SambaPOS Promotions

# About this document

Promotion is something that we use to attract customers and improve loyalty. There are lots of promotion offers that are common on most businesses. However a good promotion system should be designed carefully and should contain some unique ideas. Having a unique idea is of course not enough. You should be able to implement your idea on your POS system to be able to bring it to life. Can you implement your ideas on your POS system? If you are using SambaPOS you can. It gives you endless possibilities with the powerful Rule infrastructure. On this documentation you’ll see how to configure SambaPOS for some common promotion cases but besides that I’ll also try to give some details about the Rule system. Once you get how Rule system works you’ll be able to customize SambaPOS workflows for your needs. You’ll really love it. So let’s start.

This documentation prepared with V3.0.26. If you are using previous releases you should upgrade to try samples.

Document Version: 1.0

Index

[About this document 1](#_Toc365822250)

[Welcome Gift for new Customers 3](#_Toc365822251)

[How it works? 3](#_Toc365822252)

[Configuration 5](#_Toc365822253)

[Configuration Overview: 5](#_Toc365822254)

[New Customer Gift Pack Product 6](#_Toc365822255)

[Add new customer gift pack action 6](#_Toc365822256)

[Disable New Customer action 7](#_Toc365822257)

[Enable Ticket Promotion Action 7](#_Toc365822258)

[Disable Ticket Promotion Action 7](#_Toc365822259)

[Welcome Pack Automation Command 8](#_Toc365822260)

[Enable ticket promotion for new customers rule. 9](#_Toc365822261)

[Add welcome pack when promotion button clicked rule. 10](#_Toc365822262)

[Additional properties for Update Order and Update Order Gift State actions. 10](#_Toc365822263)

[Buy 1 Beverage, Get %50 discount for second 11](#_Toc365822264)

[How it works? 11](#_Toc365822265)

[Configuration 14](#_Toc365822266)

[Configuration Overview 14](#_Toc365822267)

[Cut price %50 Action 15](#_Toc365822268)

[Enable ticket %50 promotion Action 15](#_Toc365822269)

[Disable Ticket Promotion Action 15](#_Toc365822270)

[Update order service state Action 16](#_Toc365822271)

[Update order promotion state Action 16](#_Toc365822272)

[Add new order Action 17](#_Toc365822273)

[Load Last Order Action 17](#_Toc365822274)

[Second item %50 Automation Command 18](#_Toc365822275)

[Serve Later Automation Command 18](#_Toc365822276)

[Serve Later Automation Command Mappings 19](#_Toc365822277)

[Serve Now Automation Command 19](#_Toc365822278)

[Enable %50 Promotion Rule 20](#_Toc365822279)

[Serve Later button clicked Rule 20](#_Toc365822280)

[Serve Now button clicked Rule 21](#_Toc365822281)

[Execute Second item %50 Rule 21](#_Toc365822282)

# Welcome Gift for new Customers

This idea is about giving some small gifts or offer some special promotions for new customers. This setup is also useful for seasonal promotions such as Christmas gifts that given once for each customer.

## How it works?

|  |  |
| --- | --- |
|  | Creating new customer Create a new customer and click **Select Customer** to create a new ticket for that customer. |
|  | Selecting customer When customer selected, ticket state appears as **Promotion: New Customer** and **Welcome Pack** button appears. Welcome Pack button not enabled since no orders added yet.  Let’s add some orders. |
|  | Adding orders If you click **Welcome Pack** button it will add a gift inside ticket and disable gifts for that customer. You can cancel it to apply on further tickets but once applied you won’t be able to use it for this customer anymore.  On next step we’ll see what happens when we click on Welcome Pack button. |
|  | Adding welcome pack Welcome Pack product added as a gift, promotion button removed and new customer gift state also removed from ticket. |
|  | Tickets for that customer On new tickets for John Doe we won’t see “Welcome Pack” button. |
|  | Tickets for other new customers Welcome pack button appears for new customers who didn’t received new customer promotion before. |

## Configuration

### Configuration Overview:

|  |  |  |
| --- | --- | --- |
| Item | Name | Description |
| Product | New Customer Gift Pack | Product that we’ll add into the ticket. |
| Action | Add new customer gift pack | Adds New Customer Gift Pack into the ticket. |
| Action | Disable new customer | Updates customer state as “new customer= false” so we can determine customer received gift pack before. |
| Action | Enable Ticket Promotion | Updates ticket’s promotion state to “New Customer” so “Welcome Pack” button becomes visible. |
| Action | Disable Ticket Promotion | Updates ticket promotion state to empty value so “Welcome Pack” button disappears. |
| Automation command | Welcome Pack | The button that appears next to the ticket when ticket promotion state is “New Customer” |
| Rule | Enable ticket promotion for new customers | This rule checks customer state and if it detects a new customer it updates ticket’s promotion state to “New Customer”. |
| Rule | Add welcome pack when promotion button clicked | It adds “New Customer Gift Pack” product into the ticket, it changes that order as a gift, disables ticket’s promotion state and disables customer’s “new customer” state. |

