

The [Page wizard](#) is a set of web parts that allow you to guide users through a sequence of steps on the website. The wizard uses standard pages to define the content of steps. You can extend the logic of the wizard by [customizing the code](#) of the web parts placed onto the wizard steps, for example:

- Define validation rules for web parts — conditions that must be fulfilled before users can move forward from steps containing the web part
- Perform custom actions when moving between steps

To interact with the page wizard from the code of your web parts, override the methods listed below. The methods handle events that occur during the life cycle of each wizard step:

Wizard step event	Description
LoadStep	Occurs when a user opens the step containing the web part.
StepLoaded	Occurs after all web parts inside the given step's content have finished the <i>LoadStep</i> event.
ValidateStepData	Allows you to define step validation requirements for the web part. The event occurs when: <ul style="list-style-type: none"><li>• a user moves to the <b>next</b> step by clicking the <i>Page wizard button</i></li><li>• the <i>Page wizard step action</i> web part automatically skips the step (if validation is enabled)</li></ul> <b>Note:</b> Moving between steps via the <i>Page wizard navigation</i> web part does not trigger validation.
SaveStepData	Occurs when moving to the next step after the validation is complete.
StepFinished	Occurs after all web parts in the given step's content have finished the <i>SaveStepData</i> event.

 The base handlers are defined in the **CMSAbstractWebPart** class, so you can override the page wizard event handlers in the code of any web part.

All of the event handlers provide the **StepEventArgs** parameter, which allows you to access the data of the current page wizard step.

### Cancelling step transitions

You can interrupt transitions to the next wizard step based on custom conditions. Set the **CancelEvent** property of the *StepEventArgs* parameter during the *ValidateStepData*, *SaveStepData* or *StepFinished* events.

For example, add a custom validation condition into the *ValidateStepData* handler, and set the *CancelEvent* property to true if the validation fails.

```
e.CancelEvent = true;
```

### Skipping steps

If you need to completely skip a page wizard step under certain conditions, set the **Skip** property of the *StepEventArgs* parameter to true in the *LoadStep* or *StepLoaded* handler.

```
e.Skip = true;
```

## Example - Creating a form for wizard steps


The following example demonstrates how to create a customized version of the **On-line form** web part for use inside page wizard steps. The custom form:

- Integrates validation of the form's data into the validation of the wizard step

- Saves the form's data when the wizard moves to the next step

See also: [Forms](#)

## Cloning the form web part

1. Open the **Web parts** application.
2. Select the **Forms -> On-line form** web part in the tree.
3. Click **Clone web part** () and enter the following values:
  - **New object display name:** Wizard form
  - **New object code name:** WizardForm
  - **Clone web part to category:** Forms
  - **Clone web part files:** yes (checked)
  - **Cloned web part file name:** BizForms/WizardForm.ascx
4. Click **Clone**.

The system creates a copy of the existing web part and its code files (ASCX and CS) using the specified names.

## Modifying the web part code

1. Open your web project in Visual Studio.
2. Edit the code behind of the cloned web part: `~/CMSWebParts/BizForms/WizardForm.ascx.cs`



### Important

If you installed Kentico as a web application, the clone's code files will not be visible right away, since they are not included in the project. In this case:

- a. Click **Show all files** at the top of the Solution Explorer.
- b. Right-click the **WizardForm.ascx** file in the BizForms folder.
- c. Click **Include in Project**.

3. Add the following code into the **SetupControl** method (in the *else* branch where *StopProcessing* is false):

```
// Hides the form's default submit button
viewBiz.SubmitButton.Visible = false;
```


4. Add the following overrides for the **ValidateStepData** and **SaveStepData** handlers of page wizard events:

```
protected override void ValidateStepData(object sender, CMS.Base.Web.UI.
StepEventArgs e)
{
    base.ValidateStepData(sender, e);

    // Checks whether the form's data is valid
    if (!viewBiz.ValidateData())
    {
        // Stops the wizard from moving to the next step if the form data
is invalid
        e.CancelEvent = true;
    }
}

protected override void SaveStepData(object sender, CMS.Base.Web.UI.StepEventArgs
e)
{
    base.SaveStepData(sender, e);

    // Saves the data of the form (replaces the form's submit button)
    if (!viewBiz.SaveData(null))
    {
        // Stops the wizard from moving to the next step if the form's
data wasn't saved successfully
        e.CancelEvent = true;
    }
}
```

 **Tip:** The sample code is only a demonstration of the available event handlers. You do not need to implement the **ValidateStepData** handler for forms in real world scenarios. The **viewBiz.SaveData** method called in the **SaveStepData** handler includes validation (returns a false value if the form's data is invalid).

#### 5. Save the file.

You can now add the customized form web part into the content of page wizard steps. When moving to the next step, the wizard:

1. Validates the form data (according to the settings of the form's fields)
2. Creates a new [data record](#) for the given form (if the form data is valid)

## Resetting the page wizard

The page wizard tracks which steps individual users have passed through (the system stores the information in the HTTP session). If the **Restrict step order** property of the **Page wizard manager** web part is enabled, the wizard forces users to move through the steps in sequence:

- Users can only move forward through the wizard's steps using the **Next** button
- The **Page wizard navigation** web part displays unvisited steps as inactive items (without links)
- If a user attempts to access an unvisited step through the URL of the step page, the wizard opens the last visited step instead

You can use the API to reset the wizard from the code of any web part placed onto a step. Access the **DocumentWizardManager** property (inherited by all web parts from **CMSAbstractWebPart**) and call the **ResetWizard** method:

```
DocumentWizardManager.ResetWizard();
```

When executed, the method clears the history of visited steps for the user viewing the page.



**Note:** The wizard automatically resets when users click the **Next** button on the last step (if the **Final step URL** property is filled for the **Page wizard manager** web part).