

Wizard Web Server Control Overview

Use the [Wizard](#) control to simplify many of the tasks that are associated with building a series of forms to collect user input.

This topic contains:

- [Scenarios](#)
- [Background](#)
- [Code Examples](#)
- [Class Reference](#)

Scenarios

Collecting user input by using forms is a recurring task in Web development. A group of forms that is used to accomplish a task is often called a *wizard*.

The ASP.NET [Wizard](#) control simplifies many of the tasks that are associated with building multiple forms and collecting user input. The [Wizard](#) control provides a simple mechanism that allows you to easily build steps, add a new step, or reorder the steps. You can build linear and non-linear navigation and customize the control's user navigation without writing code.

[Back to top](#)

Background

Building a series of interconnected forms to break up data collection is a common practice. You can accomplish this by managing the navigation between forms, the data persistence, and the state management in each step. With the [Wizard](#) control, you use discrete steps to collect data, which allows users to navigate between steps at their discretion and creates a simpler user experience. As a developer, you do not have to worry about making your data persist across pages because the control maintains state while the user completes the various steps.

Wizard Steps

The [Wizard](#) control uses steps to delineate different sections of user data input. Each step within the control is given a **StepType** to indicate whether it is the beginning step, intermediate step, or completion step. The wizard can have as many intermediate steps as needed. You can add different controls, such as a [TextBox](#) or [ListBox](#) control, to collect user input. When you reach the [Complete](#) step, all of your data is accessible. The following code example illustrates the [Wizard](#) control with two steps.

```
<asp:Wizard ID="Wizard1" Runat="server">
    <WizardSteps>
```

```
<asp:WizardStep Runat="server" Title="Step 1">
</asp:WizardStep>
<asp:WizardStep Runat="server" Title="Step 2">
</asp:WizardStep>
</WizardSteps>
</asp:Wizard>
```

Within each step, you can add controls and labels, and accept user data. The [Wizard](#) control will help manage which step to display, and help maintain the collected data.

Wizard Navigation

The [Wizard](#) control features both linear and non-linear navigation. The control's state management allows the user to move forward and backward between steps, and it allows the user to select individual steps at any point, as long as the sidebar is displayed. You can customize the text for navigation in the control's root **asp:Wizard** element by using the [StepNextButtonText](#), [StepPreviousButtonText](#), and [FinishCompleteButtonText](#) properties.

```
<asp:Wizard ID="Wizard1" Runat="server"
  StepNextButtonText=" Next >> "
  StepPreviousButtonText=" << Previous "
  FinishCompleteButtonText=" Done! ">
```

Customizing Other Wizard Control Aspects

The [Wizard](#) control automatically displays a title and the control's current step. The title is customized with the [HeaderText](#) property. You can adjust the header's template by using the [HeaderTemplate](#) property.

You can optionally display a cancel button by setting the [DisplayCancelButton](#) property to **true**.

The [Wizard](#) control's [NavigationButtonStyle](#) property provides an easy way to set all of the buttons to a common style, while providing the flexibility to customize each button individually. The [NavigationButtonStyle](#) property applies to all of the buttons rendered. However, you can override this style by setting the individual button's style property.

Navigation Templates

The [Wizard](#) control supports templates that allow you to further customize the interface of the control with the [StartNavigationTemplate](#), [FinishNavigationTemplate](#), [StepNavigationTemplate](#), and [SideBarTemplate](#) properties.

Wizard Control Events

You can customize the behavior of the [Wizard](#) control by using custom code and events.

For example, you can intercept the [NextButtonClick](#) event, which is raised when the user clicks the **Next** button and captures the data of the current step. The [WizardNavigationEventArgs](#) parameter passed to this event includes the [CurrentStepIndex](#) and [NextStepIndex](#) properties, enabling you to customize the behavior of the control based on the current and next steps, or to cancel the navigation when the **Next** button is clicked.

Similarly, you can customize the behavior of the **Previous** and **Finish** buttons by using the [PreviousButtonClick](#) and [FinishButtonClick](#) events. Or you can perform cleanup when the **Cancel** button is clicked by using the [CancelButtonClick](#) event.

[Back to top](#)

Formatting and CSS Styles

By default, the [Wizard](#) control layout is rendered using an HTML table. However, using tables to lay out visual elements on a page is not semantically correct and might not be compatible with cascading style sheet (CSS) rules. To address this issue, the [Wizard](#) control can render markup that uses CSS for layout instead of using tables. This makes it easier to format the control. The [Wizard](#) control supports a [LayoutTemplate](#) property that you can use to specify the layout of visual elements. For more information, see [LayoutTemplate](#).

The [Wizard](#) control also supports the ability to use a [ListView](#) control instead of a [DataList](#) control inside its sidebar area. For more information, see [SideBarTemplate](#).

Code Examples

[Walkthrough: Creating a Basic ASP.NET Wizard Control](#)

[Walkthrough: Advanced Use of the ASP.NET Wizard Control](#)

[Back to top](#)

Class Reference

The following table lists the key classes that relate to the [Wizard](#) control.

Member	Description
Wizard	The main class for the control.
WizardNavigationEventArgs	Provides the data that is needed for the various navigation events in a Wizard control.
WizardNavigationEventHandler	Represents the method that will handle navigation events in a Wizard control.
WizardStep	Represents a basic step that is displayed in a Wizard control.
WizardStepCollection	Represents a collection of WizardStep objects that contain the user interface for each step, as defined by the page developer.
WizardStepType	Specifies the types of navigation UI that can be displayed for a step in a Wizard control.

[Back to top](#)

See Also

Reference

[CreateUserWizard](#)

© 2017 Microsoft