## Lab: Generating and loading the Grades Report

### Lab Setup

Estimated Time: **60 minutes**

### Exercise 1: Serializing Data for the Grades Report as JSON

#### Task 1: Prompt the user for a file name and retrieve the grade data

1. Open **Visual Studio 2017**.
2. In **Visual Studio**, on the **File** menu, point to **Open**, and then click **Project/Solution**.
3. In the **Open Project** dialog box, browse to **E:/Allfiles/Mod06/Labfiles/Starter/Exercise 1**, click **GradesPrototype.sln**, and then click **Open**
4. From the **Build** menu, select **Rebuild Solution**.
5. In **Solution Explorer**, expand **GradesPrototype**, expand **Views**, and then double-click **StudentProfile.xaml**.
6. Note that this view displays and enables users to add grades for a student. The solution has been updated to include a **Save Report** button that users will click to generate and save the grades report.
7. On the **View** menu, click **Task List**.
8. In the **Task List** window, double-click the **TODO: Exercise 1: Task 1a: Add Using for Newtonsoft.Json.** task.
9. In the code editor, click in the blank line below the comment, and then type the following code:

* using Newtonsoft.Json;

1. Double-click the **TODO: Exercise 1: Task 1b: Store the return value from the SaveFileDialog in a nullable Boolean variable.** task.
2. In the code editor, click in the blank line below the comment, and then type the following code:

* bool? result = dialog.ShowDialog();
* if (result.HasValue && result.Value)
* {

1. Click at the end of the last comment in this method, press Enter, and then type the following code:

* }

1. In the **Task List** window, double-click the **TODO: Exercise 1: Task 1c: Get the grades for the currently selected student.** task.
2. In the code editor, click in the blank line below the comment, and then type the following code:

* List<Grade> grades = (from g in DataSource.Grades  
   where g.StudentID == SessionContext.CurrentStudent.StudentID  
   select g).ToList();

#### Task 2: Serialize the grade data to a file stream

1. In the **Task List** window, double-click the **TODO: Exercise 1: Task 2: Serialize the grades to a JSON.** task.
2. In the code editor, click at the end of the comment, press Enter, and then type the following code:

var gradesAsJson = JsonConvert.SerializeObject(grades,

Newtonsoft.Json.Formatting.Indented);

#### Task 3: Save the JSON document to disk by using FileStream

1. In the **Task List** window, double-click the **TODO: Exercise 1: Task 3a: Modify the message box and ask the user whether they wish to save the report** task.
2. In the code editor, delete the line of code below the comment, and then type the following code:

MessageBoxResult reply = MessageBox.Show(gradesAsJson, "Save Report?",

MessageBoxButton.YesNo, MessageBoxImage.Question);

1. In the **Task List** window, double-click the **TODO: Exercise 1: Task 3b: Check if the user what to save the report or not** task.
2. In the code editor, click at the end of the comment, press Enter, and then type the following code:

if (reply == MessageBoxResult.Yes) {

1. Click at the end of the last comment in this method, press Enter, and then type the following code:

}

1. In the **Task List** window, double-click the **TODO: Exercise 1: Task 3c: Save the data to the file by using FileStream** task.
2. In the code editor, click at the end of the comment, press Enter, and then type the following code:

FileStream file = new FileStream(dialog.FileName, FileMode.Create, FileAccess.Write);

StreamWriter streamWriter = new StreamWriter(file);

streamWriter.Write(gradesAsJson);

file.Position = 0;

1. In the **Task List** window, double-click the **TODO: Exercise 1: Task 3d: Release all the stream resources** task.
2. In the code editor, click at the end of the comment, press Enter, and then type the following code:

streamWriter.Close();

streamWriter.Dispose();

* file.Close();
* file.Dispose();

#### Task 4: Run the application and check the save functionality

1. On the **Build** menu, click **Build Solution**.
2. On the **Debug** menu, click **Start Without Debugging**.
3. In the **Username** text box, type **vallee**.
4. In the **Password** text box, type **password99**, and then click **Log on**.
5. In the main application window, click **Kevin Liu**.
6. In the **Report Card** view, click **Save Report**.
7. In the **Save As** dialog box,the desired folder then click **Save**.
8. Review the JSON data displayed in the message box, and then click **Yes**.
9. Close the application.
10. In **Visual Studio**, on the **File** menu, click **Close Solution**.
11. Close **Visual Studio**.
12. Locate the destination folder that you have just saved the file to, and open the **Grades.Json** file.

