REALDOLMEN
CRM SINGLE VIEW BUILDER

Quick Start Guide

Authors: Nico De Clercq, Patrick Van Asch
Date: 14/02/2020
Version: 2.1

Table of contents

Preface

Thank you for acquiring Realdolmen CRM Single View Builder, the tool that will improve the user experience on Microsoft Dynamics CRM. The Single View Builder is a suite of visual components that is used to enrich the graphical user interfaces with tree views, timelines, tiles, map and grid views and other.

The solution is an add-on to the latest version of Microsoft Dynamics CRM. Earlier versions are limited supported. Please take a look at our website to check the requirements.
(http://www.cemaddons.com/SVB/system-requirements)

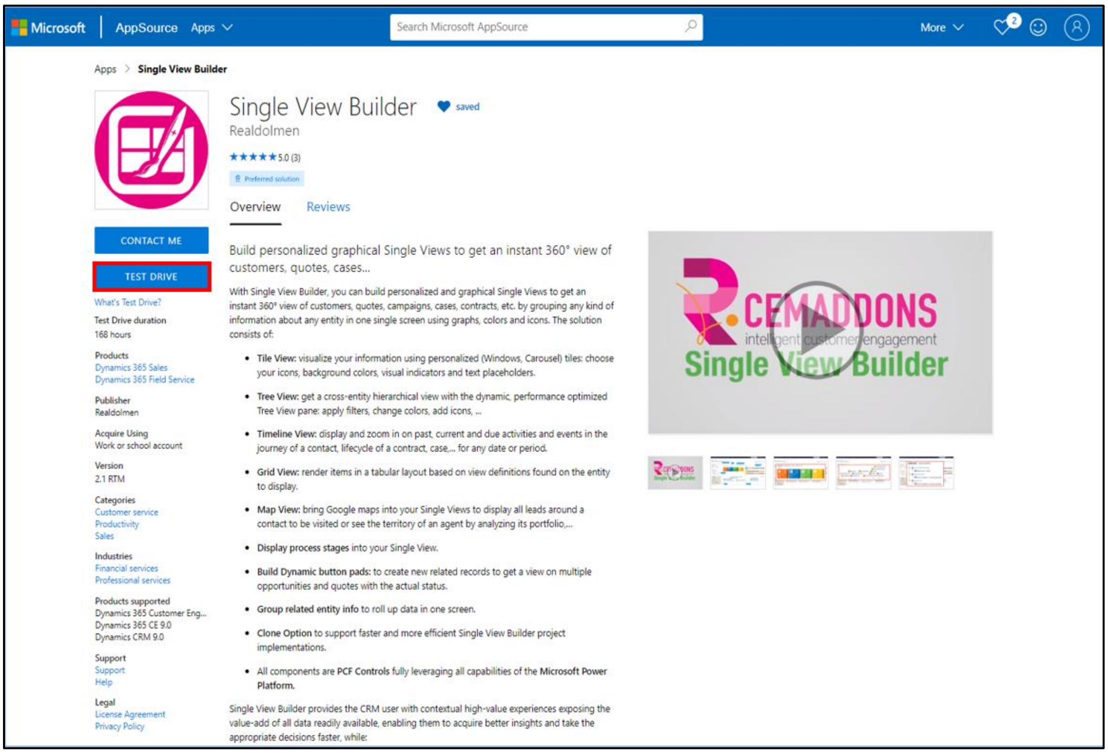
Finally Realdolmen CRM Single View Builder can be used in connected environments (online, on premises) as well as in disconnected environments (with the CRM connector for Microsoft Outlook)

Realdolmen nv
A. Vaucampslaan, 42
B-1654 Huizingen
www.realdolmen.com

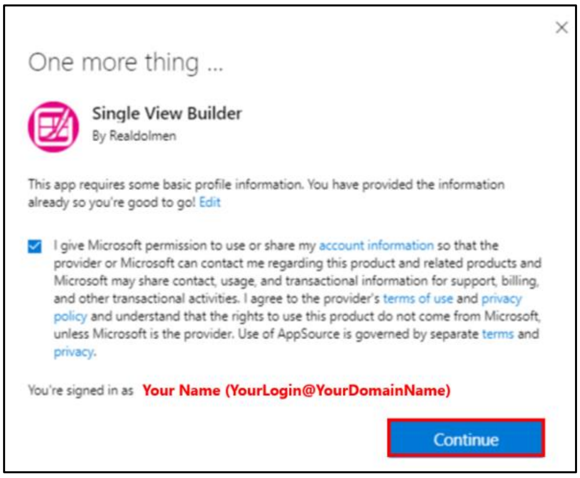
Chapter 0
Instant access to Single View Builder through AppSource Test Drive

1. Requesting your Single View Builder-Test Drive

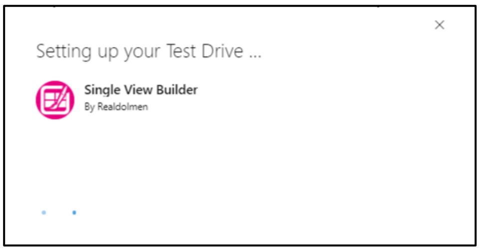
1. **Start Test Drive Process**
   From the AppSource landing page for Single View Builder, click **TEST DRIVE**.



