

**Receipt Generation Guide**

**YSoft SafeQ**

**Payment**

**Machine**

Copyright © 2011 Y Soft Corporation. All rights reserved.

Edition 1.1 [EN]

Date: 1 March 2014

YSoft SafeQ, the YSoft SafeQ logo, and the Y Soft logo are registered trademarks of Y Soft Corporation. All other trademarks and names stated in this document may be trademarks and/or registered trademarks of individual owners.

www.ysoft.com

Y Soft Corporation Y Soft North America  
Technická 2948/13 [info.us@ysoft.com](mailto:info.us@ysoft.com)  
Technology Park  
616 00 Brno  
Czech Republic

[info@ysoft.com](mailto:info@ysoft.com)

**Disclaimer**

The producer does not bear any responsibility for the use of any functions not specified here unless covered by a separate document.

The producer rejects any and all responsibility for the consequences of improper, negligent or incorrect installation, neglected maintenance, or settings of optional operational system parameters.

Contents

[Contents 3](#_Toc316305910)

[1 Introduction 4](#_Toc316305911)

[Overview 4](#_Toc316305912)

[Prerequisites 4](#_Toc316305913)

[Definition of terms and abbreviations 4](#_Toc316305914)

[2 Description 5](#_Toc316305915)

[SafeQ Payment Machine Receipt concept 5](#_Toc316305916)

[Receipt generating utility 5](#_Toc316305917)

[3 Creating receipt template 6](#_Toc316305918)

[Prerequisites 6](#_Toc316305919)

[Graphics logo 6](#_Toc316305920)

[Receipt text 7](#_Toc316305921)

[Creating text source 8](#_Toc316305922)

[Creating receipt template 10](#_Toc316305923)

[Preview of template 11](#_Toc316305924)

[4 Uploading receipt template 13](#_Toc316305925)

[5 Clearing receipt template 15](#_Toc316305926)

[6 Template format in details 16](#_Toc316305927)

[Allowed embeddable text tags 16](#_Toc316305928)

[SVG format 17](#_Toc316305929)

[Svg element image 18](#_Toc316305930)

[Svg element text 18](#_Toc316305931)

[Svg element switch 19](#_Toc316305932)

[Inline tags 20](#_Toc316305933)

[7 Tuning receipt template 21](#_Toc316305934)

[Text size and position 21](#_Toc316305935)

[Embedding image with different size 21](#_Toc316305936)

[Inline tags 22](#_Toc316305937)

[Language switch 22](#_Toc316305938)

[8 Relationship with other documents 23](#_Toc316305939)

[9 Version history 24](#_Toc316305940)

# 1 Introduction

## Overview

This guide provides additional instructions for installing and maintaining the YSoft SafeQ® Payment Machine with additional printer. Information in this guide includes:

* Creating of custom receipt template via provided utility
* Uploading procedure
* Receipt template format description
* Fine tune of receipt template

## Prerequisites

Before proceeding, ensure that all of the following items are available.

1. Installed and fully functional YSoft SafeQ Payment Machine
2. Windows notepad or any similar program for editing plain text files
3. SVG files viewer like Internet Explorer, Mozilla Firefox or Inkscape
4. Customers logo image (optional)
5. Receipt generating and sending utilities
6. Knowledge of Service menu level 2 pin (partners pin)

## Definition of terms and abbreviations

|  |  |
| --- | --- |
| SPM | SafeQ Payment Machine |
| SVG | File format - Scalable Vector Graphics |
| XML | Extensible Markup Language |
| Static text | Text that is in template same as on final receipt |
| Tag text | Text in template that instruct rendering engine to place relevant value |
| SVG element | SVG file format is based on XML markup language so contains hierarchical set of elements |
| XML tag | XML elements are defined by starting and ending tags. These XML tags are specific for each element |
| Receipt template | File for SPM that instruct rendering engine how to make clients receipt |

# 2 Description

## SafeQ Payment Machine Receipt concept

Payment machine receipt is fully configurable. Template content completely defines all text, their position, anchoring, size etc. Template also may contain embedded images. There is also possibility to include non-static text; this text is replaced by some value depending on situation. Best example is recharged amount of money. Template only defines that on this position with this size have to be placed recharged amount and payment machine itself renders to defined position current value of deposited money.

Receipt template is internally SVG file format. SVG is common format for vector graphics based on XML. As any other XML based format SVG contains many tags of internal elements. Due to limitations in SPM the rendering engine supports only very limited set of SVG tags. So receipt template could be viewed by SVG capable viewer, but rendering engine could not use normal SVG files from common graphics editors.

