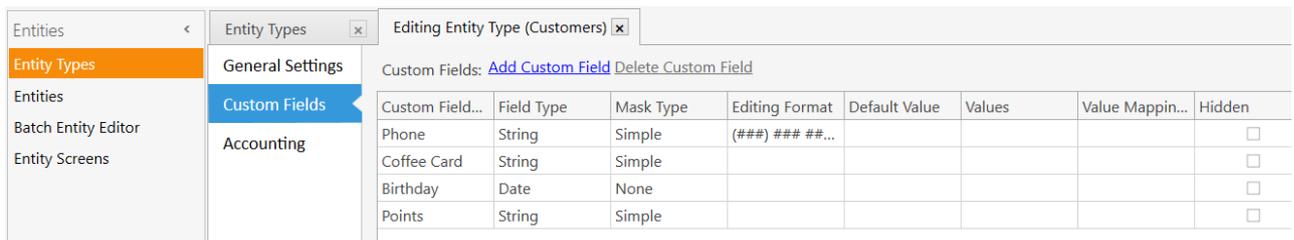


Simple Coffee Card Reward System

I wanted to implement a simple points reward system for our coffee customers. 1 point per coffee and when they had accumulated 6 points the next one was free. As there was nothing that matched my needs I came up with the following. Special thanks to Nizamettin (Nizam) for the inspiration and Joe (JTRTech), Jesse (Jesse), Rick (RickH), Bob (Bob_Be) and Arian A (ATechAsis) for all their help in getting this running!

So let's get started!

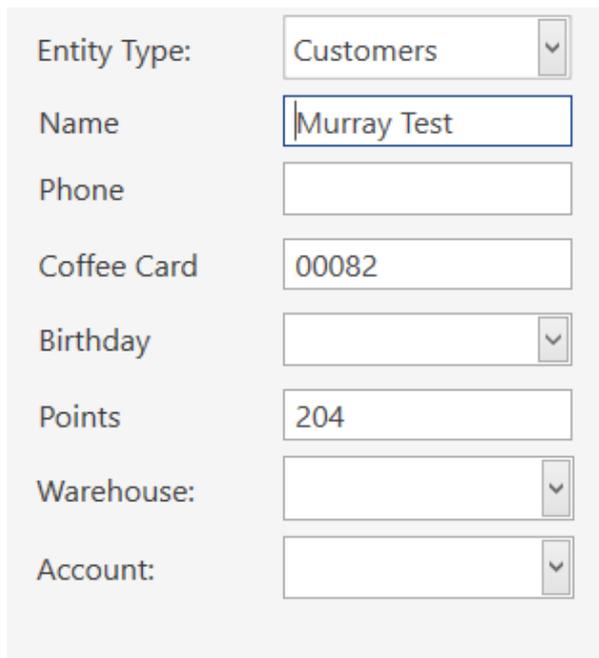
I added two custom fields to the Customer Entity; Coffee Card and Points:



The screenshot shows the 'Editing Entity Type (Customers)' interface. On the left, there is a sidebar with 'Entity Types' selected. The main area is divided into 'General Settings', 'Custom Fields', and 'Accounting'. The 'Custom Fields' section is active, displaying a table of custom fields for the 'Customers' entity type.

Custom Field...	Field Type	Mask Type	Editing Format	Default Value	Values	Value Mappin...	Hidden
Phone	String	Simple	(###) ### ##...				<input type="checkbox"/>
Coffee Card	String	Simple					<input type="checkbox"/>
Birthday	Date	None					<input type="checkbox"/>
Points	String	Simple					<input type="checkbox"/>

When I add a new Customer I get this:



The screenshot shows the form for creating a new Customer. The 'Entity Type' is set to 'Customers'. The 'Name' field contains 'Murray Test'. The 'Phone' field is empty. The 'Coffee Card' field contains '00082'. The 'Birthday' field is empty. The 'Points' field contains '204'. The 'Warehouse' and 'Account' fields are empty.

Entity Type:	Customers
Name	Murray Test
Phone	
Coffee Card	00082
Birthday	
Points	204
Warehouse:	
Account:	

I added the Coffee Card manually here but later on that number can be scanned in at the Customer Creation stage. Points will normally start at 0 but I started with a large number for testing purposes.