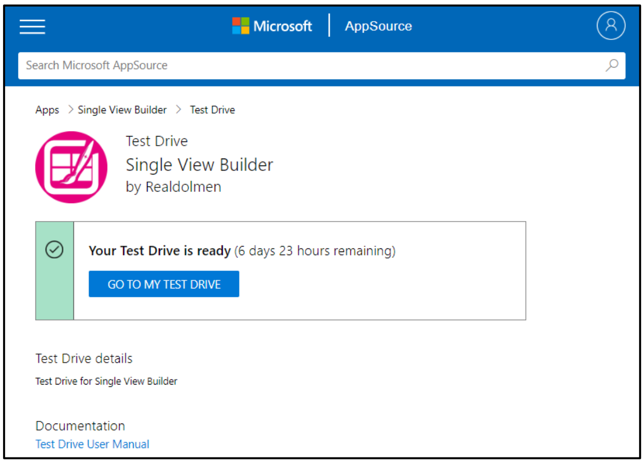
2. Mark the checkbox to share basic profile information, then click **CONTINUE** to proceed.



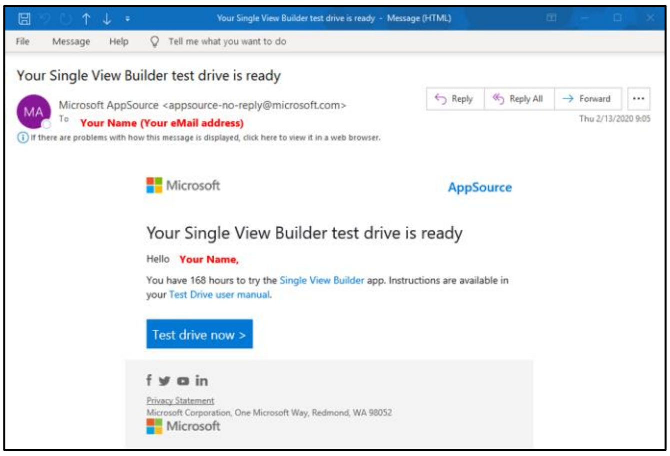
3. This setup process will take a few minutes to complete.



4. Once completed you get 7 days access to evaluate Single View Builder through the AppSource Test Drive setup.



5. You also get notified per email when your Single View Builder test drive is ready.



2. Accessing your Single View Builder-Test Drive

1. Click the AppSource link provided to access your Single View Builder-test drive

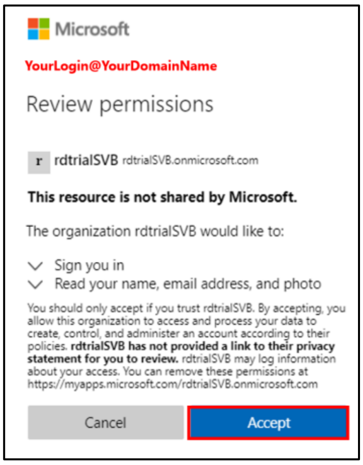
   (a) Either from the notification e-mail your received, click **Test drive now >**

   (b) Either from the result page once the test drive was setup, click **GO TO MY TEST DRIVE**

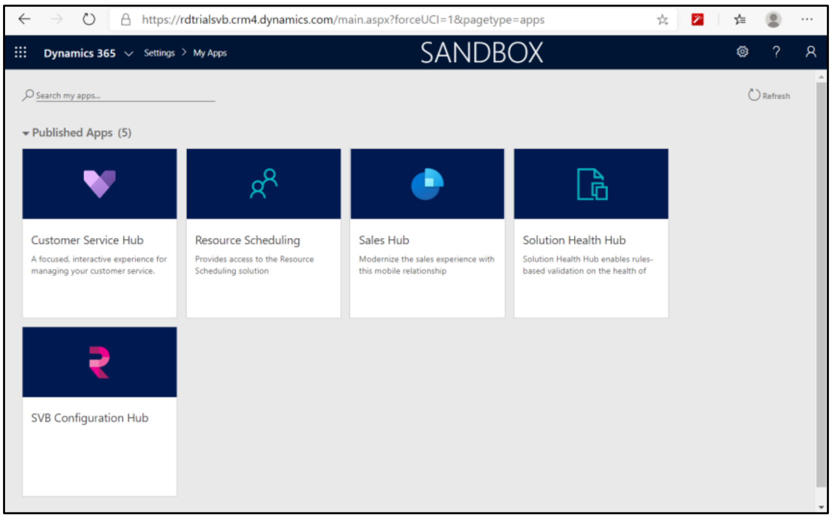
2. Configuration of your test drive user account (so you can use Single View Builder through your work account)

   (a) Test Drive enables you - customers and partners - to try out our Single View Builder solution, without signing up for a Dynamics 365 license plan nor installing any applications. You just sign into AppSource.com using your work-related Azure Active Directory (AAD) account, AppSource maps this to a test user account within the test drive-tenant setup for you to run the app in a web browser. With Test Drive, you get a better idea of what functionality and experience you get using Single View Builder.

