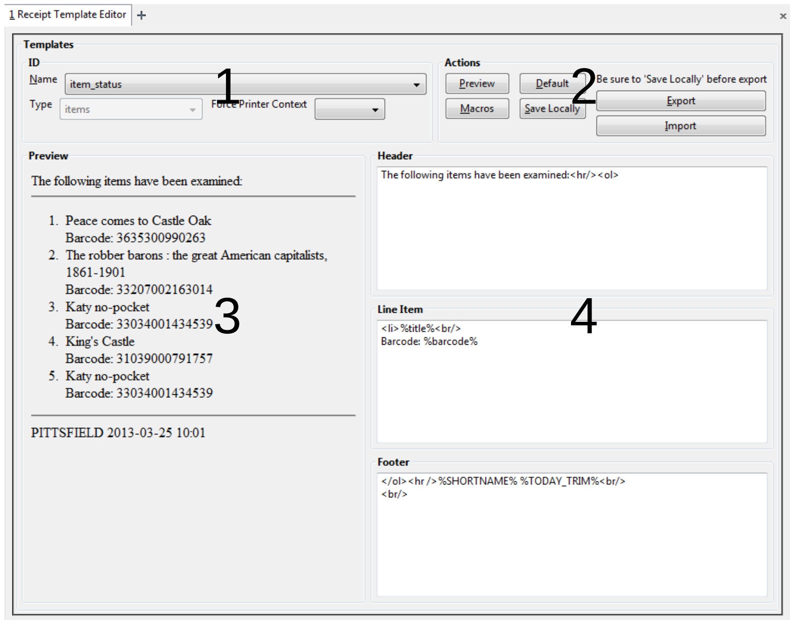
Receipt Templates in Evergreen

The Receipt Template Editor Itself 2
Adding Plain Text 3
HTML Tags 3
Using CSS For Styling 8
Template Toolkit Variables 10
FAQs and a few cool functions 11
Custom note about book sale 11
Remove staff name from the bottom of a receipt: 11
Increasing the length of a receipt: 12
Adding a picture to your receipt 12
Moving an object up/down/left/right using the CSS position property 13
Removing the time on a date stamp using the %SUBSTR% function 13
Using a barcode font to display real barcodes on receipts 15
Transferring Receipt Templates between Workstations 16
Printer Settings Editor 17
Layout 17
Printer Context 17
Normal Settings 18
Set Default Printer and Print Test Page 18
Page Settings 18
Print Strategies 19
Backup up receipts and printer settings manually 20
Which receipt does what 21

The Receipt Template Editor Itself

The receipt function created for the Evergreen ILS allows staff to customize each receipt to the local tastes and needs. These customizations can be done using a combination of plain text, HTML/CSS and Template Toolkit variables and we will discuss each of these items topics in turn. However, let's first take a look at the receipt template editor itself so that we know its parts. To access the receipt template editor, click on Receipt Template Editor button on the first screen that is seen when you login to Evergreen. Alternately, go to Admin(-) in the upper right corner, then down to Workstation Administration, and then to Receipt Template Editor. Here are the parts of the Editor screen with an explanation. The items in the numbered list correspond to the sections in the image below.



1. This first section in the upper left corner lists the receipt currently being edited. It defaults to 'item_status' when opened but is a drop box and can be changed to any of the other receipts in the system. To change to a different receipt, click on the drop box that has the current receipt name and a list will appear with all of the receipts available in Evergreen. Select a receipt you wish to customize and the bottom half (sections 3 and 4) will change to that receipt.

2. Section 2 is an admin section with a list of buttons. Here is each button and its function:

- **Preview** – A sort of dummy button but can be used to view changes made in receipt templates.
- **Default** – Resets all of the receipts to their default when the staff client is first installed. Be careful of this button and make sure that you are backing up your receipts as this button can ruin hours of work almost instantaneously. If you accidentally press this button and do not have a backup of your receipts feel free to call us as there may be something that we can do to save your changes.
- **Macros** – displays a new window with a list of Template Toolkit variables that can be added to a receipt. We will explain more about these below but the variables are the things that look like this: `%barcode%` (percent signs around a readable variable name). These variables change depending on the context and can add item, patron, or library data to receipts to make them robust and useful. For example, the item_status receipt allows you to add the item's title, author, call number, etc. by adding the following variables respectively: `%title%`, `%author%`, `%call_number%`.
- **Save Locally** – will save your work when you make a change to a receipt template. Save often and save before moving on to another receipt as moving from one receipt to another will destroy the changes made in the previous if they were not saved.

- **Export** – will bring up a standard save file window and will export all of your receipts at the same time into a text file. This serves as a backup and a way for you to transfer receipts from one workstation to another.
- **Import** – will bring up a standard open file window and, when you find an exported receipt template file, will import a batch of receipts. As with export, the importer will import all receipts at the same time.

3. The preview section allows you to see what the receipt will look like when printed out. When a change is made in section 4 (which is where changes to receipts are actually made), the preview will update instantly after you click out of the box in which you are working. The preview is close to what will actually be printed out but not a perfect representation.

4. This is where text/code is added to actually customize a receipt. Plain text, HTML, CSS, and Template Toolkit Variables are added to each of the boxes and the preview updates to show what the changes will do. There are three boxes (Header, Line Item and Footer) and they are best explained by example. When a checkout receipt is printed, Evergreen will print what is in the Header box first and only once, then it will go through Line Item box once for each item that was checked out and then finally through the Footer box once.

Adding Plain Text

Adding plain text to a receipt can be as simple as typing into one of the boxes in section 4. For example, if you want to add a 'thank you' message to the bottom of your receipt, decide where you would like it to go and start typing. Let's look at an example: start by deleting all of the text in the header portion of the checkout receipt. Type in the following and then click out of the header box to update the preview:

Code:

```
Welcome to the Free Public Library!
```

Result:

Welcome to the Free Public Library!

That is really it for adding plain text because it is truly that simple. Experiment a little with plain text so that you can become familiar with it because it is truly very useful.

