**Documentation**

**WPF/WinForms POS System**

Compuflex

30 Hillside Avenue, Springfield, New Jersey 07081

Phone: 973.379.4567 / fax: 973.379.5928

Email: [cflexinfo@compuflexcorp.com](mailto:cflexinfo@compuflexcorp.com)

Contents

[About this Document 3](#_Toc490481418)

[What is WPF/WinForms POS System? 3](#_Toc490481419)

[How Does it Work? 3](#_Toc490481420)

[Glossary of Key Terms 4](#_Toc490481421)

[Action Buttons 4](#_Toc490481422)

[Editing Product Data Sets 8](#_Toc490481423)

## About this Document

This is document is intended as a source of reference for users of **WPF POS System** and **WinForms POS System**. This reference guide consists of all the features and functions present in the application. The purpose of this document is to support the **POS System Users**.

## What is WPF/WinForms POS System?

The POS System is a mock POS System application emulating functionality of existing commercial POS Systems available. This application was developed with the intent to be used for testing/developing of a larger screen engine management software.

This software aims to provide real world features one might encounter when working with commercially available POS Systems.

## How Does it Work?

This application provides a simple and user friendly interface to simulate "Employee users", adding items to a cart, and finally processing payment while saving transaction history.

**Settings:**

Change settings by right clicking and clicking settings. Users have the option to change how the employee’s name appears in the application. Users have the ability to display the name as a label, image, button, or in a dropdown menu. Additionally users have the ability to select if payment will be processed in-window or in a separate pop-up window.

**How to use:**

Users can click on an employee’s name to change it. Then click on the “Add” button to add items to the cart. Alternatively click on “Clear” to clear the contents of the cart.

A user can only process payment if there are items in the cart. Depending on the settings, payment will be processed in-window or in a separate pop-up window. After the payment is successfully processed, when the correct total or higher is entered, the transaction will be saved in the Trans tab. The user is now free to make another transaction.

## Glossary of Key Terms

| **Term** | **Definition** |
| --- | --- |
| **WPF** | Windows Presentation Foundation. a graphical subsystem by Microsoft for rendering user interfaces in Windows-based applications. |
| **WinForms** | Windows Forms (WinForms) is a graphical (GUI) class library included as a part of Microsoft .NET Framework,[1] providing a platform to write rich client applications for desktop, laptop, and tablet PCs. |
| **POS System** | Point of Sales System, (in this case only software) allows a merchant to calculate the amount owed by a customer, handle payment, and save transaction history. |
| **Trans** | Shorthand for Transaction, Trans tab displays the transaction history. |

