Exploring the QuestionsAndAnswers Form

1. What happens before the form loads

When the QATreeForm’s editQAFileButton is clicked, the program gets the qaFileName from the Text property of the selected node and transmits it to the AnswerQuestionsDataModel’s setQAFileNameStr method

string qaFileName = subjectTreeView.SelectedNode.Text;

AnswerQuestionsDataModel.setQAFileNameStr(qaFileName);

this.Hide();

QuestionAndAnswerForm questionAndAnswerForm = new QuestionAndAnswerForm();

questionAndAnswerForm.ShowDialog();

1. What happens when the form loads

**QuestionAndAnswerForm.cs**

/// <summary>

/// This Method: 1) Opens a local copy of the qaDictionary

/// 1. If it is called as a result of using the QA Tree form's Create/Edit button

/// then the qaFileNameString is the file name.

/// a. If the retrieved qaFilePath is blank and new dictionary is called else

/// b. The dictionary associated with an extant qaFile is loaded

/// 2. If it is called by Openeing the form from the dashboard then the qaFileNameString

/// is blank and the method is skipped

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void QuestionAndAnswerForm\_Load(object sender, EventArgs e)

{

string qaFileNameString = AnswerQuestionsDataModel.getQAFileNameStr();

if (qaFileNameString != "")

{

string qaFilePath = AnswerQuestionsDataModel.getQAFilePath();

if (qaFilePath == "")

{

// Get a blank dictionary

qaDictionary = AnswerQuestionsDataModel.QandADictionary;

questionNumberValue.Text = qaDictionary.Count.ToString();

questionValue.Select();

return;

}

AnswerQuestionsDataModel.loadQAFileIntoDictionary(AnswerQuestionsDataModel.getQAFilePath());

qaDictionary = AnswerQuestionsDataModel.QandADictionary;

selectEditTypeLable.Visible = true;

}//End if (qaFileNameString != "")

}// End QuestionAndAnswerForm\_Load

- - - - - -

AnswerQuestionsDataModel.

/// <summary>

/// 1. Read all the lines in the designated file into qaLineArray

/// 2. Convert the qaLineArray into qaDictionary

/// 3. Convert the qaDictionary into QandADictionary

/// 4. Using the count of the dictionary create a '^' delimited string

/// of sequential numStrings representing all the Keys in QandADictionary

/// </summary>

/// <param name="qaFilePath"></param>

public static void loadQAFileIntoDictionary(string qaFilePath)

{

// 20200401 Determine if the file is empty

if (new FileInfo(qaFilePath).Length > 0)

{

//File is not empty

// Read all lines from the file into qaLineArray

string[] qaLineArray = File.ReadAllLines(qaFilePath);

// Parse each line in the qaLineArray creating a new entry in qaDictionary

foreach (string qaLine in qaLineArray)

{

string qaNumStr = StringHelperClass.returnNthItemInDelimitedString(qaLine, '^', 0);

int qaNumInt = Int32.Parse(qaNumStr);

string newQALine = qaLine;

newQALine = StringHelperClass.removeNthItemFromDelimitedString(qaLine, '^', 0);

// qaDictionary key is the integer value of the qaLine and the value is the remainder of the qaLine

qaDictionary.Add(qaNumInt, newQALine);

}

// Set QandADictionary to qaDictionary

QandADictionary = qaDictionary;

// Create seriatimQuestionNumberDelStr, a delimited string of seriatim question numbers,

//seriatimQuestionNumberDelStr

seriatimQuestionNumberDelStr = "";

for (int i = 0; i < QandADictionary.Count; i++)

{

string q = i.ToString();

seriatimQuestionNumberDelStr = seriatimQuestionNumberDelStr + q + '^';

}

// Remove last '^'

seriatimQuestionNumberDelStr = seriatimQuestionNumberDelStr.Substring(0,

seriatimQuestionNumberDelStr.Length - 1);

}

else

{

//File is Empty

// Create an empty QandADictionary

QandADictionary = qaDictionary;

// Create seriatimQuestionNumberDelStr

seriatimQuestionNumberDelStr = "";

}

// Create qaList from seriatimQuestionNumberDelStr

qaList = seriatimQuestionNumberDelStr;

}// End loadQAFileIntoDictionary