However, the real power of receipt templates is adding styling to this plain text with HTML and CSS. As we go through the rest of this document we will be referencing and modifying the above 'welcome to' line so that it will look a little nicer.

HTML Tags

HTML stands for Hyper Text Markup Language. Markup languages are used to structure (and style to a certain extent) plain text by surrounding the text with what are called 'tags'. Here is an example of an HTML tag:

```
<h1>
```

This tag consists of a two elements. First, there is the tag name which is in this case ‘h1’ and second is the set of brackets ‘<‘ ‘>‘ which surround the name. A tag can either surround text with a beginning and an end tag or exist on their own. The tags that surround text look like this:

```
<h1>Welcome to the Free Public Library!</h1>
```

Tags that surround text have an opening tag without a slash and a closing tag with a slash. Distinguishing between opening and closing tags is important so that multiple layers of tags are possible. A tag that exists on its own will have a slash at the end of the tag like this:

```
<br/>
```

With the concept of surrounding text with tags and using single tags understood, the rest of the HTML tutorial is to learn what the individual tags mean by example.

Here are a few examples of HTML tags: Take a look in section 4 of the receipt template editor. Much of the text is quite readable such as the ‘Welcome to…’ at the top of the screen and then there are many HTML tags that you see in all three sections of the receipt. Here are the most common tags you will see in receipt templates and what they do. Feel free to try each of these examples in the receipt editor:

 or

These are **br**eak tags and they are single, stand-alone tags. They will cause a line break to be created in the text so that the text will start on a new line. If you have a line in one of the boxes in section 4 such as ‘Welcome to your library!’ and put a
 tag between the ‘to’ and ‘your’ the following will occur:

Code:

```
Welcome to your library!
```

Result:

Welcome to your library!

— or this with a break tag —

```
Welcome to <br/>your library!
```

Welcome to
your library!

Break tags are especially necessary for receipt templates because placing text on separate lines using the Enter or Return key mean nothing. Here is an example, try adding line breaks without a
 tag in the receipt template editor:

Code:

```
Text
        broken
     up by
carriage returns
```

Result:

Text broken up by carriage returns

<hr/> *or* <hr>

These are horizontal rule tags and they are single, stand-alone tags. These will create a line across the receipt wherever you ask it. It will also create a line break before and after the tag. Enter one into the line item section and see what happens.

<h1>, <h2>, <h3>, <h4>, <h5>, <h6>

These are heading tags and the larger the number the smaller the text will be rendered. These are tags that surround text so they will have both a beginning and an end tag. The tags go from one to six in number.

Code:

```
<h1>Welcome to New Public Library!</h1>
<h2>Welcome to New Public Library!</h2>
<h3>Welcome to New Public Library!</h3>
<h4>Welcome to New Public Library!</h4>
<h5>Welcome to New Public Library!</h5>
<h6>Welcome to New Public Library!</h6>
```

Result:

Welcome to New Public Library!

Welcome to New Public Library!

Welcome to New Public Library!

Welcome to New Public Library!

Welcome to New Public Library!

Welcome to New Public Library!

 and

This is another tag that requires a beginning and an end tag. These are the **o**rdered **l**ist tags and they create a numbered list of items that, if combined with tags (**l**ist **i**tem), will create this:

Code:

```
<ol>
        <li>Milk</li>
        <li>Bread</li>
        <li>Cheese</li>
</ol>
```

Result:

1. Milk
2. Bread
3. Cheese

In this case the tags need to be inside the <ol> couplet to work. Then the text of the item in the list must go between the <li> tags.

<table> *and* </table>

These tags help to create a table with two other sets of tags <tr></tr> and <td><td> which stand for 'table row' and 'table data'. The <table> tag is a little more complicated of a tag but here is an example of how to create a table:

Code:

```
<table>
<tr> <td>January</td> <td>2,986</td> </tr>
<tr> <td>February</td> <td>3,259</td> </tr>
<tr> <td>March</td ><td>3,325</td> </tr>
</table>
```

Result:

January	2,986
February	3,259
March	3,325

The table tag starts and ends the whole unit. Each row will start and end with a <tr> tag. Inside the <tr> tag, a data cell will start and end with a <td> tag.

You can include an unlimited number of rows and data cells but it is important to keep the same number of data cells in each row. It may seem a little more complicated but tables are very useful for creating web pages but also when customizing Evergreen receipts. As with each of these examples, try it out for yourself.

 and

These tags are used to surround text but they actually do nothing without extra styling using CSS which we will talk about below.

Continuing with our 'free public library' example from the plain text section, let's try to make the plain text more interesting with HTML tags. Let's add a horizontal rule or two and surround the text with header tags. Here is what I came up with and the result:

Code:

```
<hr/>
<h2>Welcome to the<br/>Free Public Library</h2>
<hr/>
```

Result:

Welcome to the
Free Public Library

These tags above are enough to do virtually everything that is needed in receipt templates. There are many tutorials out there that will cover HTML in greater detail but Google is your best friend if you have any HTML questions. Search like: ‘HTML create unordered list’ and many results will return.

Using CSS For Styling

CSS stands for **C**ascading **S**tyle **S**heets and is the fun part of receipt templates. With this tool, you can change fonts typefaces, font sizes, move elements, and many other styling functions. CSS must be used with HTML tags either by surrounding plain text with `<span></span>` tags or using existing tags like `<h1>`. Any beginning tag or single tag can be used just by adding this after the name with a space:

```
style=''
```

so that it will look like this:

```
<h1 style=''>
      or
<span style=''>
      or
<h2 style=''>
      or
<br style=''/>
```

Once the `style=''` text has been added to a tag, you can enter any number of colon separated style information. This style information follows a fairly rigid format of property and value statements separated by a colon:

```
property: value;  property: value;
```

Let's say that you have some plain text that you would like to make **bold**. Start with the sentence:

```
This is some text
```

Add a <span> tag (or h1, etc.) before and after the text like this:

```
<span>This is some text</span>
```

Then we can add CSS to the span tag itself to make the text bold:

```
<span style='font-weight:bold'>This is some text</span>
```

These properties and values vary depending on what you would like to do. There are many lists online but here are the most commonly used in receipt templates:

`font-size: 1.5em;`

1em is the default font size while 1.5em is 1.5 times the current size and so on. It can be used make text larger or smaller very quickly.

