A', 'City' with value 'Seattle', 'Region' with value 'WA', 'Postal Code' with value '98122', and 'Country' with value 'USA'. At the bottom, there are 'Cancel', 'Back', and 'Next' buttons.

Status equals "Step 3".

Home > Employees

Employees

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

Record ▾ View: **New Employees** ▾

* - indicates a required field Cancel Back Next

Miscellaneous

Enter **John Doe's** phone number, birthday, any relevant notes

Title Of Courtesy: ▾

Birth Date:

Home Phone:

Notes:

Reports To Last Name:

Photo Path:

* - indicates a required field Cancel Back Next

Status equals "Step 4".

Home > Employees

Employees

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

Record ▾ View: **New Employees** ▾

* - indicates a required field Cancel Back Finish

Summary

Please review the summary of the new record.

Last Name: **Doe**
First Name: **John**
Title: **Sales Representative**
Title Of Courtesy: **Mr.**
Birth Date: **1/8/1973**
Hire Date: **10/21/2011**
Address: **507-20th Ave. E.Apt. 2A**
City: **Seattle**
Region: **WA**
Postal Code: **98122**
Country: **USA**
Home Phone: **(206) 555-9857**
Extension: **8880**
Notes: **Great skills and vast knowledge of our industry.**
Reports To: **Fuller**
Photo Path: **N/A**

* - indicates a required field Cancel Back Finish

Complex “When Client Script” Expressions

The configuration of navigational actions may seem to be quite a challenge at first. The truth is that there is not much value in a simple flow of wizard “pages” in our sample. A simple list of categories and ability of web pages to scroll do the job quite well.

In the real world, your *When Client Script* expression will likely be much more complex.

For example, one can imagine that different employee setup scenarios will be required, based on the employee job description or country.

You can add multiple *Status* actions activating various categories of the wizard with *When Client Script* expressions such as this.

```
[Status] == 'Step 4' && [Country] == 'USA' && [Title] == 'Senior Manager'
```

The expression can manipulate any data field if you are referencing them in square brackets.

Status Bar

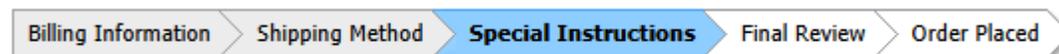
Business applications are designed to reflect complex processes of real life. Software developers and designers use the visual language of screens, menus, and various controls to approximate the business processes.

Naturally, the end users of your applications are true experts in their field and will evaluate any such approximation with a critical eye. It usually takes time for a user to understand the relationship of an application screen to an element of a real-world process.

Assigning a status to data representing a business process goes a long way towards making it easier for users to interact with your application.

The perfect example of a business process is an internet shopping cart. A seller has to collect enough information from a customer to ensure that the order is correctly placed, processed, and fulfilled. Internet customers are very impatient and will abandon their shopping cart if takes too long to complete the process or if the ordering process is confusing.

Sellers are “holding” the customer’s hand through the checkout by presenting information about the completed, current, and next steps that need to be performed. Typically this is accomplished through a progress bar that indicates the current stage of the checkout process.



A progress bar gives customers a peace of mind and helps them better understand what is going on.

The same exact care must be exercised when programming any other business process.

Start web application generator and select your project. Click *Design* button and select *Employees* data controller on All Controllers tab.

Enter the following in the *Status Bar* property of the data controller and save the changes.

Employees.createForm1.Status: null
[New Employee] > Address > Miscellaneous > Final Review >

Status: Step 1
[New Employee] > Address > Miscellaneous > Final Review >

Status: Step 2
New Employee > [Address] > Miscellaneous > Final Review >

Status: Step 3
New Employee > Address > [Miscellaneous] > Final Review >

Employees.Status: Step 4
New Employee > Address > Miscellaneous > [Final Review] >

Generate your project and observe the status bar displayed just above the description of the view *createForm1*. This illustration shows the status bar detecting that value of the *Status* field is “Step 3”.

Home > Employees

Employees

New Employee > Address > **Miscellaneous** > Final Review

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

Record ▾ View: **New Employees** ▾

* - indicates a required field

Cancel Back Next

Miscellaneous

Enter **John Doe**'s phone number, birthday, any relevant notes

Title Of Courtesy: Mr.

Birth Date: 1/8/1973

Home Phone:

Notes:

Reports To Last Name: (select)

Photo Path:

* - indicates a required field

Cancel Back Next

A status bar is defined by a collection of status values matched to the bar's topology.

A status value is defined using one of the following methods:

- Status: Value
- *ControllerName*.Status: Value
- *ControllerName.ViewId*.Status: Value

ControllerName and *ViewId* components of the status value are optional.

The second method is provided to support workflows of applications with EASE (Enterprise Application Services Engine). One workflow may define action groups, views, and virtual pages of multiple data controllers. A single status bar definition is defined per workflow.

Use the third method to create variations of status bars presented in different views. This method is also useful when there is no physical *Status* field.

If field *Status* in a given data row is empty then the status value is assumed to be null.

Status value is followed by status bar topology. Simply list logic definitions of the past, current, and future statuses; end them with a “greater than” character. The current logical status is wrapped in square brackets and separates past statuses from the future ones.

It is up to you to provide meaningful logical statuses. For example, our physical status values Step 1, Step 2, Step 3, and Step 4 are defined in status bar topology as *New Employee*, *Address*, *Miscellaneous*, and *Final Review*.

You can even make up your own logical statuses to help users establish a relationship of a visual presentation with the real world.

Home > Employees

Employees

Job Application Accepted > New Employee > Address > **Miscellaneous** > Final Review > Orientation

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

Record ▾ View: **New Employees** ▾

* - indicates a required field

Cancel Back Next

Miscellaneous

Enter **John Doe**'s phone number, birthday, any relevant notes

Title Of Courtesy

Birth Date

Home Phone

Notes

Reports To Last Name

Photo Path

* - indicates a required field

Cancel Back Next

Note that the *Status Bar* feature is available in *Premium* and *Unlimited* editions only.

Conclusion

Code On Time web applications offer first class high end features that require little to no programming. *Wizards* and *Status Bars* are a great example.

EASE workflows will make it possible creating user-specific views and pages enhanced with custom actions and status bars at runtime without changing the application.