>**Note:** If JSON file is not opening, right-click on saved file and point to open with and then click **more apps** and select **Microsoft Visual Studio Version Selector** and click **OK**.

1. Locate the **SubjectName** element with the value **Math** inside the JSON array.
2. Change **'Math' Assessment** to be **A** and the **Comments** to be **Very Good**.
3. Save and close the file.

**Result:** After completing this exercise, users will be able to save student reports to the local hard disk in JSON format.

### Exercise 2: Deserialize Data from the JSON Report to Grades Object

#### Task 1: Define the File Dialog settings to load the report file

1. Open **Visual Studio 2017**.
2. In **Visual Studio**, on the **File** menu, point to **Open**, and then click **Project/Solution**.
3. In the **Open Project** dialog box, browse to **E:/Allfiles/Mod06/Labfiles/Starter/Exercise 2**, click **GradesPrototype.sln**, and then click **Open**.
4. Build the solution.
5. On the **View** menu, click **Task List**.
6. In the **Task List** window, double-click the **TODO: 02: Task 1: Define the File Dialog settings to load the report file** task.
7. In the code editor, click at the end of the comment, press Enter, and then type the following code:

OpenFileDialog dialog = new OpenFileDialog();

dialog.Filter = "JSON documents|\*.json";

* // Display the dialog and get a filename from the user
* bool? result = dialog.ShowDialog();

#### Task 2: Load the report and display it to the user

1. In the **Task List** window, double-click the **TODO: 02: Task 2a: Check the user file selection** task.
2. In the code editor, click at the end of the comment, press Enter, and then type the following code:

if (result.HasValue && result.Value) {

1. Click at the end of the last comment in this method, press Enter, and then type the following code:

}

1. In the **Task List** window, double-click the **TODO: 02: Task 2b: Read the report data from Disk** task.
2. In the code editor, click at the end of the comment, press Enter, and then type the following code:

string gradesAsJson = File.ReadAllText(dialog.FileName);

1. In the **Task List** window, double-click the **TODO: 02: Task 2c: Deserialize the JSON data to grades list** task.
2. In the code editor, click at the end of the comment, press Enter, and then type the following code:

var gradeList = JsonConvert.DeserializeObject<List<Grade>>(gradesAsJson);

1. In the **Task List** window, double-click the **TODO: 02: Task 2d: Display the saved report to the user** task.
2. In the code editor, click at the end of the comment, press Enter, and then type the following code:

studentGrades.ItemsSource = gradeList;

#### Task 3: Run the application and check the load functionality

1. On the **Build** menu, click **Build Solution**.
2. On the **Debug** menu, click **Start Without Debugging**.
3. In the **Username** text box, type **vallee**.
4. In the **Password** text box, type **password99**, and then click **Log on**.
5. In the main application window, click **Kevin Liu**.
6. In the **Report Card** view, click **Load Report**.
7. In the **Open** dialog box, go to your destination folder where the report was saved in the previous task and locate the saved report.
8. Select the report file and then click **Open**.
9. Review the data displayed in the **Report Card** view and verify that the changes that were made in the report file are reflecting now.
10. Close the application.
11. In **Visual Studio**, on the **File** menu, click **Close Solution**.
12. Close **Visual Studio**.

**Result:** After completing this exercise, users will be able to load student reports from the local hard disk.

©2018 Microsoft Corporation. All rights reserved.