`font-family: Calibri;`

Changes the style of font from the default to any font on the computer but in this case it is being changed to Calibri. You can theoretically name any font that is installed on the workstation. If a specific font is needed or the font name has spaces in it, be sure to surround the font with quotation marks: 'Segoe UI'.

`text-align: center;`

Changes the justification of the text. Use 'center' to center the text, 'right' to justify the text to the right, and 'left' to justify to the left though left is the default if another alignment is not specified

`position:relative;` with **`top:1px;`** or **`bottom:1px;`** or **`left:1px;`** or **`right:1px;`**

`position:relative`, combined with either `top`, `bottom`, `left` or `right`, can be used to nudge specific text, pictures, or tables in any direction on a receipt. For example, if you created a table and the table seemed as though it was just a little to the left of where it should be. You could add style=`'position:relative; left:2px;'` and the table will be nudged 2 pixels **away from** the left side (or towards the right) of the receipt. These properties always move away from the side of the receipt that is named. So using `bottom` will push the object up, `top` will push it down and `right` will push the object to the left.

`font-style:italic;`

Italicizes the text

`text-decoration:underline;`

Underlines the text

`font-weight:bold;`

Bolds the text

Let's take the 'welcome to' example that we used before and add CSS to make it more attractive. I'm going to add horizontal rules, center and bold all of the text, as well as make part of it italicized:

Code:

```
<hr/>
     <h2 style='text-align:center; font-weight:bold;'>
          <span style='font-style:italic'>Welcome to the</span>
          <br/>
          Free Public Library
     </h2>
<hr/>
```

Result:

Welcome to the

Free Public Library

Template Toolkit Variables

Template Toolkit Variables (just ‘variables’ from now on) can help add great amounts of information to a receipt such as the title of an item to a checkout receipt or the barcode of a patron when a bill is paid. Just like variables in algebra, the value of the variable changes depending on the context. The variables look like this:

%LIBRARY%

A variable is made up of a word surrounded by percent signs and in this case the variable would be the name of the library. The variable names are meant to be readable so their meaning can generally be known at sight. Open up the receipt template editor, choose the checkout receipt, and click on the ‘Macros’ button in the upper right corner. A popup list will appear with two sections. Most of the variables listed in this window will best work with the checkout receipt. When switching to another receipt, be sure the press the macros button again to refresh the list of variables. This way, the popup list will always have the variables that will work best with the currently open receipt. Here are the most common variables for all receipts:

%LIBRARY%
%PATRON_BARCODE%
%title%
%author%
%barcode%
%call_number%
%due_date%

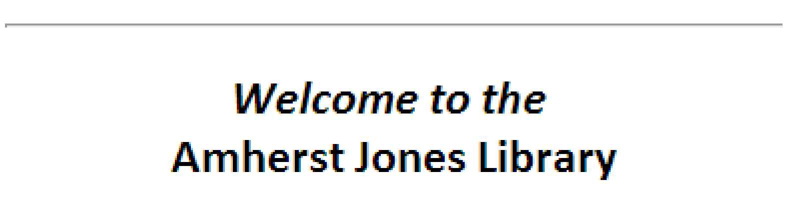
These are basic variables but you can see how easy they are to read. To explain how these variables work more clearly, let’s take the example of a checkout receipt. Say that you wanted to add the author of the item being checked out to the receipt for the patron’s reference. You would add %author% to the line item section of the receipt because, when a receipt is printed, Evergreen will go through the line item section as many items as there are items being checked out. For each item that is checked out, Evergreen will see the %author% variable, fetch the author information for the item and print it in the receipt. These variables can be added anywhere in the receipt whether in or out of an HTML tag.

Let’s now add the library name variable to the ‘welcome to’ code that we have been working with:

Code:

```
<hr/>
      <h2 style='text-align:center; font-weight:bold; '>
            <span style='font-style:italic'>Welcome to the</span>
            <br/>
            %LIBRARY%
      </h2>
<hr/>
```

Result:



FAQs and a few cool functions

Here are a few frequently asked questions and advanced functions of receipt templates that will make other staff jealous of your receipt template prowess.

Custom note about book sale

Adding a custom note to your receipt template can be as easy as adding plain text. For example, a note could be added about an upcoming book sale to the bottom of your checkout receipt. In the footer section of the checkout receipt, add the following text substituting alternate text if necessary:

```
Don’t forget about the library book sale on August 5th!
```

Pretty boring by itself but we could add HTML and CSS to make it more appealing. Here, I am changing the font and font size as well as centering with a few line breaks:

Code:

```
<h2 style='font-size:1.3em; text-align:center; font-family:”Segoe
UI Semibold”;'>Don’t forget about the<br/> library book sale
on<br/> <span style='font-size:2em'>August 5th!</span>
       </h2>
```

Output:

Don’t forget about the
library book sale on
August 5th!

Remove staff name from the bottom of a receipt:

Many receipts will print the staff member’s name that is logged into Evergreen when the receipt is printed. The code that prints the staff name will look like this:

```
You were helped by %STAFF_FIRSTNAME%<br/>
```

To stop the staff member’s name from appearing on the receipt, simply remove this line from the receipt. Here are all of names of receipts that have the staff member’s name at the bottom:

items_out
renew
checkout
holds
holds_on_bib
holds_for_patron

holds_shelf
holds_pull_list
hold_slip
transit_slip
hold_transit_slip

Increasing the length of a receipt:

To add length to the receipt, add a few
 tags at the bottom of a receipt's. After a few of the break tags (about six
 tags equal an inch) add a <hr/> tag. Without the ending <hr/> tag the other
 tags will not print. There will be a line at the end of the receipt but we can make it less visible using the following CSS:

```
<hr style="border:1px dashed lightgray;"/>
```

This CSS will change the solid black line to a dashed gray line.

