SYSTEMS DESIGN / CAPSTONE PROJECT
MIS 413

User Guide 5.2

Check Box Lists

Retreiving a value from a checkbox list

1. A checkbox list is built the same way as a radio button list or a drop down list. You should bind the checkbox to a stored procedure and set the **value** and **text** properties.

2. However to indicate which item(s) the user has selected is more tricky as the **.selectedvalue** property only **returns the 'first item' checked** in the list. Thus we must loop through the check box list looking for each item that has been 'checked'. This is easy if you think that the check box list is really a 'list' of items and we can review/inspect each item one at a time to determine if it has been 'checked' or selected.

3. Once you have bound the check box list to a stored procedure, you generally will have an Update or Add button on the web form. The following code will inspect the properties of every item in the check box list:

```
foreach ( ListItem item in  this.CheckBoxList1.Items)
      {
          if (item.Selected)
         {
               Call the sub or function you need to update a database etc.
                    (or insert the proper code)
               The value (index) would be  oneItem.value and you generally
               Pass this value to the sub or function to update the database
          }
      }
```

Specific Example:

```
  // set up the name and parameters of the stored procedure before
  // going into the loop

string dbConn =
System.Configuration.ConfigurationManager.ConnectionStrings["mis413ConnectionStri
ng"].ConnectionString;

//build a connection to the database
SqlConnection conn = new SqlConnection(dbConn);
using (SqlCommand cmd = new SqlCommand("[personMajorsInsert]", conn))
      {
cmd.CommandType = CommandType.StoredProcedure;
//build the parameters (input items) that the stored procedures requires
 //note below that iPK is the personID of the person who has signed into the
 //system, you can retrieve this from the cookie set on the login page
 //look back on your menu page to see how to retrieve data from the cookie
cmd.Parameters.AddWithValue("personID", iPK)
// add more parameters as needed by your stored procedure
cmd.Parameters.AddWithValue("majorID", "abc")
```

```
            // open the database and run the stored procedure, also catch any errors
and display them in your _message label
            try
            {
                conn.Open();
                foreach ( ListItem item in  this._CheckBoxName.Items)
                {
                    if (item.Selected)
                    {
                       // gets the PK for the Major ID selected
                        cmd.Parameters["@majorID"].Value = item.SelectedValue;
                        cmd.ExecuteNonQuery();
                    }
                }

                this._message.Text = "Majors were updated";
            }
            //if there are any errors with the store procedure, display them in the
message label
            catch (SqlException ex)
            {
                this._message.Text = "Error on Inserting New Majors " + ex.Message;
            }
```

Populating the checkbox list with the values that have already been selected by pulling data from the database table containing that data (i.e. personMajors)

The reverse process to indicate in advance which items have been checked when you 'load the checkbox' to the web form is somewhat tricky.

1. first bind the checkbox list to a stored procedure which shows all items, insure the value and text fields have been properly built (this SP is usually the select all from your valid table)

2. once the checkbox list is databound you want it to call another stored procedure to determine which items in the checkbox list have already been selected. Click on the properties for the checkboxlist and then select the EVENT – ***databound*** for the check box list

    - Under the subroutine that is created for the databound event, write the code to execute another stored procedure that selects all elements in the checkbox list that have been 'checked' or have a value of true. For this example it is the select all majors from the personMajors table for one person (i.e. Select majorID from personMajors where personID= @personID)

3. Now loop through the data reader and find the matches in the checkbox list as in:

    Start your Try/Catch as you normally have done
    i.e name of stored procedure, open the db etc.
    the following picks up at the conn.open command

```
            try
            {
                conn.Open();
                 SqlDataReader dtrReader = cmd.ExecuteReader();
                do {
                        foreach (ListItem item in this.CheckBoxList1.Items)
                        {
                            if (item.Value ==  dtrReader["majorID"])
                            {
                                item.Selected = true;
                                break;
                            }
                } while( dtrReader.Read());

            }
            //if there are any errors with the store procedure, display them in the
message label
            catch (SqlException ex)
            {
                this._message.Text = "Error on retrieving current majors for this
person " + ex.Message;
            }
```