

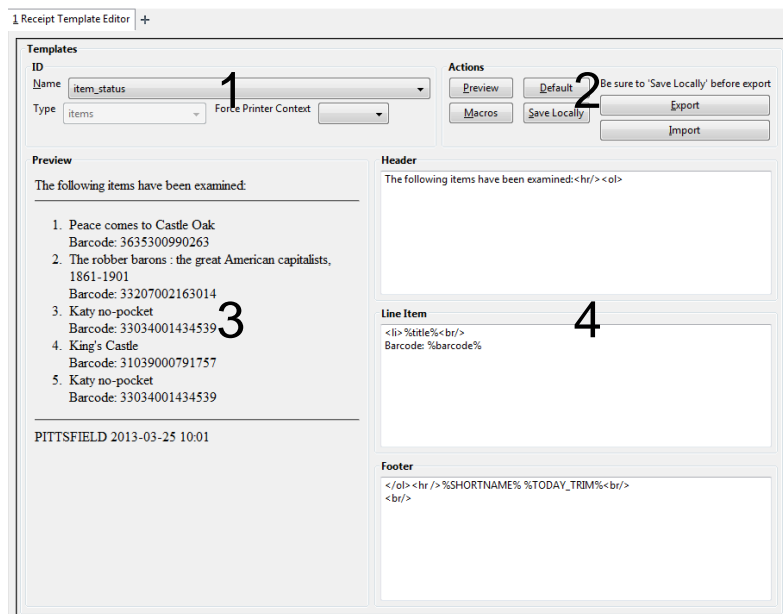
Receipt Templates in Evergreen

Customizing Receipt Templates in Evergreen	2
Adding Plain Text	3
HTML Tags	3
CSS.....	7
Template Toolkit Variables	9
FAQs and a few other functions.....	10
Transferring Receipt Templates between Workstations	14
Export.....	14
Transfer	14
Import	14
Printer Settings Editor.....	15
Layout.....	15
Printer Context.....	15
Normal Settings.....	16
Set Default Printer and Print Test Page	16
Page Settings.....	16
Print Strategies.....	17
Where Receipts are used in Evergreen	18

Customizing Receipt Templates in Evergreen

The receipt function created for the Evergreen ILS allows staff to customize each receipt to the local tastes and needs of patrons and library staff. These customizations can be done using a combination of plain text, HTML/CSS and Template Toolkit variables but we will talk about each of these items 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, either click on Receipt Template Editor button on the first screen that is seen when you login to Evergreen or 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 picture below.

1. This first section in the upper left corner lists the receipt currently being edited. It defaults to 'item_status' when opened but it 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 the text for that receipt.



2. Section 2 is an admin section with a list of buttons. Here is what each button can do:

- **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** – This button 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 more 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** – This button 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** – This button 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** – This button will bring up a standard open file window and, when you find an exported receipt template file, will import a batch of 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 almost exact and one of the only issues is that the preview window is usually wider than receipt paper so a receipt may appear slightly crowded when printing.
 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. This will be further explained when we get to Template Toolkit Variables below.

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 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:

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

That is really it for adding plain text. 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 mostly 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 are the 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. However, a tag that exists on its own will have a slash at the end of the tag like this:

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 **break** tags. They will cause a line break to be created in text so 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 ordered list tags and create a numbered list of items that, if combined with tags (list item), 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 couplet to work. Then the text of the item in the list must go between the 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>

```

Result:

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

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

The table tag starts and ends the table but each row will start and end with a <tr> tag. Each piece of cell data will need to be put into a set of <tr> tags and 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 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.

CSS

CSS stands for Cascading Style Sheets 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 `` 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 separated by a colon like this:

```
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 `` 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 but there are lists online and here are the most commonly used in receipt templates:

font-size: 1.5em;

1em is the current size while 1.5em is 1.5 times the current size and so on. It can make text large very quickly.

font-family: calibri;

Changes the style of font from the default to Calibri. You can theoretically name any font that is installed on your computer.