Adding a picture to your receipt

Adding a picture to a receipt can make it unique and look appealing to patrons. The picture does need to be accessible by URL (like on a website) in order to work. Start by adding the following text to the section of the receipt where you would like the picture to go:

```
<img src='RESOURCE URL' style='width:50px; height:50px;'/>
```

You will need to replace the text 'RESOURCE URL' with the URL of the image. In this case, I am taking an image from a website and adding it to the header portion of a receipt You may need to change the width and height of the image to ensure the proportions are correct. The position of the image may need to be changed as well but this is the basic structure of adding an image.

An excellent example is to place a picture to the left of a body of text. In this case, I will be placing an image next to our welcome message using an HTML table. Notice the change to the width and height of the image:

Code:

```
<table>
	<tr>
		<td>
<img
src='http://cwmars.org/sites/intranet.cwmars.org/files/images/tree.jpg
'  style='width:90px; height:90px;'/>
	</td>
		<td>
	<h2 style="text-align:center">
			<i>Welcome to the</i>
			<br/>
			<span style="font-size:1.1em">Amherst Library</span>
			</h2>
		</td>
```

```
    </tr>
</table>
```

Output:

Moving an object up/down/left/right using the CSS position property

In the last example, we added a picture to the text but as you can see the text and the picture do not line up very well. Moving the image up slightly will help it to look centered next to the text. Moving the picture can be done by inserting the following code into the picture tag's styling from the above example:

Code:

```
<img
src='http://cwmars.org/sites/intranet.cwmars.org/files/images/tree.jpg
'  style='width:90px; height:90px; position:relative; bottom:5px;'/>
```

Result :

Removing the time on a date stamp using the %SUBSTR% function

There is a great function that the receipt template editor uses that allows staff to cut off part of a variable's output. The most common use for this is to remove the time portion from a date/time stamp. For example, the date due on a checkout receipt usually looks like this '12/15/14 11:59 PM'. To remove the time, we can use a function called '%SUBSTR%'. It looks like a Template Toolkit variable in that it is a word surrounded by percent signs. Here is what %SUBSTR% looks when used:

%SUBSTR(0,11)% TEXT TO BE MODIFIED %SUBSTR_END%

While it does look like a variable, it requires a beginning and an end sort of 'tag'. However, there are also comma separated values surrounded by the parentheses attached to the first part of the function. These numbers are very important. The first number indicates the character at which Evergreen should start printing the enclosed text and the second is how many characters to print. In this case, Evergreen would start printing the text at the first letter of the text string because %SUBSTR% considers the first character of a text string to be in position zero. Evergreen would then print eleven letters so that it will look like this in the end:

Code:

```
%SUBSTR(0,11)% TEXT MODIFIED %SUBSTR_END%
```

Result:

TEXT MODIFI

%SUBSTR% counts spaces as characters. Using %SUBSTR% to remove the time from variables such as a %due_date% does not always work perfectly because of the way that due dates print. For example, the due date could be 12/15/14 which would put the output text as 12/15/14 11:59 PM. Using (0,9) as the numbers for %SUBSTR% works well in this case and prints out just the date.

Code:

```
%SUBSTR(0,9)% %due_date% %SUBSTR_END%
```

Result:

12/15/14

But what if the due date were 3/3/14? The output would be 3/3/14 11:59 PM without using %SUBSTR%. There are two fewer characters in this date as compared to 12/15/14 so using (0,9) in this case would print out part of the time and look like this printing out part of the time:

Code:

```
%SUBSTR(0,9)% %due_date% %SUBSTR_END%
```

Result:

3/3/14 11

The solution is to use negative numbers. The number in the %SUBSTR% function always tells the system what to print. If negative numbers are entered, this will tell the system where to start printing not from the beginning of the variable text but from the end. In this case, what we need to cut off is ‘ 11:59 PM’ which is 9 characters including the space before the ‘11’. We then pick an arbitrary stop number of -100 so that the numbers to use would be (-9,-100)

Here is a complete solution for removing the time from the due date variable:

Code:

```
Due: %SUBSTR(-9,-100)% %due_date% %SUBSTR_END%
```

Output:

Due: 2013-03-27

Using a barcode font to display real barcodes on receipts

Adding the actual barcode of an item and/or patron to a receipt can help staff members get information about a transaction more quickly. Plus it looks cool. For example, if a patron would like more information about a set of checkouts, they could hand the receipt to a staff member who could scan in the items or retrieve the patron from the barcodes on the receipt.

To add the barcode to the receipt, a Codabar barcode font must first be downloaded and installed on the workstation. The following link will download the font required in a zipped format: http://freebarcodefonts.dobsonsw.com/downloads/freewarecodabarfont.zip. Open the file and open the folder called ‘freewarecodabarfont’. Next, drag and drop the two TrueType font files to the desktop. They will be called ‘CodabardLarge’ and ‘CodabarMedium’. Select both files, right-click on one of them, and select ‘install’. The fonts will now be installed on your computer and can be used for word processing, presentations, and Evergreen receipts.

