

# Developer Documentation

---

## Table of Contents

---

1. [Object Model](#)
2. [CsvHelper](#)
3. [Business Logic](#)
4. [Windows Forms](#)

## Object Model

---

The object model can be found in the same directory as this file with the filename *ObjectModel.png*

## CsvHelper

---

CsvHelper is an external NuGet package. Read the Documentation at <http://joshclose.github.io/CsvHelper/>

## BL

---

This can be found in the BL Project

## Item

---

Item Class

### Code

The Item's Code

### CurrentCount

The Current Count of the amount of this Item in stock

### Description

A Description of the Item

### OnOrder

Whether the Item is On Order or not

## ItemRepository

---

This class contains the methods to read and write items too a csv file

## Constructor

ItemRepository Constructor

## ItemMap

---

This class tells CsvHelper how to convert the file rows into items and vice versa

## MyBooleanConverter

---

This class just converts Yes/No values into True/False

## BL.ItemRepository.Retrieve(stream)

Retrieves a list of items from the supplied stream

Name	Description
stream	<i>System.IO.Stream</i> The stream to read to

### Returns

A IEnumerable of the items in the stream

*System.ArgumentNullException*: Thrown when the stream is null

*System.ArgumentException*: Thrown when the stream has incomplete items

## BL.ItemRepository.Save(stream)

Saves a list of items from the supplied stream

Name	Description
stream	<i>System.IO.Stream</i> The stream to save to

*System.ArgumentNullException*: Thrown when the stream is null

## BL.ItemRepository.Update(itemCode, currentCount)

Updates an Item's Current Count.

Name	Description
itemCode	<i>System.String</i> The Item's code used to find the item
currentCount	<i>System.Int32</i> The Item's new CurrentCount

*System.ArgumentNullException*: Thrown when itemCode is null

*System.ArgumentException*: Thrown when itemCode is not in the itemRepository items dictionary

## WF

---

This can be found in the WF Project

### WF.ItemView.CreateColumn(name, columnType, type, readOnly)

Creates a Column

Name	Description
name	<i>System.String</i> Name of the Column
columnType	<i>WF.ColumnType</i> The column type of the column, used to set the position
type	<i>System.Type</i> The type of the column
readOnly	<i>System.Boolean</i> Is the column editable

### WF.ItemView.CreateColumns

Add the Columns to the DataGridView

### WF.ItemView.ItemDataGridView\_CellEndEdit(System.Object,System.Windows.Forms.DataGridViewCellEventArgs)

This function is run when the user ends editing a field. It then updates the item edited.

### WF.ItemView.ItemDataGridView\_DataError(System.Object,System.Windows.Forms.DataGridViewDataErrorEventArgs)

Custom Errors

### WF.ItemView.OpenFile

Open a Open File Dialog and then display the Contents

### WF.ItemView.OpenToolStripMenuItem\_Click(System.Object,System.EventArgs)

Open a file when import menu item is clicked.

### WF.ItemView.PopulateRow(item)

Adds a row to the DataGridView

Name	Description
item	<i>BL.Item</i> The Item to populate the DataGridView with

## WF.ItemView.PopulateRows(items)

Populates the rows of the DataGridView

Name	Description
items	<i>System.Collections.Generic.IEnumerable{BL.Item}</i> An IEnumerable of the items to populate the DataGridView with

## WF.ItemView.SaveFile

Open a Save File dialog and then save the file

## WF.ItemView.SaveToolStripMenuItem\_Click(System.Object,System.EventArgs)

Save the file when export menu item is clicked.