DetailsView Web Server Control Overview

The DetailsView control displays a single record from a data source, where each data row represents a field in the record. It is often used in combination with a GridView control for master/detail scenarios.

This topic contains:

- Background
- Code Examples
- Class Reference

Background

The DetailsView control gives you the ability to display, edit, insert, or delete a single record at a time from its associated data source. By default, the DetailsView control displays each field of a record on its own line. The DetailsView control is typically used for updating and inserting new records, often in a master/detail scenario where the selected record of the master control determines the record to display in the DetailsView control. The DetailsView control displays only a single data record at a time, even if its data source exposes multiple records.

The DetailsView control relies on the capabilities of the data source control to perform tasks such as updating, inserting, and deleting records. The DetailsView control does not support sorting.

The DetailsView control can automatically page over the data in its associated data source, provided that the data is represented by an object supporting the ICollection interface or that the underlying data source supports paging. The DetailsView control provides the user interface (UI) for navigating between data records. To enable paging behavior, set the AllowPaging property to **true**.

You select a particular record from the associated data source by paging to that record. The record displayed by the DetailsView control is the current selected record.

Data Binding with the DetailsView Control

The DetailsView control provides you with these options for binding to data:

- Data binding using the DataSourceID property, which allows you to bind the DetailsView control to a data source control. This is the recommended approach because it allows the DetailsView control to take advantage of the capabilities of the data source control and to provide built-in functionality for updating and paging.
- Data binding using the DataSource property, which allows you to bind to various objects, including ADO.NET datasets and data readers. This approach requires you to write the code for any additional functionality such as updating and paging.

When you bind to a data source using the DataSourceID property, the DetailsView control supports two-way data binding. In addition to the control displaying data, you can enable the control to automatically support insert, update, and delete operations on the bound data.

Working with the DetailsView Control Data

The DetailsView control binds to a data source control, which in turn handles the tasks of connecting to a data store and returning the selected data. Binding the DetailsView control to data is as simple as setting the DataSourceID property declaratively. You can also bind to a data source in code.

To enable editing, set the AutoGenerateEditButton property to **true**. The DetailsView control will then render an **Edit** button in addition to the data fields. Clicking the **Edit** button causes the DetailsView control to enter edit mode. In edit mode, the DetailsView control's CurrentMode property changes from **ReadOnly** to **Edit** and each field of the control renders its edit UI, such as a text box or check box. You can also customize the edit UI by using styles, DataControlField objects, and templates.

Note
For the DetailsView control to support editing, the bound data source must support update operations on the data.

You can configure the DetailsView control to display a **Delete** and an **Insert** button so that you can delete the corresponding data record from the data source or insert a new data record. Similar to the AutoGenerateEditButton property, when the AutoGenerateInsertButton property is set to **true** on the DetailsView control, it renders a **New** button. When the **New** button is clicked, the DetailsView control's CurrentMode property changes to **Insert**. The DetailsView control renders appropriate UI input controls for each bound field, unless the **InsertVisible** property of the bound field is set to **false**.

Customizing the DetailsView Control User Interface

The DetailsView control supports a Fields collection property, which contains DataControlField objects of type BoundField, CommandField, or HyperLinkField. This is similar in function to the Columns collection of the GridView control, except that the DetailsView control renders each field as a row instead of a column.

As with the GridView control, you can customize the UI of the DetailsView control by using style properties such as the HeaderStyle, RowStyle, AlternatingRowStyle, CommandRowStyle, FooterStyle, PagerStyle, and EmptyDataRowStyle properties.

The DetailsView control offers you additional customization through templates, which give you more control over the rendering of certain elements. You can define your own EmptyDataTemplate, HeaderTemplate, FooterTemplate, and PagerTemplate properties for the DetailsView control. You can also create a template for an individual field by adding a TemplateField object to the Fields collection.

The DetailsView control exposes several events that you can handle to execute your own code. The events are raised before and after the insert, update, and delete operations of the associated data source control. You can also write handlers for the ItemCreated and ItemCommand events.

Note
The event model of the DetailsView control is similar to that of the GridView control. However, the DetailsView control does not support a selection event, because the current record is always the selected item.

Back to top

Code Examples

Walkthrough: Editing and Inserting Data in Web Pages with the DetailsView Web Server Control

Paging in a DetailsView Web Server Control

Modifying Data Using a DetailsView Web Server Control

Creating a Custom Row in a DetailsView Web Server Control

Back to top

Class Reference

The following table lists the key classes that relate to the DetailsView control.

Member	Description
DetailsView	The main class for the control.

Back to top

See Also

Reference
GridView Web Server Control Overview
Concepts
ASP.NET Web Server Controls Templates
Other Resources
DetailsView Web Server Control Events
ASP.NET Data Access Options
ASP.NET Data Access Content Map