## Receipt generating utility

The receipt generating utility allows you to create receipt template from very simple plain text file without knowledge of internal template structure.

According to this very simple use, final receipt template has these limitations:

* None or one image (logo) with predefined size is supported
* Placement of image is fixed to the top of receipt
* Fixed font size
* Fixed line height – space could be done by empty line
* Only 3 alignments – to left or right corner of receipt, or to middle.
* Only static text and text tags are supported (no inline tags)
* Only UTF-8 encoded text
* Text is not automatically wrapped. Maximal number of characters per line depends on content and is about 35-45 chars.

To break these limitation users are of course allowed to modify generated templates or write own template from the scratch.

# 3 Creating receipt template

## Prerequisites

### Graphics logo

This step is optional; receipts could be generated with or without customer’s logo.

If you would like to have logo included you have to prepare it first. Then receipt generating utility includes customer’s logo directly into receipt template.

Used image could be almost any common format (for example JPEG or PNG or BMP), it will be transformed as needed. Size has to be 560x220 pixels, if image has different size it will be automatically transformed. If you supply color image then generating utility will convert it to black and white, but this automatic transformation is not perfect often. Black and white images are used due to we will print them, and printer could not make gray dot. If you need to have any gray color in image you have to use dithering algorithms to produce gray by group of black or white dots. So prepare this image to have only black or white pixels.

Example:

We start with full color logo of one company. It is JPG image 1077x207 pixels.



If we like to have all text black we have to transform image, because automatic transformation will produce orange “Q” as dithered gray.

1. fill black color to orange Q
2. Apply color level to transform all non-white color to black
3. Resize image with respect to ration of image to 560x108 pixels

After that we get following image:



Image when using automatic conversion, image in receipt template will look like this:



### Receipt text

Text of receipt is fully configurable, so you can have any UTF-8 text. As mentioned before receipt template contains static texts and tags – values that will be filled according to situation.

Receipt texts could not be automatically wrapped, so make only short lines. Maximal number of characters per line depends on content and is about 35-45 chars.

Receipt generation utility supports basic alignment support. Text could be aligned to left/middle/right.

Example:

Let’s prepare some basic text, for better understanding tags will be in red:

This is only preview of what text we would like to have, not real text source for generating utility.

Y Soft Corporation

Receipt no.

123456789

User: currently logged user

Old credit: 123$

Recharged: 12$

Thank you

2.10.2014 12:05

When you have problem with recharging

contact local administrator on

phone no. +420 549 240 925 or by e-mail

info@ysoft.com

## Creating text source

Source text file for receipt generating utility must meet these specifications:

* Plain text file
* UTF-8 encoding

Receipt generating utility allows you to use 3 different alignments – left (default), center and right.

All or any combination of these alignments could be used together.

Left – default alignment has no special marks

Change to center aligned text has to be marked by ONE TAB SPACE (**ASCII code 09**).

Change from center aligned text to right aligned text has to be marked also by ONE TAB SPACE.

So if you want to have only right aligned text you have to prepend two TABs before your text.

Examples (TAB space is shown as <TAB>):

Left aligned text<TAB>center aligned text

left aligned text <TAB><TAB>right aligned text

<TAB>center aligned text <TAB>right aligned text

<TAB><TAB>right aligned text

Vertical arrangement is possible only by using empty lines.

To insert text tags simply use correct name from table in section Allowed text tags on page 9 as text of line. Alignment tabs are allowed and other text with another alignment too.

It is recommended to use Windows notepad or similar simple editing tool to prepare source text.

Step by step example:

1. Open windows notepad
2. Fill in text (Following text source is for prepared receipt in previous section). TAB spaces are shown as <TAB>.

<TAB>Y Soft Corporation

<TAB>Receipt no.

<TAB>charge\_time\_stamp

User:<TAB><TAB>logged\_user\_name

Old credit:<TAB><TAB>logged\_user\_oldcredit

Recharged:<TAB><TAB>charge\_value\_sign

<TAB>Thank you

<TAB>charge\_time\_dmyy\_hms

When you have problem with recharging

contact local administrator on

phone no. +420 549 240 925 or by e-mail

info@ysoft.com

1. Save source text. Select In menu “File” -> “Save As…” -> define file name for example receipt.txt and set encoding “UTF-8”. Click Save to finish saving.

## Creating receipt template

For generating receipt template you will need generating utility (folder Generate).

1. Copy source Text file from previous section to folder with generating utility
2. Optionally copy prepared image with logo to same folder
3. Open command line window in same folder
4. a) To generate template with logo image type

“generate\_receipt output\_filename input\_text\_filename image\_filename”

(replace output\_filename to any name for example template.svg, input\_text\_filename to name witch you define in previous section and image\_filename to you logo image).

