

## Creating Master/Detail Pages in ASP.NET – Single Tier, using Visual Web Developer Tools

Master/detail pages are a common interface on the web – the master list displays several records and the detail display shows information about a single record. The master and detail can be displayed on the same page or on separate pages. The steps below are heavily drawn from *Teach Yourself ASP.NET in 24 Hours* by Scott Mitchell who also has online tutorials on create master/detail pages (<http://msdn.microsoft.com/en-us/library/bb331182.aspx>) - page references are to the book.

The approach below uses Visual Web Developer tools and a single tier rather than the writing of code and a multiple tier/layer (data access, business logic, presentation) approach. In particular, in step 5, an instance of a SqlDataSource is placed on the webpage. In a multiple tier approach, this would be an instance of the ObjectDataSource. This will be explored in a separate writeup.

1. Create the database (pp. 290-291) – To create a new SQL Server Express Edition database with Visual Web Developer, create a new website using the Empty ASP.NET template. To add a database, go to Solution Explorer, right click on the website name, choose Add New>SQL Server Database. You can respond yes about putting the database in the App\_Data folder.
2. Creating Database Tables (pp. 292-297) – In Visual Web Developer, using Database Explorer, right click the Tables folder and choose the Add a New Table option. Enter information for the field name, datatype, and Allow Null columns.
  - a. To designate a field as the primary key, select the row showing the appropriate field name and then click the primary key icon in the toolbar or right click the row and choose Set Primary Key.
  - b. To mark a field as auto-increment, select the appropriate row column and then in the Column Properties at the bottom, find the Identify Specification property, expand it and change the (Is Identity) option to Yes.
3. Adding Data to the Tables (pp. 298-300) - Right click on the table in the Database Explorer and select the Show Table Data option and enter the data. Where you see red exclamation point icons, the record has yet to be created so click in the next record or Tab to it, to save the record.
4. Create a new web page - In Visual Web Developer, in the Solution Explorer, right click on the website name>Add>New Item>Web Form and give it an appropriate name at the bottom of the page, e.g. default.aspx

5. **SQLDataSource (pp. 308-313)** - Drag an instance of the SQLDataSource from the Toolbox onto the page. In Design view, click on the Smart Tag at the right of the control and click on the Configure Data Source option which will launch the Configure Data Source Wizard.
  - a. Choose Your Data Connection – The drop down list contains those databases listed in Database Explorer and should include the database created in step 1 above. Choose this database. If you click the plus sign next to the Connection string label, you will see the connection string used to access the selected database. Click the Next button. The first time you do this, you will be prompted to save the connection information in the web.config file. Saving this information there makes subsequent changes easier since the connection string will appear in one file rather than in the multiple files that access the database.
  - b. Configure the Select statement – This window allows one to specify which fields you want to select – to select all, choose the asterisk. You can specify a WHERE clause, or ORDER BY clause but for now, we will leave these blank. On the next screen you can test the Query to make sure it is functioning properly and that the data has been correctly entered and saved in the database. Click Finish.
  - c. If this is the first SQLDataSource you have added to this page, the SQLDataSource will be given an ID of SQLDataSource1. If you like, you can rename it to something more descriptive either by switching to code view and changing the specified ID, or selecting the control and changing the ID property in the Properties panel.
  
6. **Master display using GridView (pp. 335-347)** – With the page in Design view, drag a GridView control from the Toolbox's Data section onto the page. In the Smart Tag, there's a Choose Data Source option which contains the data source controls on the page. Go ahead and select the SQLDataSource you created as the GridView's data source (the SQLDataSource will be named SQLDataSource1 unless you have renamed it). When you do this, the GridView display will be updated to reflect the field names specified in the data source, but the data will not be visible (until the page is viewed in the browser).
  
7. **Customizing the Appearance of the GridView** – one can customize at several levels:
  - GridView level – affects all of the data,
  - Field level – formats a particular field, and
  - Row level – formats a particular row

These various levels can be set through the AutoFormat dialog box, the Fields dialog box, or the Properties window. To do field-level formatting, click on the Smart Tag for the GridView and click on the Edit Columns link. Currently all the fields used by the GridView are BoundFields, i.e. fields which display the value from a particular column in the data source control. In the available fields list at the top left, there are other types of fields – the HyperLinkField displays a hyperlink in each row and the ButtonField displays a button in each row.

8. Displaying HyperLinks with the HyperLinkField (pp. 420-425) – One can specify static or dynamic values for the HyperLinkField. To specify dynamic values,
  - a. For the text of the URL, use the DataTextField and DataTextFormatString properties
  - b. For the URL, use the DataNavigateURLFields and DataNavigateURLFormatString properties

The DataTextField and DataNavigateURLFields properties specify what database column values are used in the text and URL of the rendered link. Once these are specified, one can use DataTextFormatString and DataNavigateURLFormatString properties to surround the database value with static text, if needed. The string {0} (curly brackets, and this is a zero, not an “O”) indicates where to place the dynamic value (see <http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listcontrol.datatextformatstring.aspx> and <http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.hyperlinkfield.datanavigateurlformatstring.aspx> )

The DataTextField can be set by choosing from the drop down list showing database fields but the DataNavigateURLFields property must apparently be typed in.

9. Showing One Record at a Time with DetailsView (pp. 347-351) – In GridView, since the fields are each displayed in a separate column, one generally won’t want to display more than a few columns. To display more or all of the fields for a record, the DetailsView is useful since it only displays one record at a time and each field appears in a separate row thus permitting the convenient display of more fields than GridView.

Drag an instance of the DetailsView onto the page. Click the smart tag and for Choose Data Source, choose the same data source as before (SqlDataSource1). The DetailsView displays the first record returned by its SqlDataSource control. To allow paging through records with the DetailsView, click the Smart Tag and then click the Enable Paging checkbox. By default, the paging interface uses page numbers to allow the user to jump to a particular record.

10. Delete, Update, and Inserting Records - Both the GridView and DetailsView controls can be used to delete and edit existing data and the DetailsView can also be used to insert records. If the SqlDataSource has been configured to support inserting, updating, and deleting, the GridView and DetailsView controls can utilize this functionality.
11. Enable delete, update, and insert functionality in the SqlDataSource (pp. 360-361) - Click on the SmartTag for the SqlDataSource and then Configure Data Source. On the Choose Your Data Connection window, ConnectionString should be showing (because you created this previously), choose Next>Advanced>check the checkbox for “Generate Insert, Update, and Delete statements” and use Optimistic Concurrency. Click through to Finish.

12. Editing and Deleting Data with the GridView (p. 366-384) – Click the Smart Tag for the GridView. Now that the SqlDataSource has been enabled for Insert, updating, and deleting, extra checkboxes will appear in the SmartTag.
- a. If you check Enable Deleting, a column of Delete links will appear in the GridView.
  - b. If you check Enable Editing, a column of Edit links will appear in the GridView. When the user clicks the Edit link, the row becomes editable and the row's fields turn into text boxes and the Edit button is replaced by two new buttons Update and Cancel. Update will save the changes and Cancel will return the editable row to its read-only display without saving any changes.
  - c. The appearance of deleting and editing can be customized in various ways.
13. Inserting, Editing, and Deleting Data with the DetailsView (pp. 385-387) - Once the SqlDataSource has been modified to provide support for these functions, the DetailsView Smart Tag will have checkboxes for each of these capabilities. If one views this page through a browser, there will be a New link which when clicked, displays the DetailsView in an editable mode for a new record with an Insert link which when clicked, will add the record – the change should immediately appear in the GridView display.