

Forum:[Hack Forums](#)

Thread: [Exporting and Importing CSV](#)

Guillyian - July 7, 2022, 12:33 am

My dumba\$\$ did something wrong with the importing of CSV files code and I can't figure out what. Any help would be much appreciated!

```
Code private async void savCoord_Click(object sender, EventArgs e)
{

    SaveFileDialog saveFileDialog = new SaveFileDialog();
    saveFileDialog.InitialDirectory = "c:\\";
    saveFileDialog.Filter = "CSV|*.csv";
    saveFileDialog.FilterIndex = 1;

    if (saveFileDialog.ShowDialog() == DialogResult.OK)
    {
        using (StreamWriter sw = new StreamWriter(new FileStream(saveFileDialog.FileName,
        FileMode.Create), Encoding.UTF8))
        {
            StringBuilder sb = new StringBuilder();
            foreach (ListViewItem item in PositionsListView.Items)
            {
                sb.AppendLine(string.Format("{0},{1},{2},{3}", item.SubItems[0].Text,
item.SubItems[1].Text, item.SubItems[2].Text, item.SubItems[3].Text));
            }
            await sw.WriteLineAsync(sb.ToString());
            MessageBox.Show("Success!", "Notice", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }
    }
}

private void openCoord_Click(object sender, EventArgs e)
{

    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.InitialDirectory = "c:\\";
    openFileDialog.Filter = "CSV|*.csv";
    openFileDialog.FilterIndex = 1;

    try
    {
        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            FileStream srcFS;
```

```

srcFS = new FileStream(openFileDialog.FileName, FileMode.Open);
StreamReader srcSR = new StreamReader(srcFS, System.Text.Encoding.Default);
do
{
    string ins = srcSR.ReadLine();
    string[] columns = ins.Split(',');

    ListViewItem lvi = new ListViewItem(columns[0]);

    lvi.SubItems.Add(ins);

    for (int i = 1; i < columns.Count(); i++)
    {
        lvi.SubItems.Add(columns[i]);
    }

    PositionsListView.Items.Add(lvi);

} while (true);
srcSR.Close();
}
}
catch (Exception errorMsg)
{
    MessageBox.Show(errorMsg.Message, "Error reading a file", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}
}

```

MavenGradle - July 7, 2022, 7:13 am

Importing? This seems to be more like exporting.

In addition, if your data source has values that contain comma's, your code will fail. You will need to add quotes to values that contain the same value as the separator in CSV files, which are not limited to comma's either.

Code `saveFileDialog.FilterIndex = 1;` This is not necessary. The default value is already set to 1:

<https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.filedialog.filterindex?view=windowsdesktop-6.0>

Code `saveFileDialog.InitialDirectory = "c:\\";` This is dangerous, not everyone uses a C drive.

Instead you could use something like:

Code `Path.GetPathRoot(Environment.SystemDirectory);`

On to the next problem / question mark:

Why use a `StringBuilder` when you can directly write a Line to the `StreamWriter`? IO vs Memory?

Your `FileStream` is also dangerously misplaced and might not get properly closed.

Something like this might be more appropriate for your needs:

Code using `(FileStream fileStream = new FileStream("example.csv", FileMode.Create))`

using `(StreamWriter streamWriter = new StreamWriter(fileStream))`

```

{
    foreach (ListViewItem item in PositionsListView.Items)

```

```

    {
        await sw.WriteLineAsync(string.Format("{0},{1},{2},{3}", item.SubItems[0].Text,
item.SubItems[1].Text, item.SubItems[2].Text, item.SubItems[3].Text));
    }
}

```

Do note, I leave the problem mentioned above regarding the need for quotation marks that need to surround values that contain the separator to you. It's a simple if statement to change the formatting, to be honest.

Guillyian - July 7, 2022, 5:40 pm

(Jul 7, 2022 07:13 AM)MavenGradle Wrote: [\[CITE\]](#) Importing? This seems to be more like exporting. In addition, if your data source has values that contain comma's, your code will fail. You will need to add quotes to values that contain the same value as the separator in CSV files, which are not limited to comma's either.

Code saveFileDialog.FilterIndex = 1;This is not necessary. The default value is already set to 1:

<https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.filedialog.filterindex?view=windowsdesktop-6.0>

Code saveFileDialog.InitialDirectory = "c:\\";This is dangerous, not everyone uses a C drive. Instead you could use something like:

Code Path.GetPathRoot(Environment.SystemDirectory);

On to the next problem / question mark:

Why use a StringBuilder when you can directly write a Line to the StreamWriter? IO vs Memory?

Your FileStream is also dangerously misplaced and might not get properly closed.