Next, go to the receipt template editor and select a receipt. In this tutorial, I will be editing a checkout receipt. Let us first add the patron barcode to the bottom of the receipt. Add the following code to the Footer portion of the receipt:

```
<span style="font-family:'CodabarLarge';
font-size:1.25em; text-align:center;">
*%PATRON_BARCODE%*
</span>
```

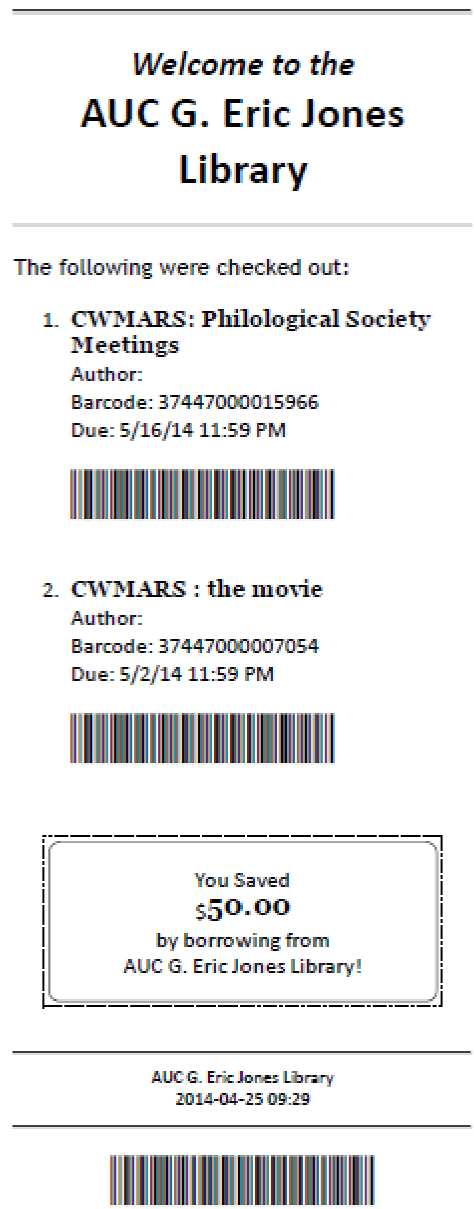
Breaking this down, we are using a span tag to style the variable %PATRON_BARCODE%. The styles here are what actually change the font from the default to the barcode font. Here are the styles listed:

```
      font-family:'CodabarLarge';
font-size:1.25em;
```

The font-family code changes the font of the barcode to the barcode font we installed from freebarcodefonts.com. The font size resizes the font so it looks proportional on the receipt.

A last note concerns the asterisks(*) surrounding the template toolkit variable %PATRON_BARCODE%. The asterisks need to surround what is being barcoded in order that a scanner will actually be able to read the printed barcode. To add the item barcode(s) to the receipt, the same text as above will need to be added to the Line Item section but with a different variable. For item barcode, the variable will be %barcode%.

In the end, a barcode can be printed for each item being checked out as well as the patron barcode at the bottom of the receipt or anywhere on the receipt itself. The receipt to the right was created combining all of the techniques in this tutorial.


Welcome to the

AUC G. Eric Jones Library

The following were checked out:

1. **CWMARS: Philological Society Meetings**
   Author:
   Barcode: 37447000015966
   Due: 5/16/14 11:59 PM
2. **CWMARS : the movie**
   Author:
   Barcode: 37447000007054
   Due: 5/2/14 11:59 PM

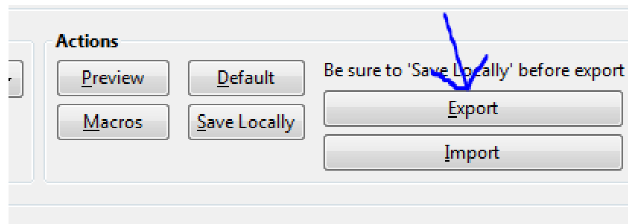
You Saved
$50.00
by borrowing from
AUC G. Eric Jones Library!

AUC G. Eric Jones Library
2014-04-25 09:29

Transferring Receipt Templates between Workstations

We recommend that all receipts be edited/created on one workstation and the edited receipts be transferred to other workstations to help to prevent version control issues. Keeping a backup of the receipts somewhere is also a great way to preserve your work in case of workstation failure, upgrade concerns, etc. The process of transferring receipts starts with exporting the receipts, then transferring to another workstation, and lastly importing them to another workstation. If backing up, only export and transfer may be necessary.

Export



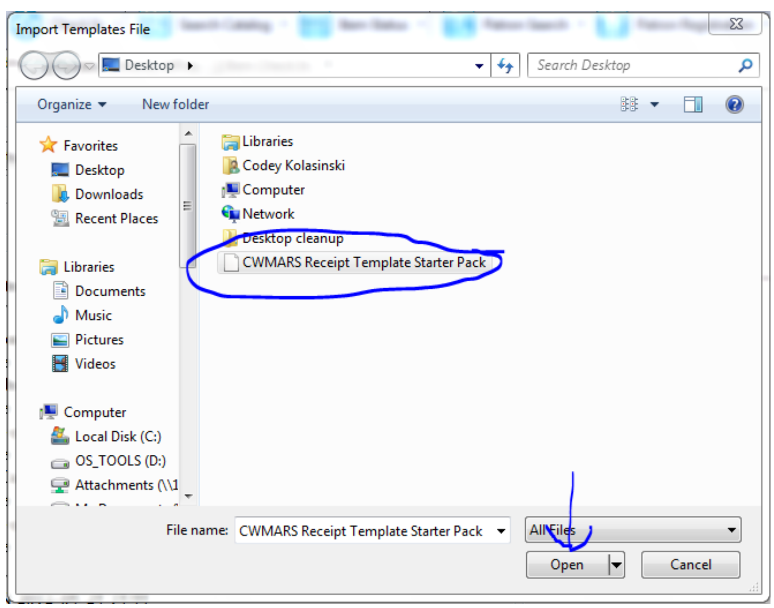
Start by customizing all of the templates on one workstation so they are exactly the way that you/the library would like them to appear. Next, click the ‘Export’ button in the upper right corner of the Receipt Template Editor screen and a pop-up box will appear which will allow you to name the file and choose an export location. Make the export location the desktop of the workstation for now but we will look at a few different ways of getting the file to any other workstations. The file exported from the editor contains all receipt templates so there is no need to export individually.

Transfer

Once the file is on the desktop, we must find a way to move it to other workstations. The file could be placed on a USB drive and the drive brought to the other computers. If the same network drive is available on all computers, the file could be placed on the drive by one workstation and accessed on the other. One of the easiest ways to transfer the files is to email the file to yourself and to then retrieve the file on other workstations using a web-based email client. Any way of transferring the file to another workstation is valid as long as the file ends up being accessible.