Example: “generate\_receipt template.svg receipt.txt logo.jpg”

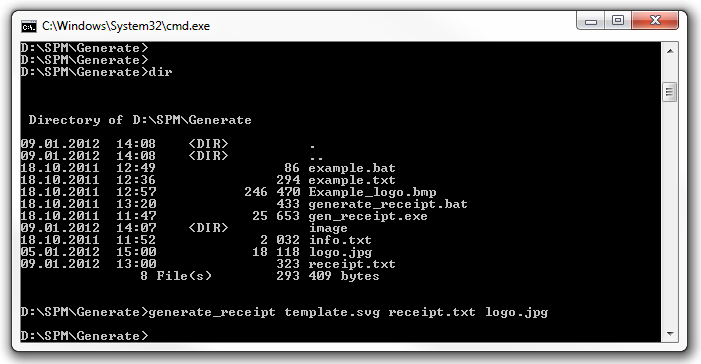
b) To generate template without logo image type

“generate\_receipt output\_filename input\_text\_filename”

(replace output\_filename to any name for example template.svg, input\_text\_filename to name witch you define in previous section).

Example: “generate\_receipt template.svg receipt.txt”

1. Press enter and your receipt template is now generated in output file.



## Preview of template

To preview receipt you can open output svg file in Internet Explorer or similar svg capable viewer.



You can see in preview that texts are aligned as you defined. Tags are not changed to final values of course. Right corner is not visible in IE, but it is immediately after right aligned text so it’s after end of logged\_user\_name.

Layout of this preview is nearly same as the final receipt printed from SPM. Small differences like a little bit different line width may occur.

If you use Inkscape for viewing you will get this preview:



NOTE: Last line “Printed by Y Soft SafeQ©” is automatically appended by generating utility.

# 4 Uploading receipt template

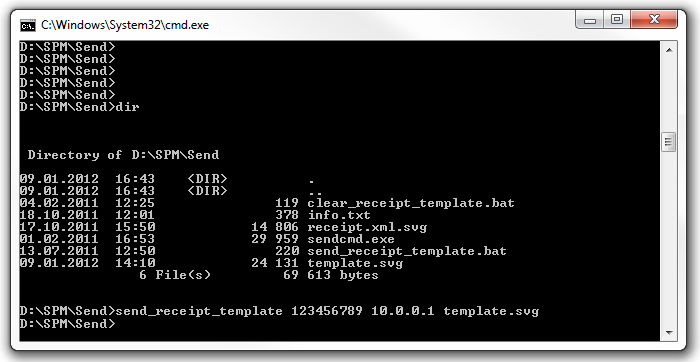
For uploading receipt template you will need sending utility (folder Send).

1. Copy prepared receipt template from previous section to folder with sending utility
2. Open command line window in same folder
3. To upload receipt template type

“send\_receipt\_template sm\_pin\_level2 ip\_address output\_filename”

(replace sm\_pin\_level2 with service menu level 2 pin, ip\_address with IP address of SPM – could be obtained in SPM service menu - informations, output\_filename with your template name for example template.svg).

Example: “send\_receipt\_template 123456789 10.0.0.1 template.svg”



1. Test print of receipt (follow instructions in YSoft SPM - Installation, Operation and Maintenance Guide)

**Example of final receipt:**

This is scan of really printed receipt generated from service menu printer test so values are changed to some defaults like username and recharged credit which is negative.



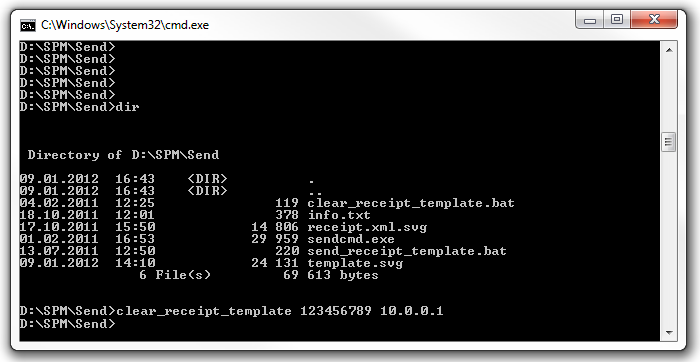
# 5 Clearing receipt template

For clearing current receipt template you will need sending utility (folder Send).

1. Open command line window in same folder
2. To clear receipt template type

“clear\_receipt\_template sm\_pin\_level2 ip\_address”

(replace sm\_pin\_level2 with service menu level 2 pin, ip\_address with IP address of SPM – could be obtained in service menu i- informations).