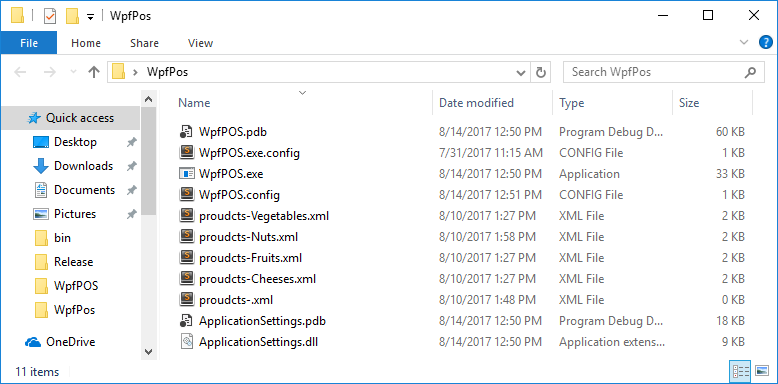
## Action Buttons

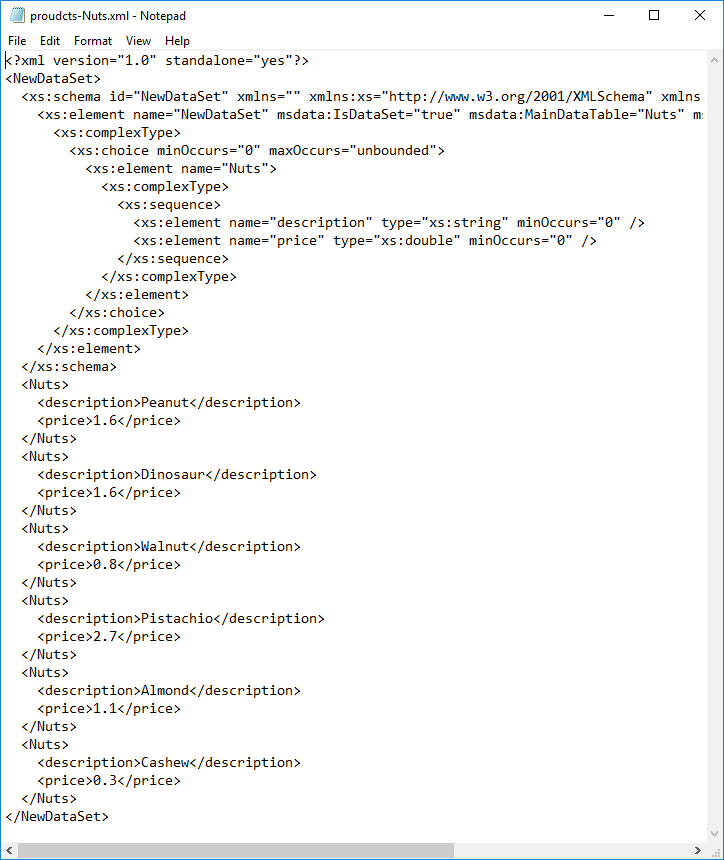
| **Icon** | **Action** |
| --- | --- |
| [Name Label](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/label.PNG) | Label that displays an "employee's" name. |
| [Time Label](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/timeStamp.PNG) | Displays the current time, used for transaction history. |
| [Add Button](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/addBtn.PNG) | Adds a random nut product to the cart (quantity is also random). |
| [Clear Button](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/clearBtn.PNG) | Clears the cart of all items. |
| [Total Label](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/totalLabel.PNG) | Displays the updated total for the current transaction. |
| [Pay Button](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/payBtn.PNG) | When clicked it gives the user the option to pay in-window or through a popup window. (Option for in-window/popup window located in settings) |
| [TenderInWindow](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/tenderInWindow.PNG) | Visible once the [Pay Button](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/payBtn.PNG) is clicked, a text box where a user can enter payment in the format x.xx . |
| [Process](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/processBtn.PNG) | Once the user enters enough money, the user may process the transaction. Each transaction is saved in the 'Trans' tab. **Note:** Once the application is closed, all transaction history is lost. |
| [tabs](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/tabs.PNG) | Tabs, the first shows the items currently in the cart. The second tab 'Trans' displays transaction history. |
| [Settings](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/settings.PNG) | When a user right-clicks, a settings window will appear. |
| [payWindow](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/payWindow.PNG) | Based on what option is selected in the settings menu, the pay window may appear for a user to process payment. |
| [Settings Window](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/settingsWindow.PNG) | Settings window, a user can choose how the "employee's" name is displayed. The name can be displayed as a label, image, in a drop down container, or as a button. (Default is set to labels). |
| [Empty Cart](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/emptyCartScreen.PNG) | An example of what an empty cart screen looks like. |
| [Empty Transaction](https://github.com/carlos42044/Wpf_POS_system/blob/master/img/emptyTransScreen.PNG) | An example of what an empty transaction screen looks like. |

## Editing Product Data Sets

Both the WPF and WinForms application have 4 product data sets: Nuts, Cheeses, Fruits, and Vegetables. A user can add or edit products in each data set.

Steps to edit a product:

1. Navigate to the directory where your application is running from (i.e. C:\Users\yourName\Desktop\WpfPos) you should see a window like this:   
   
2. Next you choose which product set you would like to edit. For this example we will choose the Nuts data set (products-Nuts.xml). Open up the product-Nuts.xml file in a text editor (Notepad will work just fine). Your file should look like this:



1. Now to **add a new nut product**, we will be adding a new Hazelnut with a price of 40 cents. Your entry will be in the following form:

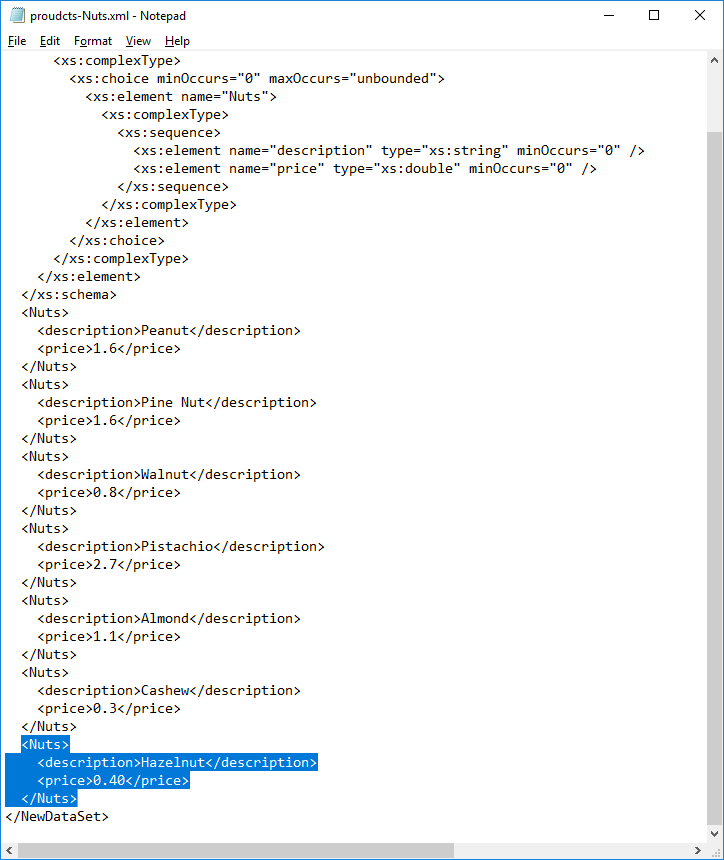
<Nuts>

<description>Hazelnut</description>

<price>0.40 </price>

</Nuts>

You will be adding your new nut product at the end of the document after the last </Nuts> tag and before the </NewDataSet> tag. Your products-Nuts.xml should now look like this:



(blue highlighted portion is the new nut product we just added)

1. In a similar fashion you can edit existing product names and prices by finding a product between <Nuts> … </Nuts> tag and editing the <description>…</description> field to change the name of a product. To change the price, simply edit the <price>…</price> field.
2. Save the document and your changes will have been made!