Import

After moving the receipts to the target workstation’s desktop, login to Evergreen and go to the Receipt Template Editor. Click the ‘Import’ button just below the ‘Export’ button. A pop-up box will appear. Then navigate to the receipt file in the pop-up. Once the file is found, either double click on it or click it and click the open button and all receipts will be imported.

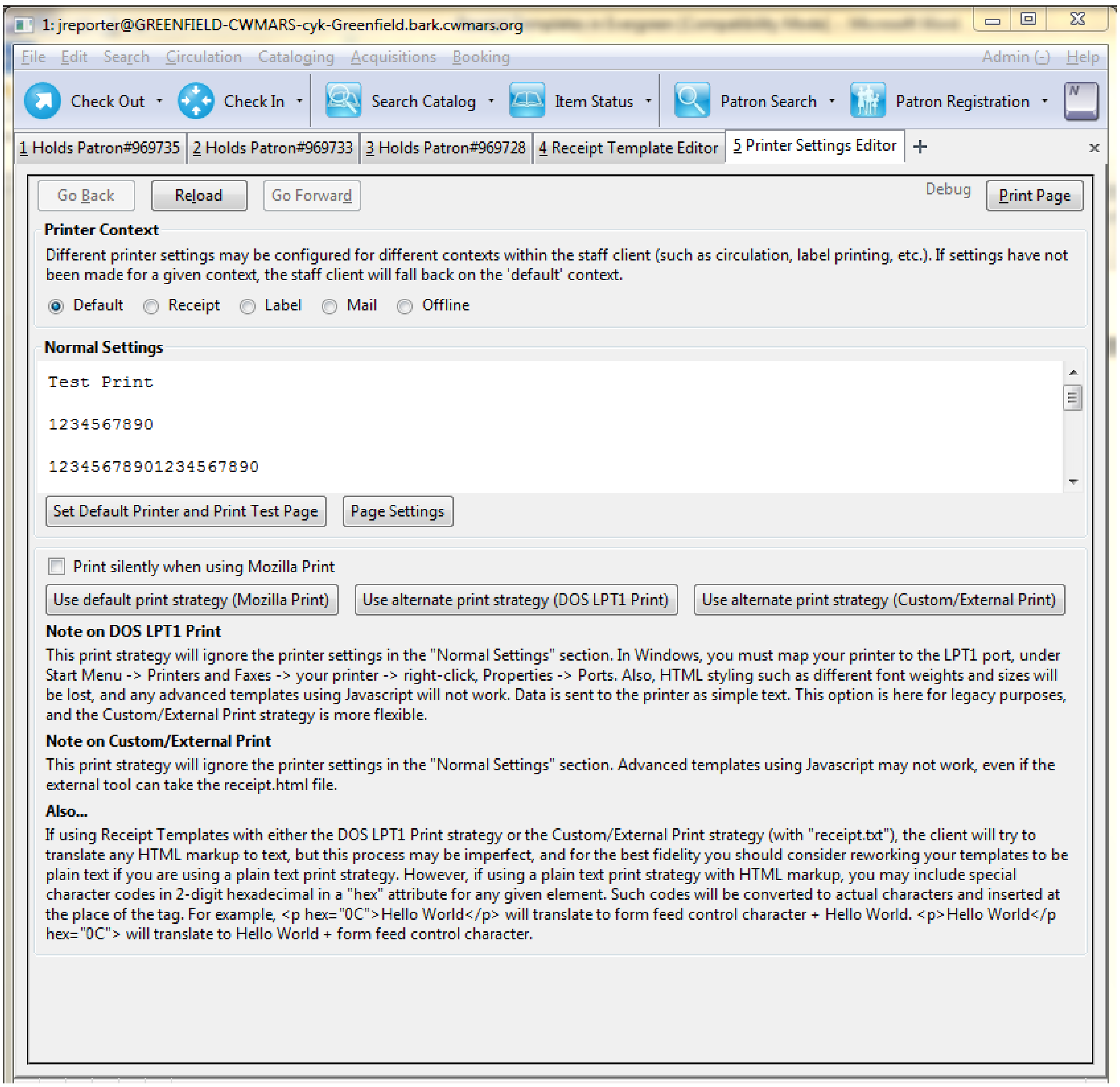


Printer Settings Editor

The Printer Settings Editor is a critical portion of the receipt setup process but also setting up an Evergreen workstation in general. There are several areas in the editor that need to be updated for a workstation to run smoothly. The Printer Settings Editor can be found by going to Admin, then Workstation Administration and finally to Printer Settings Editor.

Layout

There are two sections to the printer settings editor pager: Printer Context and Normal Settings.



Printer Context

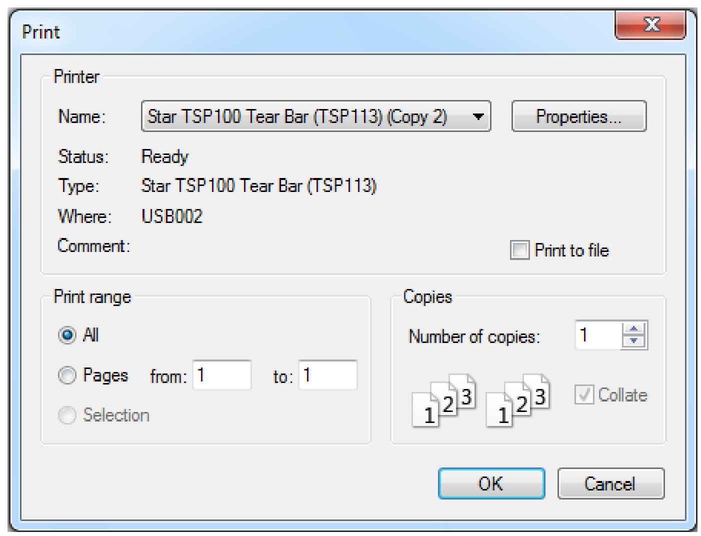
Printer contexts allow different printer settings to be in force when printing in different areas of Evergreen. For example, if a specific printer could be used for printing (spine) labels and another for receipts. This configuration can be set up using printer contexts. To create different settings for the different contexts, select the context by moving the radio button and proceed to make the changes under Normal Settings section. When you are done making changes for a context, move the radio button to add settings to another context or close the editor. Changes will automatically be saved.

