Scripting

Megan De Freitas - 2024-11-22 - Setup

Overview

Scripting gives users the ability to add customized behavior at certain times during data processing, allowing great flexibility in implementing rules and actions custom tailored for a specific business. The scripts are written in the C# programming language and allow a great deal of added functionality. For example a script can be written so that each time a document is about to be saved, the script will examine the line items on the document and cancel the save if any of the line items have been discontinued and preventing the selling of items that are no longer available.

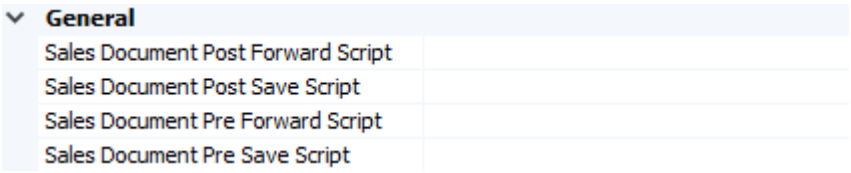
Settings

Scripts are implemented through settings.

Sales Document Post Forward Script – Script that is run after forwarding a Sales Document.

Sales Document Pre Forward Script – Script that is run before forwarding a Sales Document.

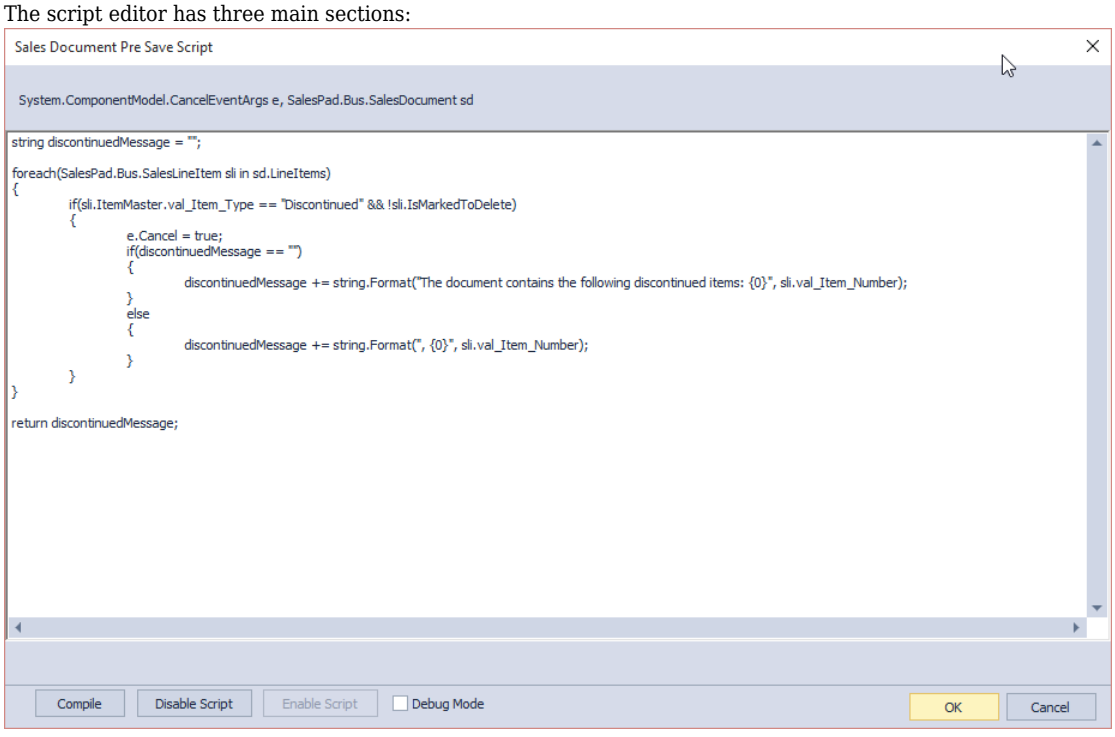
Sales Document Pre Save Script – Script that is run before saving a Sales Document.



Usage

To create or edit scripts, click the ellipsis (**...**) on the right side of the input field on the desired setting to open the script editor.

The script editor has three main sections:



1. Above the text edit area are listed the parameters available for the script. The parameters may include a CancelEventArgs, labeled as e, which by setting to true can be used to tell the process the script is running with to stop. Parameters may also include business objects related to the ongoing process.
2. In the middle is the text edit area, this is where the actual script will be written using C#.
3. Along the bottom are several buttons:
    - *Compile* – When the script is complete, this can be used to test that the script can compile. Any compilation errors found, such as errors with language semantics or missing parentheses, will be displayed. If no errors are found, a message will appear above the button stating that the script was compiled successfully.
    - *Disable Script* – If you do not want to delete the script, you can keep it but turn it off so that it is no longer used.
    - *Enable Script* – Turn a disabled script back on

Once a script is finished and can successfully compile, click **OK** and the script will be saved to the setting. Next time the operation is run the script will be loaded and run as part of the operation.

Important Reminders

- Please install this on a test machine and run it against a test database before using it on your live system.
- You should always make sure you have a database backup prior to installing new scripts.
- C# scripts, unless written by SalesPad, are the responsibility of the dealer/customer.
- Anytime a change to a script is made, the Mobile Server needs to be restarted for the changes to take effect.

Pre Save Script Example

```
string msg = "";

foreach (SalesPad.Bus.SalesLineItem sli in sd.LineItems)
{
	//Prevents a markdown discount above 5% or a discount that would bring price below cost
	if(columnName == "colMarkdown_Pct")
	{
		decimal markdown = (sli.val_Unit_Price * sli.val_Quantity) - ((sli.val_Unit_Price * sli.val_Quantity) * (sli.val_Markdown_Pct / 100));

		if(markdown < sli.val_Extended_Cost)
		{
			msg += sli.val_Item_Number.ToString() + ": Markdown brings price below cost. Lower markdown percent.";
			sli.val_Markdown_Pct = 0;
			e.Cancel = true;
		}
		else if(sli.val_Markdown_Pct > 5)
		{
			msg += sli.val_Item_Number.ToString() + ": Markdown percentage is too high. Lower markdown percent.";
			sli.val_Markdown_Pct = 0;
			e.Cancel = true;
		}
	}

	//Prevents a markup greater than 10% of original price
	if(columnName == "colUnit_Price")
	{
		SalesPad.Bus.ItemPriceList plist = new SalesPad.Bus.ItemPriceList(sli.val_Item_Number, sd.val_Currency_ID, sli.val_Price_Level, sli.val_Unit_Of_Measure, null);

		decimal prevPrice = sli.ItemMaster.val_List_Price * (plist.val_UOfM_Price /100m);
		decimal maxPrice = prevPrice * 1.10m;
		decimal minPrice = prevPrice - (prevPrice * .05m);

		if(maxPrice < sli.val_Unit_Price)
		{
			msg += sli.val_Item_Number.ToString() + ": New price is too high. Max price available is " + maxPrice.ToString("#.00#");
			sli.val_Unit_Price = prevPrice;
			e.Cancel = true;
		}
		else if(minPrice > sli.val_Unit_Price)
		{
			msg += sli.val_Item_Number.ToString() + ": New price is too low. Min price available is " + minPrice.ToString("#.00#");
			sli.val_Unit_Price = prevPrice;
			e.Cancel = true;
		}
	}
}

return msg;
```

Script runs before saving a document that will cancel the save if any of the line items on the document are being sold too high or too low of the items cost.