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;**

When `position: relative` is combined with either `top`, `bottom`, `left` or `right`, these can be used to nudge specific text, pictures or tables around 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 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 the title of an item to a receipt when it is checked out 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 made to be readable as you can know what they mean when looking at a list. Open up the receipt template editor, choose the checkout receipt, and click on the Macros button in the upper right corner. A list will appear with two sections. Most of the variables listed in this window will best work with the checkout receipt. When you switch to another receipt and press the macros button again the list of variables will reset to the list that will best work with that new receipt. The most common for checkout are the following:

```
%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 for as many items as are 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.

If you would like to add the title of an item to a bill, you can copy and paste from the macros window or type in the title variable where you would like it in the receipt. 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:

Welcome to the
Amherst Jones Library

FAQs and a few other 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 your upcoming book sale to the bottom of your checkout receipt. In the footer section of your checkout receipt, add the following text substituting your own 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 of the receipts have plain text and a variable at the bottom of them that will print the staff member's name that is logged into Evergreen when the receipt is printed. The code will look like this:

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

To stop the staff member's name from appearing on the receipt, it is as simple as removing 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 the footer of a receipt and then at the end of the `
` tags add one `<hr/>` tag. There are approximately six break tags to an inch. Without the `<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;"/>
```

Adding a picture to your receipt

Adding a picture to your receipt can make it unique and look appealing to your patrons. The picture does need to be on the internet so that you can have the URL or web address of the picture. 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 the receipt will then display an image in the header. You may need to change the width and height of the image to get the proportions correct or to position the image correctly but this is the basic structure of adding an image.

If you wanted the image to the left of your welcome message you could use a table to make something like this. Notice that I did need to change the width and height:

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

```

                <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. I would take the picture and move it up a little to make it 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:

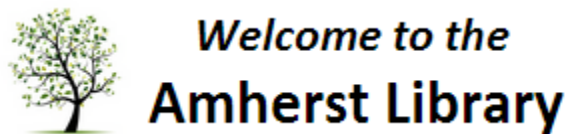
Code:

```

<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 datestamp using the %SUBSTR% function

There is a great function that the receipt template editor uses that allows staff to limit the output of variables. For example, if you use a variable like %TODAY_TRIM% which will print the current time and date it would look like this: 2013-03-27 12:36. Many staff members do not like to include the time as it is confusing and not usually added as part of a date for daily use. To remove this time portion from the above we can use the substring function that looks like this: %SUBSTR%. As you can see, it is almost 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 MODIFIED %SUBSTR_END%
```

While it does look like a variable, it requires a beginning and an end sort of 'tag'. However, you can also see the comma separated values surrounded by the parentheses in the first part of the

function. These numbers are very important. The first number indicates the character at which Evergreen should start printing the letters of the text of the variable 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:

TEXT MODIFI

As you can see %SUBSTR% does count spaces as characters. Here is a complete solution for including the date on a receipt but not including the time:

Code:

```
Day & Time: %SUBSTR(0,10)% %TODAY_TRIM% %SUBSTR_END%
```

Output:

```
Day & Time: 2013-03-27
```

There are several date formats that Evergreen uses on receipts and the above will work with most. However, the due date format is slightly different and requires a different technique. The following technique can be used in the checkout, renew, and items out receipts. Normally, the due dates on receipts will print like this:

```
Due: 3/4/14 11:59 PM
```

Many staff members do not like to include the time because it can be confusing for patrons who may say that they have to until the end of the day to return items. To remove the time, we can use the SUBSTR technique that we use above but using negative numbers instead:

Code:

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

Output:

```
Due: 3/4/14
```

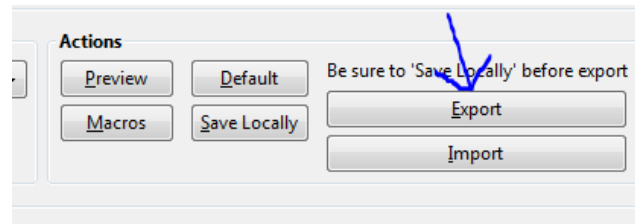
What this code is saying is to start printing nine character positions from the end of the string and print out to 100. 100 is an arbitrary number but anything over ten or so will work.

Transferring Receipt Templates between Workstations

We recommend that receipts be customized on one workstation and transferring the changes to any others helps 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, 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 Edit Templates 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 that is exported from the editor contains **all** receipt templates so receipts do not need to be individually exported.

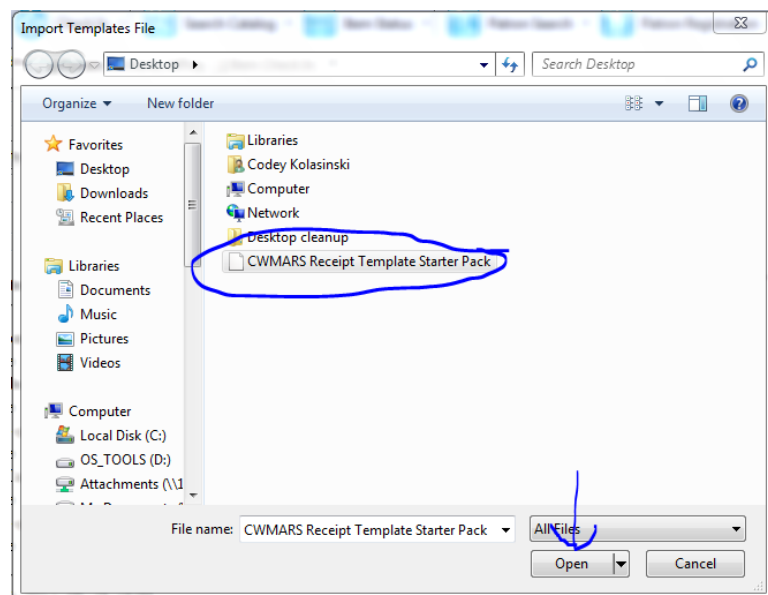


Transfer

Once the file is on the desktop we must find a way to get it to any other workstations. The file could be placed on a USB drive and the drive brought to the other computers. A network drive would work well if it is connected to all other computers. One of the easiest ways to transfer the files is to email the file to yourself and to then retrieve the file on any other workstations using a web-based email client. There is a client available at mail.cwmars.org if you have a cwmars.org email address where staff can log in from anywhere making the transfer easy.