|  |  |
| --- | --- |
|  | New Customer Gift Pack Product First of all we’ll create a dummy product that represent your gifts. It might be a sticker, a key ring or any other special gift. When we click on “Welcome Pack” we’ll add this product into the ticket. You can configure this product with zero price since this is a free product. |
|  | **Add new customer gift pack action**  |  | | --- | | Add Order Action | | Menu Item Name: | | New Customer Gift Pack | | Quantity: | | 1 |   This action adds New Customer Gift Pack product to the ticket. You can define a portion and order state name. |
|  | **Disable New Customer action**  |  | | --- | | Update Entity State Action | | Entity Type Name: | | Customers | | Entity State Name: | | New Customer | | Entity State | | False |   This action updates “New Customer” state of the Entity to false so we can understand this customer received a new customer promotion before. You can name and update any state of customers without pre-configuration. |
|  | **Enable Ticket Promotion Action**  |  | | --- | | Update Ticket State Action | | State Name: | | Promotion | | State: | | New Customer |   This action is called when we determine customer is a new customer from customer state. We change ticket’s promotion state to **New Customer** so Automation Buttons that mapped to Promotion state will appear. |
|  | **Disable Ticket Promotion Action**  |  | | --- | | Update Ticket State Action | | State Name: | | Promotion |   When we left State value empty it clears state value from ticket. As this state cleared Automation Buttons mapped to this state also removes. You can map different **Automation Buttons** to ticket states. For example you can display **Welcome Pack** button when Promotion state is “New customer” or “Free Hot Drink” button when promotion state is “VIP Customer”. |

**State tracking is one of most important features of SambaPOS V3. State is a small piece of information we attach to Tickets, Entities or Order Lines. When we are thinking about unpaid Tickets, new Customers, voided Orders, etc… most probably what we need is tracking a state. You can configure unlimited number of states for your needs. SambaPOS already uses states to track features such as table occupation, ticket statuses or void orders. You can customize them or add additional states for your needs. For example SambaPOS automatically closes a ticket once paid. If your customers still occupies tables after paying tickets and you want to track that issue you can add additional states. Remember a state change generally triggers a printout or a visual indication. There are rules to track state changes and execute needed actions.**

**Now let’s continue by configuring required automation command.**

|  |  |
| --- | --- |
|  | Welcome Pack Automation Command This automation command displays a “Welcome Pack” button next to the ticket. Button Header contains \r to define where to wrap button text. Color and Font Size values are visual options. When we click on an automation command button it triggers “Automation Command Executed” rule. If you enter values here it asks choosing a value before executing the rule. It is useful to configure things like “Gift Reasons”.  This button should be visible when a new customer selected. We’ll configure it on “Mappings Page |
|  | Terminal, User Role, Department and Ticket Type used to map the button to specific Departments or Users. You can map this button to Administrators or just Delivery Department. \* means all.  This button will be displayed when ticket state is **New Customer**. Normally typing state name is not required when state value is unique. If you need you can type full state as **Promotion=New Customer**.  This button will be enabled for **New Orders** and **Unpaid** ticket states. That means when ticket is Paid or Locked we won’t be able to click this button. |

**You can map it to multiple states by separating values with comma. Leaving Visible States value empty means it will be “always visible” even there is no state value and typing \* means it will be visible for all states. You can change where this button is displayed by changing Visibility value.**

**On next step we’ll configure rules to wire things together.**

**A rule executes selected actions in order on certain SambaPOS events. After choosing the event you can add constraints to configure conditions that controls rule execution. Finally you can click Select Actions link to add actions you configured. On mappings page you can map this rule to certain user roles, departments, ticket types or terminals. For example if you map a rule to a department that rule only works for that department.**

|  |  |
| --- | --- |
|  | Enable ticket promotion for new customers rule. This rule calls **Enable ticket promotion** action to change ticket’s Promotion state to **New Customers**  Event Name is **Ticket Entity Changed** so this rule executes when we select an entity to the Ticket.  This rule contains two constraints to check if selected entity is a Customer and the Customer’s **New Customer** state is empty. If empty we determine customer is new.  We’ll add default mapping to all rules by clicking add mapping link on mappings page. |

**As V.3.0.26 release Rule constraints feature improved for easier configuration. You can click “Add Custom Constraint” link to add a constraint. You can add multiple constraints to a single Rule Property. For example if you have separate “Delivery Customers” entity type you can add additional constraint row to check if selected entity is “Delivery Customer”.**