I use an RFID tag and reader for customer identification so the first thing is to get the reader info into the system:

Here are the Actions:

Action Name:	<input type="text" value="CC-Search Coffee Card Entity"/>
Action Type:	<input type="text" value="Change Ticket Entity"/>
Parameters:	
Can Create Ticket:	<input type="text" value="True"/>
Can Create Entity:	<input type="text"/>
Entity Type Name:	<input type="text" value="Customers"/>
Entity Name:	<input type="text"/>
Entity Search Value:	<input type="text" value="[:Search Value]"/>
Update Null Entity:	<input type="text"/>
Entity Screen Name:	<input type="text"/>
Entity Screen Search Value:	<input type="text"/>

And:

Action Name:	<input type="text" value="CC-Refresh Ticket"/>
Action Type:	<input type="text" value="Display Ticket"/>
Parameters:	
Ticket Id:	<input type="text"/>

And the Rule:

Rule Name:

Event Name:

Custom Constraint List

Execute rule if

[Add Custom Constraint](#)

[Remove](#)

Actions: [Select Actions](#)

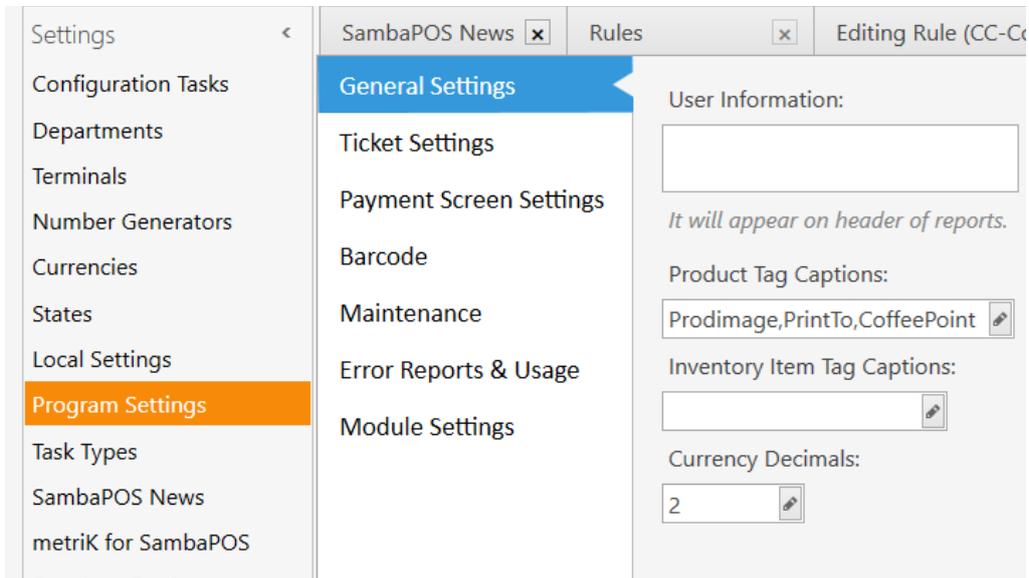
CC-Search Coffee Card Entity ▲
Constraint: <input type="text"/>
Search Value <input type="text" value="[:NumberpadValue]"/>

CC-Refresh Ticket ▲
Constraint: <input type="text"/>

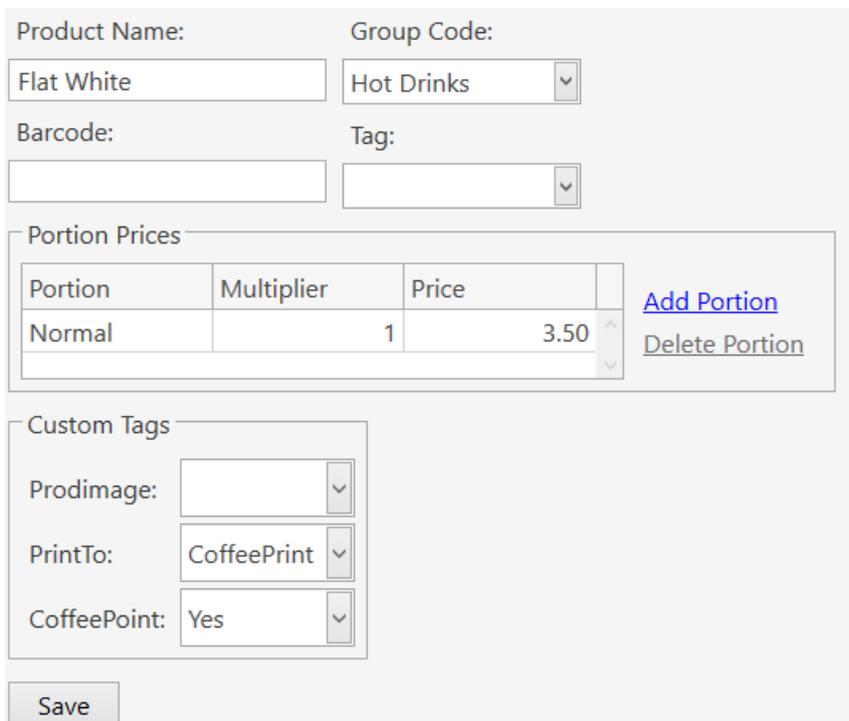
The 00 in the Numberpad Value can be anything, it just that all my RFID tags start with 00

Then I added a Custom Tag to the system to facilitate the addition and subtraction of the points.