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 and in the box navigate to the receipt file. Once the file is found, either double click on it or click it and click the open button.

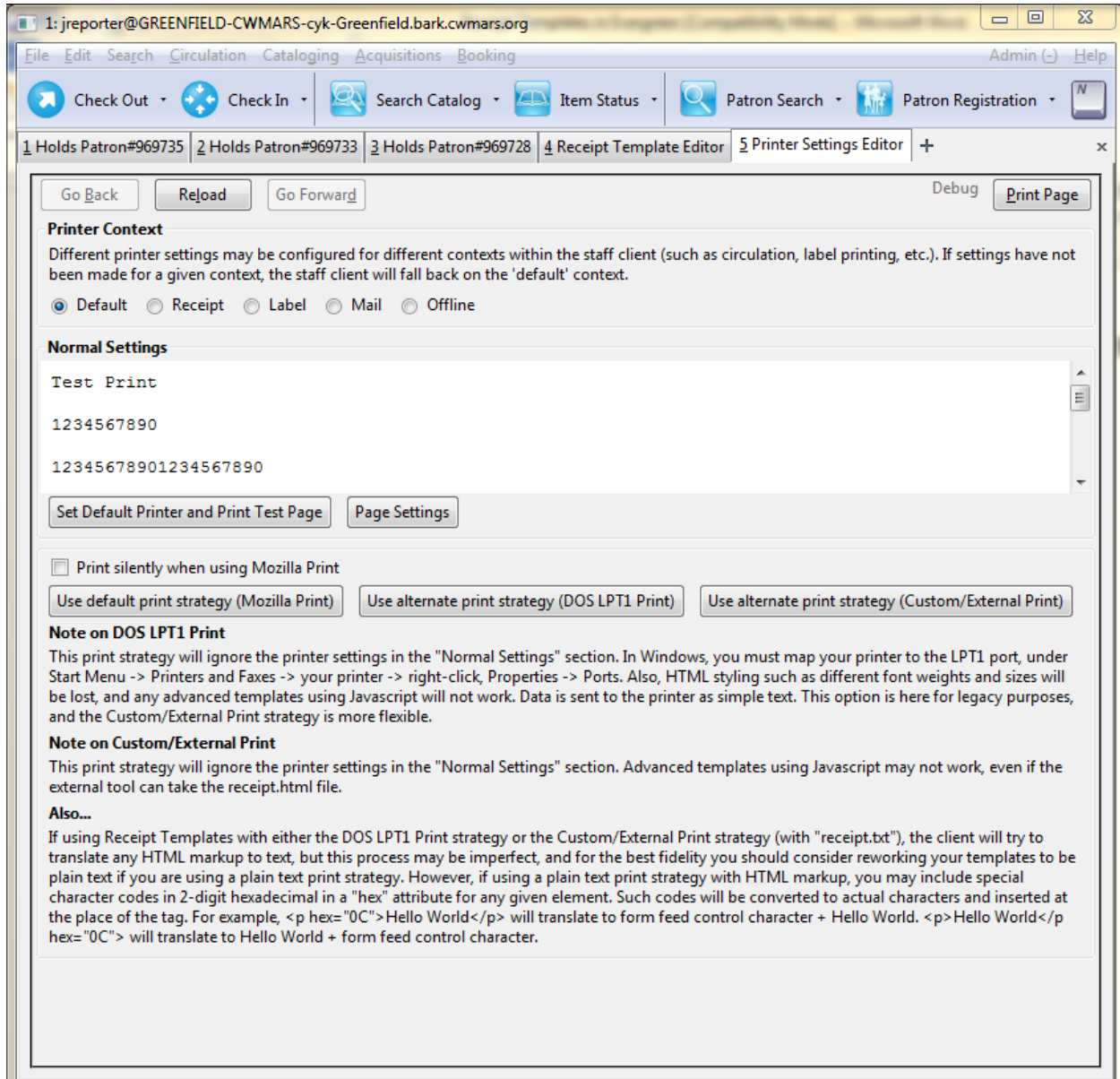


Printer Settings Editor

The Printer Settings Editor is a critical portion of setting up receipts 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 should be used for printing (spine) labels and another for

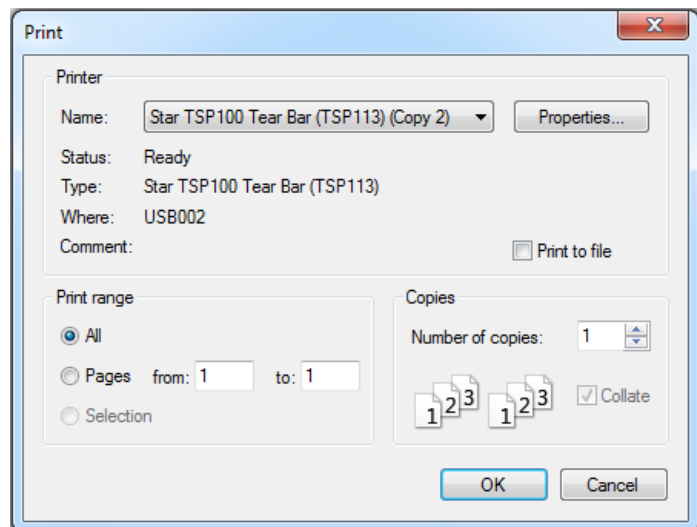
receipts, this configuration can be set up using printer contexts. To create different settings for the different contexts, select the context by moving the radio button and proceed to make the changes under Normal Settings section. When you are done making changes for a context, move the radio button 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 applied. 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 the context and is done by clicking on the button 'Set Default Printer and Print Test Page'. A regular print pop up will appear and a printer can be selected from the 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.



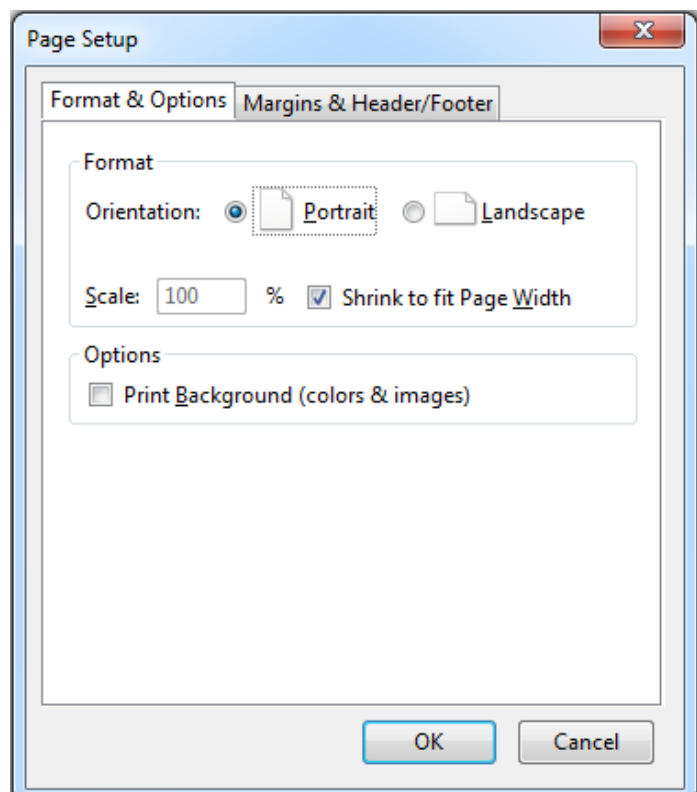
Page Settings

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

