SYSTEMS DESIGN / CAPSTONE PROJECT
MIS 413

User Guide 4.4

Grid Views – Link Buttons, Check Boxes, Radio Button Lists

Link Buttons inside a Grid View (to determine who was clicked)

At times there is a need to have a select button or a link button in your grid. When the users clicks on the link button the developer needs to identify which 'row' and the PK of the row that has been selected to perform an action.

This hint assumes you want to click on a person's name and then do some action.

a) build the grid view as normal (the stored procedure has returned a field named **'full name'** which we will click. Hint in SP: lastname + ', ' + firstname as fullname

b) in the grid view, convert the column that has the full name to a template column

c) in edit templates, click on the template column full name, and then the item template option. Delete the label and slide a link button into the template, name this link button (_FullName for our example)

d) right click on the link button and then edit data bindings, change the command argument property to the primary key field. Hint: commandArguement= '<%# eval("personid") %>'

e) end template editing and add a an 'event' for the grid view

f) click on properties for the grid, select the event icon (lightning bolt) and select **row command** and double click

g) in the subroutine that opens you can check that you have returned the PK by checking:
response.write (e.commandarguement)
Once you have verified the proper PK, call the subroutine to perform your desired action.

Check Boxes inside a grid (updating all by looping through the rows in the grid), i.e. updating attendance for an event

This example will read each row of the grid, inspecting the checkbox for its value and then updating the database based on the PK for that row.

a) insure your grid has Check Box (not a CHECK BOX LIST) column, normally generated for your for YES/NO bit columns in the database

©2014, Thomas Janicki

b) convert the check box column to a template column and provide the check box a name you will reference (_alumni for our example)

c) make the check box in the item template mode enabled = true (if needed)

d) place an update button under (or above or both!) the grid, this update button will loop through each row of the grid to determine if an item has been checked add then update the db as appropriate.

e) Coding under the update button

```
  //build a link the the name/pwd/user for your particular database
        //note below the [mis413...] should be the name of your connection string,
open your webconfig file and find the connection string name
        string dbConn =
System.Configuration.ConfigurationManager.ConnectionStrings["mis413ConnectionString"].
ConnectionString;
        //build a connection to the database
        SqlConnection conn = new SqlConnection(dbConn);

        //use the above connection to execute a particular stored procedure [substitue
your personInsert stored procedure name]
        using (SqlCommand cmd = new SqlCommand("[tnjPersonUpdateAlumni]", conn))
        {
            cmd.CommandType = CommandType.StoredProcedure;
        //build the parameters (input items) that the stored procedures needs
        //initially build the 2 parameters needed with fake data that represents the
data type, i.e. if string would put ,a instead of ,0
            cmd.Parameters.AddWithValue("@personID", 0);
            cmd.Parameters.AddWithValue("@alumni", 1);

           //now begin to loop through the grid,
            try
            {
                int pk = 0;
                int thisRowIndex = 0;
                conn.Open();
                // the following will loop one row at a time, find the value of the
alumni checkbox and update the database table
                CheckBox chk = default(CheckBox);
                foreach ( GridViewRow oneRow in this._checkboxdemo.Rows)
                 {
                    chk = oneRow.FindControl("_alumni") as CheckBox;
                    thisRowIndex = oneRow.RowIndex;
                    pk =
Convert.ToInt32(_checkboxdemo.DataKeys[thisRowIndex].Values[0]);
                    //put this line to test if you are looping through the grid
                   Response.Write(pk + " " + chk.Checked + "<br/>");
                    //remark out the above line and bring in the following line once
you know the loop is working (unremark them)
                   //cmd.Parameters["@personID"].Value = pk;
                   //cmd.Parameters["@alumni"].Value = chk.Checked;
                   //cmd.ExecuteNonQuery();
                }
                this._message.Text = "Alumni Status was Updated";
            }
            catch (SQLException ex)
            {
                this._message.Text = "Error on updating alumni " + ex.Message;
```

©2014, Thomas Janicki

```
            }
        }
```

Radio Button Lists inside grids

This above example will work for radio button lists inside a grid. The only changes would be to

Replace CheckBox chk = default(CheckBox);
With: RadioButtonList rad = default(RadioButtonList);

Replace: chk = oneRow.FindControl("_alumni") as CheckBox;
With rad = oneRow.FindControl(“_name”) as RadioButtonList

Use a SelectedValue versus a checked when passign the value to the Stored Procedure

©2014, Thomas Janicki