Under Settings, Program Settings, General Settings there is a field Product Tag Captions: Just add whatever Tag/s you want to use separated by commas.



Now when you add a product and it earns a coffee point, just select (or type in) Yes into the CoffeePoint Tag field:



Create this action to update the points on the coffee card:

Action Name:
CC-Update Coffee Card Points

Action Type:
Update Entity Data

Parameters:

Entity Type Name: Customers

Entity Name:

Field Name: Points

Field Value: [:Points]

To add points:

Rule Name:
CC-Add Coffee Point

Event Name:
Order Added to Ticket

Custom Constraint List

Execute rule if Matches

[Add Custom Constraint](#)

{ITEM TAG:CoffeePoint} Equals Yes [Remove](#)

Actions: [Select Actions](#)

CC-Update Coffee Card Points

Constraint:

Points [=TN('{ENTITY DATA:Customers:Points}')+{QUANTITY}]

[=TN('{ENTITY DATA:Customers:Points}')+{QUANTITY}]

(I found that copying and pasting sometimes messed up the format so I had to type the apostrophes manually, YMMV)

To subtract points (if a coffee is cancelled at the order stage):

Rule Name:
CC-Subtract Coffee Point

Event Name:
Order Cancelled

Custom Constraint List

Execute rule if Matches

[Add Custom Constraint](#)

{ITEM TAG:CoffeePoint} Equals Yes [Remove](#)

Actions: [Select Actions](#)

CC-Update Coffee Card Points

Constraint:

Points [=TN('{ENTITY DATA:Customers:Points}')-{QUANTITY}]

[=TN('{ENTITY DATA:Customers:Points}')-{QUANTITY}]

Both these Rules handle multiple order quantities for example if someone orders 4 x Flat Whites they will receive 4 points. If the change their mind cancelling the order will subtract the 4 points.

Create an Automation Command:

Name:

Category:

Button Header:

Color:

Font Size:

Confirmation:

Execute Once

Clear Selection

Toggle Values

Values:

With the mapping:

General Settings

Navigation Settings

Mappings

Mappings: [Add](#) [Delete](#)

Terminal	User Role	Department	Ticket Type	Enabled States	Visible States	Visibility
*	*	*	*	GStatus=	GStatus=	Order Line

Time for a free coffee?

Rule Name:
CC-Give Free Coffee Rule

Event Name:
Automation Command Executed

Custom Constraint List

Execute rule if Matches All

[Add Custom Constraint](#)

Automation Command Name	Equals	CC-Free Coffee	Remove
{ENTITY DATA:Customers:Points}	Greater	6	Remove
{ITEM TAG:CoffeePoint}	Equals	Yes	Remove
[='{SELECTED ORDERS}'.indexOf(',')]	Less	0	Remove
{QUANTITY}	Equals	1	Remove

Actions: [Select Actions](#)

Update Order	
Constraint:	
Decrease	True
Calculate Price	False

Update Order Gift State	
Constraint:	
Status	Free Coffee, Yay!
Value	[:CommandValue]

CC-Update Coffee Card Points	
Constraint:	
Points	-7

This way the customer can only receive 1 free coffee at a time. You can select another coffee and click the Free Coffee button and if they have enough points they can receive another one. What you can't do is select an order of '4 x Flat Whites' and make them all free by mistake (or a Burger and fries).

What if the customer doesn't have enough points?:

Create this Action:

Action Name:

Action Type:

Parameters:
Message:

Then this Rule:

Rule Name:

Event Name:

Custom Constraint List

Execute rule if

[Add Custom Constraint](#)

<input type="text" value="Automation Command Name"/>	<input type="text" value="Equals"/>	<input type="text" value="CC-Free Coffee"/>	Remove
<input type="text" value="{ENTITY DATA:Customers:Points}"/>	<input type="text" value="Less"/>	<input type="text" value="6"/>	Remove

Actions: [Select Actions](#)

CC-Free Coffee Not Enough Points Action ▲

Constraint:

That's it! Is it perfect? Probably not. But it does the job. Feel free to make any suggestions for additions/alterations to this and I'll update it.