**You can also use Printer Template Tags in rules to read values. For example we can use {ENTITY STATE:New Customer} printer template tag to print customer’s New Customer state. We can also use this tag to read that state in rule constraints.**

|  |  |
| --- | --- |
|  | Add welcome pack when promotion button clicked rule. This rule triggers when Welcome Pack button clicked and executes actions we configured. It also executes additional actions to update the order.   1. Adds New Customer Gift Pack product into the ticket. 2. Updates order to exclude its price from calculation. 3. Marks order as gift. 4. Disables ticket’s promotion state. 5. Disables customer’s new customer state. |
|  | Additional properties for Update Order and Update Order Gift State actions. These actions are used to mark the promotion ticket line as gift. Decrease means it will decrease inventory. Calculate Price sets if it will be included in ticket sum calculation.  Second action changes orders Gift state. You can consider typing another value such as “Promotion” here to differ it from regular gifts. Additionally when you update it as “Promotion” “Cancel Gift” command won’t be visible when we click on that order. You can fine tune to match it to your business ideas. |

**Configuration for “New Customer Gift Promotion” ends here. If you offer multiple promotion choices for new customers you can improve “Automation Command” to ask values or you can configure multiple automation commands for different types of promotions. We’ll see some examples on further topics.**

# **Buy 1 Beverage, Get %50 discount for second**

When any customer buys a beverage we’ll offer %50 discount for second beverage. We’ll assume beverage is same here. That means if customer buys Tea we’ll offer %50 for second Tea. Promotion should applied immediately but order can be marked as “Serve Later”

## How it works?

|  |  |
| --- | --- |
|  | Creating new ticket We’ve created a new ticket, added a breakfast item and switched to Beverages Page. |
|  | Adding Coffee When we add a coffee orange **Second for %50 off** button appears on next to ticket. We’ll ask customer if she wants second coffee for %50 and when confirmed we’ll click that button. |
|  | Clicking Promotion Button When we click on **Second for %50 off** button it duplicates last order with %50 price. |
|  | Pausing service At that point we should ask customer if she wants second coffee now or later. If she wants it later we’ll click on the order.  When we click on the order, **Serve Later** button will appear. |
|  | Marking order as Serve Later Click on that button to mark the order as “Serve Later”. |
|  | Printing On kitchen print, orders marked as **Serve Later** won’t appear. |
|  | Start Service When customer wants her coffee open ticket and click on the order marked as **Serve Later.** When you click on that order green **Serve Now** button should appear. Click that button to serve. Order should marked as **Serve Now** and its state should be new to trigger a printout. |
|  | Coffee Printout When we close ticket Kitchen will receive this printout. |

When an order marked as Serve Later you can print a coupon for customer. To be able to keep tutorial short we’ll discuss this case on another documentation.

## Configuration

### Configuration Overview

|  |  |  |
| --- | --- | --- |
| Item | Name | Description |
| Action | Cut Price %50 | Decreases order price by %50 |
| Action | Enable ticket %50 promotion | Updates ticket state to switch its state to %50 Off to display related automation commands (buttons) |
| Action | Disable ticket promotion | Disables ticket promotion state |
| Action | Update order service state | Updates order state to mark it as **Serve Now** or **Serve Later** |
| Action | Update order promotion state | Updates order state to display **%50 off** promotion state. |
| Action | Add new order | Reads last order name and adds an order with that name. |
| Action | Load last order | This action loads last line of ticket into memory so actions running after that action can access that order. |
| Automation Command | Second Item %50 | Command to execute when we click on **Second item for %50 off** button. |
| Automation command | Serve Later | Command to execute when **Serve Later** button clicked. |
| Automation Command | Serve Now | Command to execute when **Serve Now** button clicked. |
| Rule | Enable %50 Promotion | This rule checks if added item is a beverage and if so it updates ticket state to **%50 Promotion** |
| Rule | Execute Second Item %50 | This rule calls required actions when **Second Item for %50 off** button clicked |
| Rule | Serve Later button clicked | Calls required actions to mark an order as **Serve Later.** |
| Rule | Serve Now button clicked | Calls required actions to mark an order as **Serve Now** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cut price %50 Action  |  | | --- | | Update Order Action | | Price | | [=TN('{PRICE}') / 2] |   This action decreases order price to half. We are using both printer template tags and expressions here. TN function is needed to change Price value to a number. We’ll read current order price with **{PRICE}** tag, convert it to a number with **TN** function and divide it by **2.** |
|  | Enable ticket %50 promotion Action  |  | | --- | | Update Ticket State Action | | State Name | | Promotion | | State | | %50 Off |   It will change ticket’s promotion state to **%50 Off** so automation commands mapped to ticket state will appear as buttons. |
|  | Disable Ticket Promotion Action  |  | | --- | | Update Ticket State Action | | State Name | | Promotion | | State | |  |   This action will disable promotion state to hide mapped automation command buttons. Leaving State value as empty will disable state value. |
|  | Update order service state Action  |  | | --- | | Update Order State Action | | State Name | | Service Status | | Group Order | | 5 | | State | | [:Status] | | State Order | | 5 |   Updates order service state. This action demonstrates variable usage. [:Status] means we’ll set this value from rule. Group Order values needed to sort state values display under order line |
|  | Update order promotion state Action  |  | | --- | | Update Order State Action | | State Name | | Promotion | | Group Order | | 3 | | State | | %50 Off | | State Order | | 3 |   It updates order’s promotion state to %50 off so we can determine this order is a promotion. |
|  | Add new order Action  |  | | --- | | Add Order Action | | Menu Item Name | | {NAME} | | Quantity | | 1 |   This action reads current order product name and adds a new order with that product name. So when you execute this action when an order is selected on ticket it duplicates that order. |
|  | Load Last Order Action Action type of this action is **Load Last Order** with no parameters. When we click on promotion button there is no selected order so it selects last order of the ticket. Order related actions running after this action can access that order. |