Normal Settings

Normal Settings is where printer settings are actually edited. There are three categories including Default Printer, Page Settings, and Print Strategy.

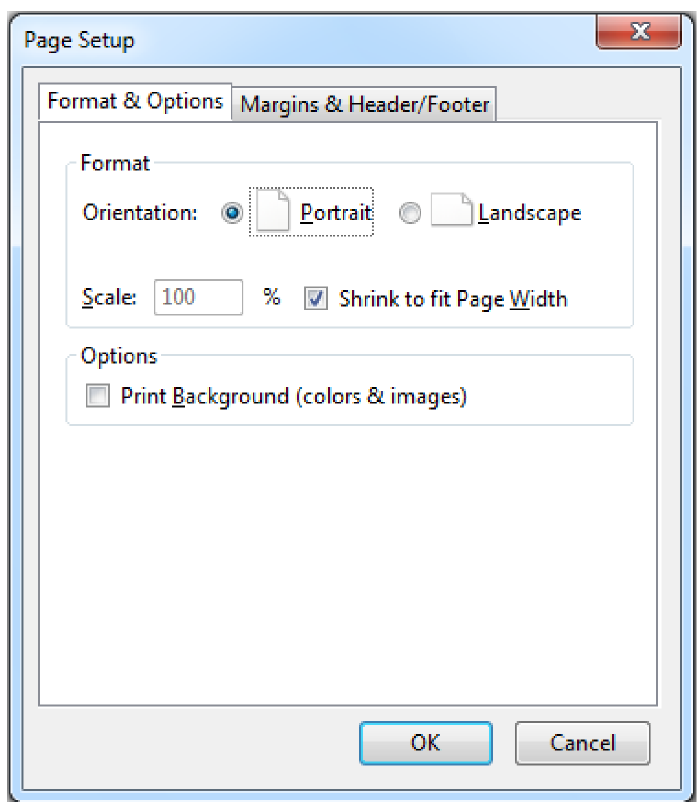
Set Default Printer and Print Test Page

This is where the default printer is set for a context and is done by clicking on the button 'Set Default Printer and Print Test Page'. When the button is pressed, a regular print pop up will appear and a printer can be selected from a drop box. Click 'OK' after selection and a test print page will be sent to the chosen printer. Once a test page is printed, that will be the default printer for that context.



Page Settings

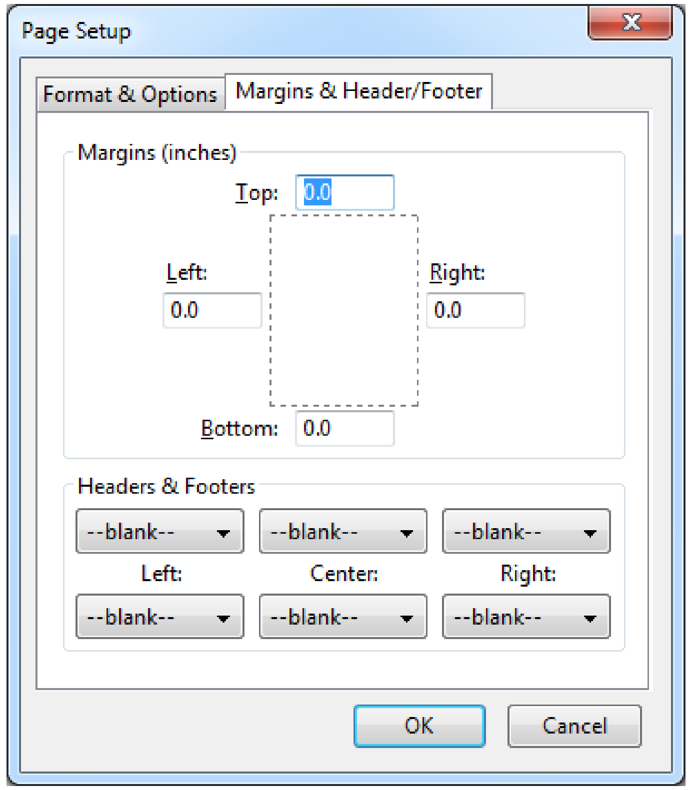
Clicking on Page Settings button will produce a popup box with two tabs: 'Format & Options' and 'Margins & Header/Footer'



<u>Format & Options</u> contains options for the format of the printout including orientation and scale.

Orientation should generally be left on 'Portrait' because the suggested receipt printer (TSP 100) is a portrait oriented printer.

Scale is the other option which allows the relative size of the printout to be changed. However, the option 'Shrink to fit Page Width' is a good option to leave set because the receipt will automatically scale to fit the receipt printer no matter the size of what is being printed.



Margins & Header/Footer contains options for margins and for the header/footer of the receipt. We recommend that the margins all be set to zero because if the margins are left at .5 the receipt will be printed with a full eleven inches of receipt regardless of the length of what is actually being printed. The text will also appear small. Setting the margins to zero will reduce the length of the receipt as well as make the print appear larger.

The drop boxes in the Headers & Footers section usually have Title, URL, or a Date options selected. These options will print the date/time or the title in the corners of the receipt and can look unsightly. To remove these options, set each of the drop boxes to '--blank--'.

Print Strategies

NOTE: Unless using an older printer that uses a parallel port to connect to the computer, print strategies can generally be ignored.