1. Test print of receipt (follow instructions in YSoft SPM - Installation, Operation and Maintenance Guide)

After successful receipt removing SPM will print only default demo receipt rendered according to default YSOFT demo template.

# 6 Template format in details

## Allowed embeddable text tags

Currently SPM support these text tags:

|  |  |  |  |
| --- | --- | --- | --- |
| Tag | Mean | Example | Notes |
| charge\_time\_stamp | charge timestamp in unix format (seconds) | **1325796352** | could be used as unique identifier |
| charge\_time\_dmyy\_hms | recharge time in indicated format | 10.01.2012 09:31:30 |  |
| charge\_time\_yymd\_hms | recharge time in indicated format | 2012/01/10 09:31:30 |  |
| charge\_time\_dmyy | recharge time in indicated format | 10.01.2012 |  |
| charge\_time\_yymd | recharge time in indicated format | 2012/01/10 |  |
| charge\_time\_hms | recharge time in indicated format | 09:31:30 |  |
| charge\_value\_sign | charged value and currency sign | 12 $ |  |
| charge\_value | charged value | 12 |  |
| charge\_sign | currency sign | $ |  |
| logged\_user\_name\_login | logged user name and login name | John Doe (JDoe) |  |
| logged\_user\_name | logged user name | John Doe |  |
| logged\_user\_login | logged user login name | JDoe |  |
| logged\_user\_oldcredit | logged user credit before recharging | 123 $ |  |
| cfg\_hostname | payment machine name | PM001 | could be used as a part of unique identifier |

When SPM is connected to YSoft Payment System there are allowed also these new tags:

|  |  |  |  |
| --- | --- | --- | --- |
| Tag | Mean | Example | Notes |
| logged\_user\_currentcredit | logged user credit after recharging | 234 $ |  |
| receipt\_id | unique identifier | 12345 |  |
| servercfg\_receipt\_text | Fixed text of receipt configurable on server | Any text |  |

## SVG format

This section contains detailed description of used format of receipt template and is useful only for fine tuning of receipt. So you can kindly skip this section.

SVG file format is based on XML language. So it has elements with opening and closing tags (example: <svg>content</svg>) or elements with tags that end immediately (example: <image …… />).

Whole document has this structure:

<?xml version="1.0" encoding="UTF-8"?>

<svg

xmlns:svg="http://www.w3.org/2000/svg"

xmlns="http://www.w3.org/2000/svg"

xmlns:xlink="http://www.w3.org/1999/xlink"

version="1.1"

width="576"

height="510"

id="svg1"

font-family="Arial Unicode">

HERE INSERT CONTENT

</svg>

Note in red is not part of document.

Inserting one element into others is allowed only by way defined in this document.

Only one interesting part is value of attribute “height” witch defines maximal height of receipt in pixels and has to respect receipt content.

Maximal allowed value is 2200.

In content part (opened svg element) there are only these elements allowed:

|  |  |
| --- | --- |
| image | Allows to insert embedded image (base64 encoded bitmap with 1bit/pixel, uncompressed encoding) |
| text | Allows to insert static texts, text tags, and inline tags |
| switch | Allows to select from multiple texts according to current language |

### Svg element image

Svg element “image” allows you to embed bitmap directly to svg file. SPM supports only black & white bitmaps (1bit/pixel), in uncompressed encoding.

Element format:

<image xlink:href="data:image/bmp;base64,HERE ARE BASE64 ENCODED IMAGE DATA "

x="6" y="0" width="560" height="220" />

Note in red is in real template whole bitmap encoded into base64.

Attributes “x” and “y” defines position of top left corner of image. Values are in pixels from left top corner of whole receipt.

Attributes width and height defines image size in pixels.

### Svg element text

Svg element “text” allow you to insert static texts, text tags (values will be filled after recharging) and inline tags.

No text wrapping is done so all text must be only one line that fits on receipt.

Element format:

<text id="text0001" x="288" y="250" text-anchor="middle" font-size="30px">SOME TEXT</text>

Attribute id:

* Unique identification of element
* Inline tags starts with “inctag”
* Tags starts with “tag”
* Normal static text starts with text but it’s not necessary
* Rest of string could be anything

Text-anchor:

* Allowed are values “start”, “middle” and “end”
* Defines where located anchor of text is
* Anchor is point on baseline of text
* According to setting points to beginning, center or end of text

Attributes “x” and “y” defines position of anchor. Values are in pixels from left top corner of whole receipt. Attribute “x” defines horizontal value; attribute “y” defines vertical value. Both could not be negative.

Attribute “font-size” defines size of text; only allowed unit is pixels (px).

