



In this lab, you are required to save changes to both parent and child tables if no errors in the tables.

Estimated time to complete this lab: 120 minutes.

A) Creating a new Project and Form

- In the Visual Studio 2010 environment, select **File | New Project** from the menu.
- In the **New Project** dialog box, select **Visual C# | Windows** on the left and then **Windows Application** on the right.
- Enter a **Name** for the new project and select a save **Location** and click **OK**.
- Copy the *Northwind* database file from our course page and save it in the **H:\compsci280** folder.

B) Creating and Configuring a Data Connection, Tables Adapters and a Dataset

- To connect your application to a database, create a table adapter and a DataSet using the Data Source Window
 - Select **Data | Add New Data Source** menu command
 - Select **Database** on the **Choose a Data Source Type** page.
 - Choose <ACCESS.H:\compsci280\Northwind.accdb>
 - Click **Next** on the **Save connection string to the Application Configuration file** page.
 - Expand the Tables node on the **Choose your Database Objects** page.
 - Select the **Products** and the **Order Details** tables and click Finish.
 - Note that the configuration wizard also creates the Relation between Products and Order Details.
- To add a Foreign key constraint
 - Double-click NorthwindDataSet.xsd in the Solution Explorer.
 - Right-click on the relation and select Edit Relation. The DataSet Designer displays the Relation dialog box.
 - Select “Both Relation and Foreign Key Constraint”,
 - Set Update, Delete and Accept/Reject rules to Cascade, and then click OK.
- Select **Build | Build <projectName>** menu command

C) Creating controls

- Select **View | Designer** menu command to open the Form Designer Window, and then Open the **Toolbox**
- Drag the NorthwindDataSet from the <projectName> Components section onto the form designer surface.
 - Set the following properties of the DataSet using the **Properties** window
 - Name: NorthwindDataSet
- Drag the ProductsTableAdapter from the <projectName> Components section onto the form designer surface
 - Set the following properties of the TableAdapter component using the **Properties** window
 - Name: ProductsTableAdapter
- Drag a BindingSource component from the Data section of the **Toolbox** onto the form designer surface.
 - Set the following properties of the BindingSource component using the **Properties** window
 - Name: ProductsBindingSource
 - DataSource: NorthwindDataSet
 - DataMember: Products
- Drag the Order_DetailsTableAdapter from the <projectName> Components section onto the form designer surface
 - Set the following properties of the TableAdapter component using the **Properties** window
 - Name: Order_DetailsTableAdapter
- Drag a BindingSource component from the Data section of the **Toolbox** onto the form designer surface.
 - Set the following properties of the BindingSource component using the **Properties** window
 - Name: Order_DetailsBindingSource
- Drag 3 button controls from the **Toolbox** and drop them onto the form
 - Right-click the button control and select the **Properties** menu command
 - Set the following properties of the button control using the **Properties** window
 - Name: btnLoad
 - Text: Load
 - Right-click the other button control and select the **Properties** menu command
 - Set the following properties of the button control using the **Properties** window
 - Name: btnCancelAll

- Text: Cancel All
 - Right-click the last button control and select the **Properties** menu command
 - Set the following properties of the button control using the **Properties** window
 - Name: btnSave
 - Text: Save
- Drag a data grid view control from the **Toolbox** and drop it onto the form
 - Right-click the data grid view control and select the **Properties** menu command
 - Set the following properties of the data grid view control using the **Properties** window
 - Name: ProductsDataGridView
 - DataSource: ProductsBindingSource
- Drag another data grid view control from the **Toolbox** and drop it onto the form
 - Right-click the data grid view control and select the **Properties** menu command
 - Set the following properties of the data grid view control using the **Properties** window
 - Name: Order_DetailsDataGridView

D) Code

- Double-click the Load button control. Visual Studio 2010 will navigate to the code view of Form1 (Form1.cs) and create an event handler for the **Click** event of the button
 - Add a Try-Catch block statement in the btnLoad_Click method.

```
try {
} catch (Exception ex) {
    MessageBox.Show(ex.ToString(), "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

- Complete the Try-catch block above
 - Fill the **Products** and **Order Details** tables in order into the NorthwindDataSet dataset using the TableAdapters created above.
 - Set the DataSource and DataMember properties of the OrderDetailsBindingSource component
 - DataSource = ProductsBindingSource
 - DataMember = "ProductsOrder Details"
 - Set the DataSource property of the Order_DetailsDataGridView
 - DataSource = Order_DetailsBindingSource
- Write code to handle the **ColumnChanging** event for the Order Details table
 - Complete the OnColumnChanging procedure to check for invalid value of the **Quantity**
 - Set the Column Error of the DataRow object
 - To "Quantity must be greater than 0." if the proposed value is less than or equals to zero
 - To "" if it the proposed value is greater than zero.

```
private void Column_Changing(object sender, DataColumnChangeEventArgs e) {
    if (e.Column.ColumnName == "Quantity" )
        if (Convert.ToInt32(e.ProposedValue) <= 0)
            ...
        else
            ...
}
```

- Double-click in design area of **Form1**, Visual Studio 2010 will navigate to the source file of **Form1** and will create an event handler for the form load event (**Form1_Load**).

```
private void Form1_Load(object sender, EventArgs e) {
    ...
}
```

- Connect the event handler to the event. Place the following code within the form's **Load** event.
- ```
northwindDataSet1.Order_Details.ColumnChanging ...
```
- Double-click the Cancel All button control. Visual Studio 2010 will navigate to the code view of Form1 (Form1.cs) and create an event handler for the **Click** event of the button
    - Reject all changes from the DataSet
  - Double-click the Save button control. Visual Studio 2010 will navigate to the code view of Form1 (Form1.cs) and create an event handler for the **Click** event of the button

```
private void btnSave_Click(object sender, EventArgs e) {
 ...
}
```

- Validate the value of the control losing focus
- End edits of the two BindingSource components
- Check for any changes in the NorthwindDataSet DataSet
- Check for any errors in the NorthwindDataSet DataSet
- Declare a Order\_DetailsDataTable called dtDeleted to retrieve the changes for Child Deletes
 

```
NorthwindDataSet.Order_DetailsDataTable deletedOrders =
(NorthwindDataSet.Order_DetailsDataTable)northwindDataSet1.Order_Details.GetChanges(DataRowState.Deleted);
```
- Declare a Order\_DetailsDataTable called dtChanged to retrieve the changes for Child Updates and Inserts
- Add a try-catch block to catch any exception may be generated when saving the changes to the actual data source

```
try{
```

```

...
}
catch (Exception ex)
{
 MessageBox.Show(ex.ToString(), "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
finally
{
 if (deletedOrders != null)
 deletedOrders.Dispose();
 if (newOrders != null)
 newOrders.Dispose();
 if (modifiedOrders != null)
 modifiedOrders.Dispose();
}
}

```

o Complete the try-catch block

- Call the Update method to save the changes of Child Deletes

```

if (deletedOrders != null)
 order_DetailsTableAdapter1.Update(deletedOrders);

```

- Call the Update method to save all changes of Parent tables
- Call the Update method to save the changes of Child Inserts and Updates

#### **D) Compiling Your Application**

- To compile and run your application select **Debug | Start Debugging** menu command or press **F5**.