Format & Options contains options for the format of the printout including orientation and scale.

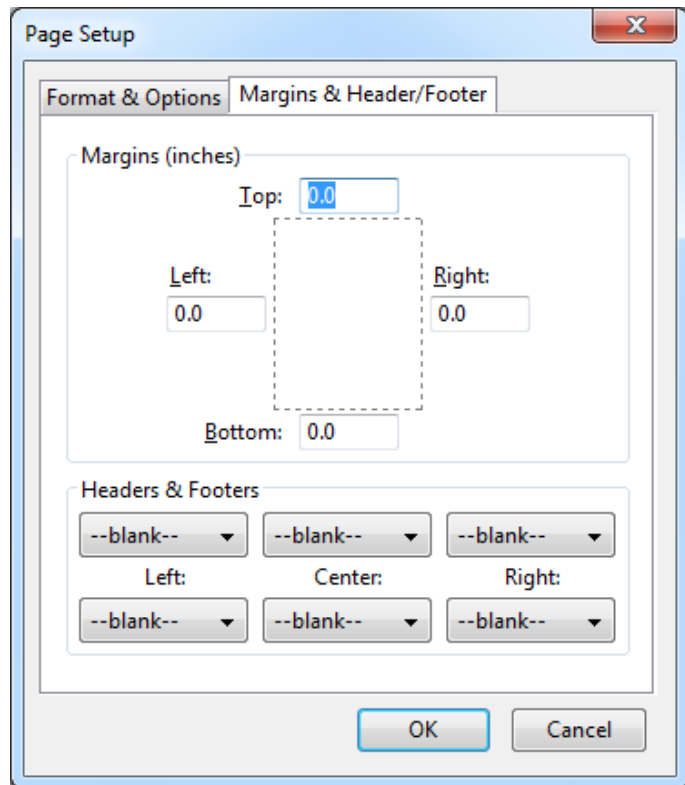
Orientation should generally be left on 'Portrait' because the suggested receipt printer (TSP 100) is a portrait oriented printer. However, there is an option to print in Landscape.

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 another 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 print a full eleven inches of receipt regardless of the length of what is actually being printed. Setting the margins to zero makes the printer only print what is being sent to the printer. The drop boxes in the Headers & Footers section usually have Title, URL, or a Date options selected but what these will do is print the date/time or the title of the receipt and can look unsightly. To remove these options, set each of the drop boxes to '--blank--'.



Print Strategies

NOTE: Unless the printer being used is a little older and 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. LPT1 in this context allows Evergreen to print to older printers using a parallel port.

Where Receipts are used in Evergreen

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

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