Something like this might be more appropriate for your needs:

```

Code using (FileStream fileStream = new FileStream("example.csv", FileMode.Create))
using (StreamWriter streamWriter = new StreamWriter(fileStream))
{
    foreach (ListViewItem item in PositionsListView.Items)
    {
        await sw.WriteLineAsync(string.Format("{0},{1},{2},{3}", item.SubItems[0].Text,
item.SubItems[1].Text, item.SubItems[2].Text, item.SubItems[3].Text));
    }
}

```

Do note, I leave the problem mentioned above regarding the need for quotation marks that need to surround values that contain the separator to you. It's a simple if statement to change the formatting, to be honest.

this helps with half my problem, I still need a way to open the csv file and write the contents of it into the listview component.

MavenGradle - July 7, 2022, 6:25 pm

(Jul 7, 2022 05:40 PM)Guillyian Wrote: [\[CITE\]](#) this helps with half my problem, I still need a way to open the csv file and write the contents of it into the listview component.

The process of reading is extremely similar to writing.

1. You read using a [StreamReader](#)
2. You split values using the required separator
3. You parse the data
4. You add the data to the ListView

Splitting strings:

<https://docs.microsoft.com/en-us/dotnet/csharp/how-to/parse-strings-using-split>

To create / add items to a ListView control:

```
Code string[] row = { "ABC", "DEF", "XYZ" };  
ListViewItem listViewItem = new ListViewItem(row);  
IsvCollection.Items.Add(listViewItem);
```

Do note:

Values can be placed between quotation marks and may contain the separator themselves, and as such, your CSV parser will need to be aware of that. It's essentially the exact same thing you need to do when writing.

Guillyian - July 7, 2022, 7:04 pm

(Jul 7, 2022 06:25 PM)MavenGradle Wrote: [\[CITE\]](#) The process of reading is extremely similar to writing.

1. You read using a [StreamReader](#)
2. You split values using the required separator
3. You parse the data
4. You add the data to the ListView

Splitting strings:

<https://docs.microsoft.com/en-us/dotnet/csharp/how-to/parse-strings-using-split>

To create / add items to a ListView control:

```
Code string[] row = { "ABC", "DEF", "XYZ" };  
ListViewItem listViewItem = new ListViewItem(row);  
IsvCollection.Items.Add(listViewItem);
```

Do note:

Values can be placed between quotation marks and may contain the separator themselves, and as such, your CSV parser will need to be aware of that. It's essentially the exact same thing you need to do when writing.

This is what I have and it isn't working, I'm clearly having a brain shit right now

```
Code private void openCoord_Click(object sender, EventArgs e)  
{
```

```
    OpenFileDialog openFileDialog = new OpenFileDialog();  
    openFileDialog.InitialDirectory = Path.GetPathRoot(Environment.SystemDirectory);  
    openFileDialog.Filter = "CSV|*.csv";
```

```
    try  
    {
```

```
        if (openFileDialog.ShowDialog() == DialogResult.OK)  
        {
```

```
            FileStream srcFS;  
            srcFS = new FileStream(openFileDialog.FileName, FileMode.Open);  
            StreamReader srcSR = new StreamReader(srcFS, System.Text.Encoding.Default);  
            foreach (ListViewItem item in PositionsListView.Items)  
            {  
                string ins = srcSR.ReadLine();  
                string[] row = { "X", "Y", "L/R", };  
                ListViewItem listViewItem = new ListViewItem(row);
```

```

        PositionsListView.Items.Add(listViewItem);
    }
}
}
catch (Exception errorMsg)
{
    MessageBox.Show(errorMsg.Message, "Error reading a file", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}
}

```

MavenGradle - July 11, 2022, 5:19 am

(Jul 7, 2022 07:04 PM)Guillyian Wrote: [\[CITE\]](#) This is what I have and it isn't working, I'm clearly having a brain shit right now

...

You need to close your streams to begin with and you also need to revisit your logic. Your code isn't making much sense; you need to loop over the lines in your stream, split the lines with the delimiter and add the values to your ListView control.

You'll need something like this:

```

Code using (StreamReader sr = new StreamReader(xyz))
{
    while (!reader.EndOfStream)
    {
        string line = reader.ReadLine();
        string[] values = line.Split(','); // Split line using the delimiter and make sure to account for
values between quotation marks. This example doesn't do that, I'll leave that up to you
        // Create and add the new ListViewItem to your ListView Control with the values from the
array above
    }
}

```

So, something I also forgot to mention when dealing with CSV files besides from values that could be encapsulated with quotation marks, are comment fields, which start with a hashtag (#). When splitting values, you will need to account for this if you want to code a proper CSV parser.