We can make simple calculations by using Expressions. Typing a value as [=10+3] means 13 and [=10/2] means 5. Instead of typing 10 we can read values by using printer template tags. So since {PRICE} tag replaces with order price [={PRICE}/2] will be equal to order price divided by 2. Printer template tags designed to generate formatted numbers with thousand separators so if {PRICE} generates a value formatted with a thousand separator calculation will fail because this is not a valid number. So to be able to ensure {PRICE} tag always generates a valid number we’ll use TN (to number) function. So calling it as TN(‘{PRICE}’) will always produce a valid number. Our final expression becomes **[=TN(‘{PRICE}’)/2]**. Using calculations is very handy when we need to decrease prices or make calculations by reading different values from order or ticket. For most cases you won’t need using expressions but when you need it will be ready for you for implementing advanced ideas.

Now we’ll continue configuration by adding needed automation commands and rules.

|  |  |
| --- | --- |
|  | Second item %50 Automation Command This automation command displays promotion button. We’ll configure button properties on this page.  We want to show it when ticket’s promotion state is %50 off so we’ll configure it on Mappings page.  **Hint:** Ticket’s promotion state becomes “%50 Off” when we add a beverage so we’ll see this button after adding a beverage into ticket. |
|  | Visible States value is %50 Off. So we’ll see this button as soon as ticket state becomes %50 Off. We can map this button to a Specific user role or department. So only managers can use this button or it will be available to Fast Food department. You can read previous promotion case for more info about this page. |
|  | Serve Later Automation Command This automation command used to mark an order to Serve Later. This button will be visible when we select a new order.  After configuring visual properties we’ll switch to “Mappings” page for map this button to orders. |
|  | Serve Later Automation Command Mappings It will be visible for New orders so the visible state will be **New**. Since this command should work when we select an order we’ll change Visibility Property of that button as **Order Line.** That means we can use this button when we select an order line. |
|  | Serve Now Automation Command I’ve used different colors for these buttons. You can change them for your taste.  This button should appear when we click on an order marked as “Serve Later”. We’ll configure it on Mappings Page. |
|  | Visibility States of this button will be “Serve Later” since it will appear if current order state is “Serve Later”. Again Visibility of this button should be Order Line because this command will work for orders. |
|  | Enable %50 Promotion Rule This rule handles **Order Added to Ticket** event and if added item’s group code equals to Beverages it update **Promotion** state of ticket to **%50 Off**.  We’ll add default mapping to the rule from mappings page and save. |
|  | Serve Later button clicked Rule When **Serve Later** automation command button clicked this rule executes and updates selected Order’s service state to **Serve Later**. This rule executes every time an automation command executed. By adding a custom constraint as shown in the screen shot we can determine **Serve Later** button clicked.  We’ll add and expand **Update Order Service State** action and type Status value there. Remember how we used **[:Status]** variable while configuring this action to update state value from rule. |
|  | Serve Now button clicked Rule This rule does something similar to Serve Later button handling rule. It updates order’s servicing state to **Serve Now.** Additionally it changes order status to **New** so kitchen printing triggers to print that line. Remember we do not print lines marked as **Serve Later**.  If you are using Localized versions of SambaPOS **New** value should be localized value. Please check Kitchen Print template to see which value is used for marking orders as New. |
|  | Execute Second item %50 Rule This rule executes when we click on **Second Item %50** button. We determine which button is clicked by adding a custom constraint for Automation Command Name.  It calls all configured actions in that order.   1. It loads last order line so next rules will read values from that order. 2. Adds a new order same as last order. 3. Decreases its price by %50 4. Changes order’s promotion state to “%50 Off” 5. Disables ticket’s promotion state. 6. Refreshes ticket.   **Hint**: If you do not have Refresh Ticket action create a new “Display Ticket” action with Ticket ID parameter is 0. |

This is the end of second promotion case.