### Svg element switch

Svg element “switch” allows you to insert texts according to currently selected language by user. So you are allowed to include some part of receipt text only for foreign users etc.

Element format:

<switch>

<text id="text0020" x="288" y="695" text-anchor="middle" font-size="30px"

systemLanguage="cs-CZ">Only in czech</text>

<text id="text0021" x="288" y="695" text-anchor="middle" font-size="30px"

systemLanguage="en-US">Only in english</text>

<text id="text0022" x="288" y="695" text-anchor="middle"

font-size="30px">Default one</text>

</switch>

Only “text” elements are allowed.

Text elements has one more attribute “systemLanguage” which defines required language to use.

Texts are handled from first and that one witch first has language as currently selected or has no language settings is used and rendered to receipt. Other texts are not used so always only one of text in switch element is shown.

## Inline tags

Inline tags are similar to normal embeddable text tags, but allow you to have mixture of static text and tag together.

For example normal text tag “charge\_value\_sign” results in text like “12 $”. If you like to have result text “=12 $” you needs to prepend “=” char but to unknown location.

This situation solves inline tag.

Basically prepare template with normal text tag and then make these modifications to final receipt template:

1. Current tag or text has attribute like id=”text0001” or id=”tag0001”. Number may be different. To allow inline tags in this text part replace part of value “tag” with “inctag”.
2. Modify content of svg tag text (content between <text> and </text>). Insert static text as you need and add BEFORE and AFTER text tag two chars “%”

Example:

We have part of receipt template:

<text id="tag0009" x="575" y="490" text-anchor="end"

font-size="30px">charge\_value\_sign</text>

Witch results after one recharging to “12 $” and we wants to have result string “=12 $”, then modify template to

<text id="inctag0009" x="575" y="490" text-anchor="end"

font-size="30px">=%%charge\_value\_sign%%</text>

Using multiple text tags in one text tag is allowed (<text…> static text %%tag1%% other static text %%tag2%%</text>)

# 7 Tuning receipt template

All following sections are about detailed tuning of receipt template. Generating utility is made as simple as possible to allow very simple usage so it could not handle detailed tuning of template. According to this all following changes have to be made by direct editing of receipt template by user.

For template editing use same editor as for text source file so use windows notepad or similar program.

## Text size and position

As described before receipt generating utility use only fixed font size and fixed line height.

If you like to have some text with different size or on another position you could edit final receipt by yourself.

To move texts edit template, search for required text and change value or “x” or “y” attribute. Don’t forget that values define where located anchor of rendered text is so respect alignment also.

To define text with another size simply edit “font-size” attribute.

Anchor of text is located on baseline of it so larger text takes more space above. To prevent text overlapping reserve enough space above text by adding empty line before text in text source or by moving anchor position down.

## Embedding image with different size

Prepared generating utility only accepts one image witch is automatically resized to default size.

To use more images or image with different size some special handling is needed.

When you need to include more images you could simply generate more templates and copy image elements from one template to another one with editing their position.

To embed image with different size follow these steps:

* Prepare image, extend/resize image to final required size
* On command line instead of running “generate\_receipt.bat” run “generate\_receipt\_noresize.bat”

By this command you skip resizing of image; only format conversion is called. Resulting receipt will include your image with original size. Image definitions in receipt template will also include correct values of image size automatically.

## Inline tags

Inline tags are similar to normal embeddable text tags, but allow you to have mixture of static text and tags together on single line.

To use them prepare you receipt with normal static text and after generating template simply edit resulting receipt template according to specifications of inline tags – see section “Inline tags” on page 19.

## Language switch

Language switch allows you to define different texts for different language currently selected. Only one text from this text group is rendered on final receipt so these texts may overlap.

This feature is usable for some notes to foreign students in universities. When the Payment station will be using Multilanguage community, complete localization of receipt is also possible.

To implement language switch simply prepare your receipt template in one language and then edit every text with you like according to specification in section ”Svg element switch” on page 18.

# 8 Relationship with other documents

|  |  |  |  |
| --- | --- | --- | --- |
| Nr. | Document | Version | Author |
| 1. | YSoft SPM - Installation, Operation and Maintenance Guide [EN] | 1.0 | PST |
| 2. |  |  |  |
| 3. |  |  |  |

# 9 Version history

Document author and (owner): Pavel Staněk

*Update Name, Surname and email address. Owner means who is responsible for keeping the document updated.*

**This version list is intended only for basic orientation. The list of changes is not guaranteed to be complete.**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Change / Document status | Author |
| 0.4 | 6.2.2012 | First draft | PST |
| 1.1 | 10.2.2014 | Add modifications related to Payment system | PST |
|  |  |  |  |