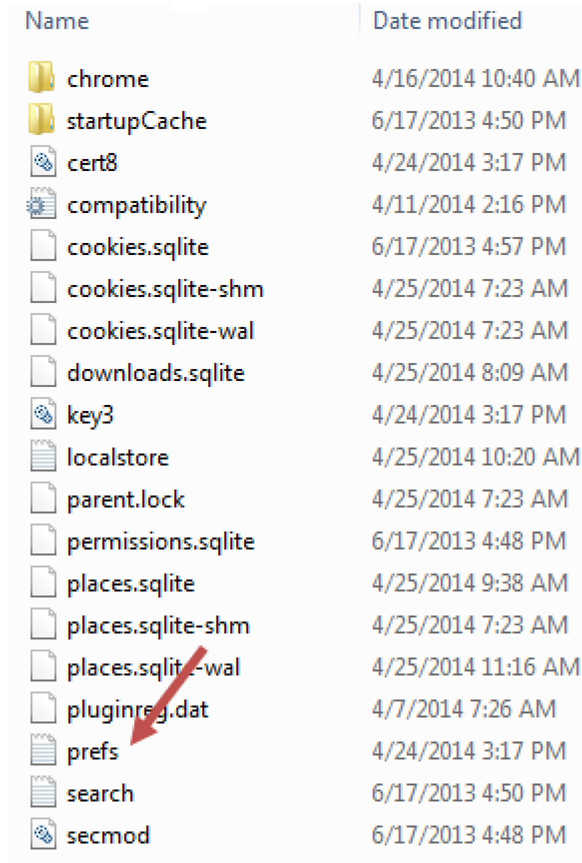
There are three print strategy options but only two are important in this context.

- Use default print strategy (Mozilla Print)
- Use alternate print strategy (DOS LPT1 Print)

Mozilla print is the default and the strategy that should be used for the recommended printer (TSP 100). It is a more regularly used Windows print strategy and, again, is the default so nothing needs to be done for this option to be used. LPT1 in this context allows Evergreen to print to older printers using a parallel port.

Backup up receipts and printer settings manually

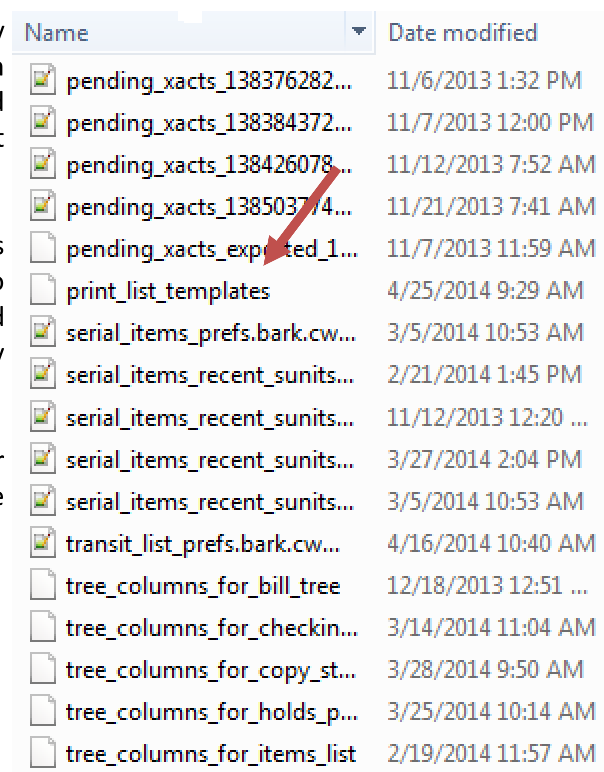
Whether adding a workstation, replacing one that has crashed, or swapping out on that is dated, setting up a workstation exactly as it was on the workstation before can help staff hit the ground running in terms of productivity. What follows will show how to access both the receipt template file on the workstation hard drive for backup or transfer to a new workstation.



Start by going to the computer's hard drive by going to 'Start' => 'My Computer' for XP and 'Start' => 'Computer' (for Windows 7). Next, go to the 'Users' folder. The next screen will list all users that have ever logged into the computer. Open the folder of the user that uses Evergreen most often. Next, open the 'AppData' folder, then 'Roaming', 'OpenILS', 'open_ils_staff_client', 'Profiles'. The next screen should only contain one folder so enter it. If there is more than one folder, see note at the bottom of the page. **

This is the folder where Evergreen stores all of its local settings info. Here is the file to make copies of in order to back up the receipts templates:

Chrome/print_list_templates
//in the 'Chrome' directory



Please do not remove the file from the folder but simply make a copy on the desktop, a network drive, or a flash drive. If a workstation ever breaks down or you would simply like all workstations to have the same receipt templates, this is the way to go.

WARNING: When moving the copying the backed up files onto another workstation, the workstation will need to have been logged into Evergreen at least once and Evergreen must be completely closed out of before any preferences files be moved.

* One may need to type 'AppData' into the file browser navigation bar in order to get to the screen where 'Roaming' is entered.
** If there is more than one profile in the 'profiles' folder, back up each profile's column configuration if desired.

Which receipt does what

Below is a list of all receipt templates and where they are used in Evergreen itself.

Receipt Name	Location in Evergreen where used	Public or Staff Receipt
item_status	item status screen	Staff
transit_list	transit list under Admin => Local Administration	Staff
items_out	patron items out screen (top and bottom)	Public
renew	renew screen	Public
checkout	patron checkout screen	Public
offline_checkout	offline checkout	Public
checkin	checkin screen	Staff
bill_payment	patron bill payment screen	Public
bills_historical	patron bills history	Public
bills_current	patron main bills screen	Public
offline_checkin	offline checkin	Staff
offline_renew	offline renew	Public
offline_inhouse	offline in-house	Staff
in_house_use	in-house use	Staff
holds	?	Staff
holds_on_bib	the view holds screen in the OPAC	Staff
holds_for_patron	patron holds screen	Public
holds_shelf	hold shelf screen	Staff
holds_pull_list	pull list screen	Staff
hold_slip	checkin screen when and item is going to the hold shelf	Public/Staff
transit_slip	checkin screen when an item is NOT going to fill a hold	Staff
hold_transit_slip	checkin screen when an item is going to fill a hold	Staff
holdings_maintenance	holdings maintenance screen	Staff