   (b) Click **ACCEPT** to provide the necessary information and allow the sign-in setup process to continue

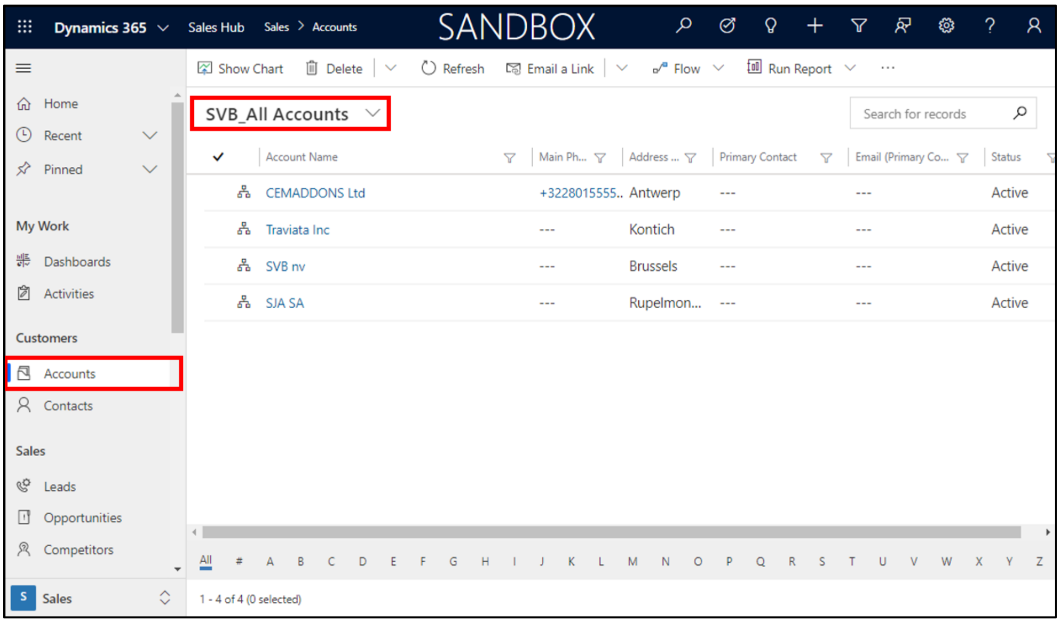


3. Your browser will open your Single View Builder-test drive session at https://rdtrialsvb.crm4.dynamics.com/default.aspx



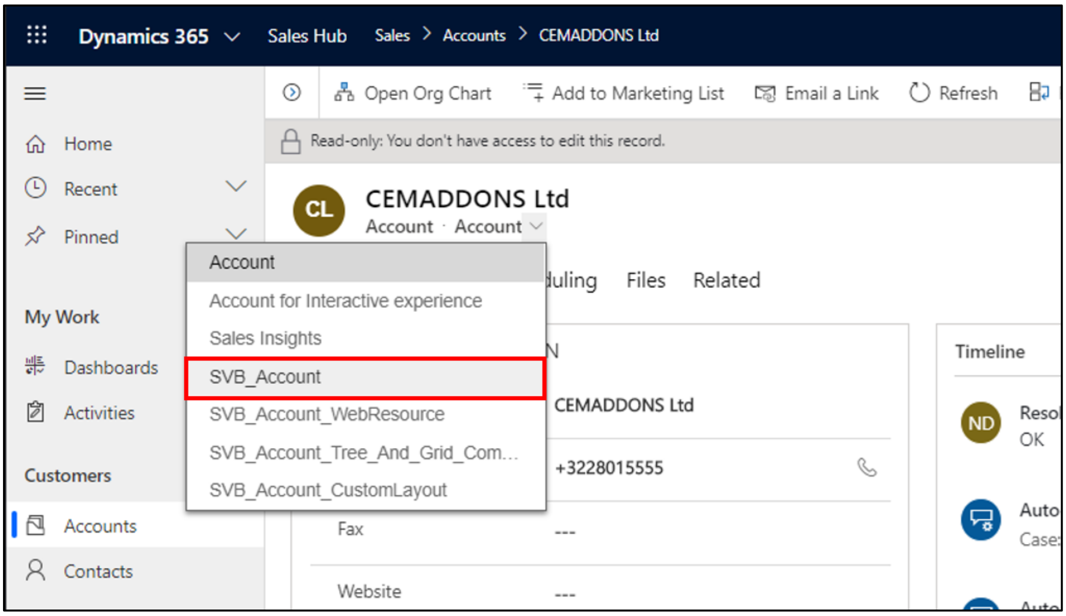
3. Showing Single View Builder Components

1. Go to **SALES HUB,** navigate to subarea **ACCOUNTS** and open an account that has cases available. Make sure you select the correct view to visualize the account, **select the SVB_All accounts view.**

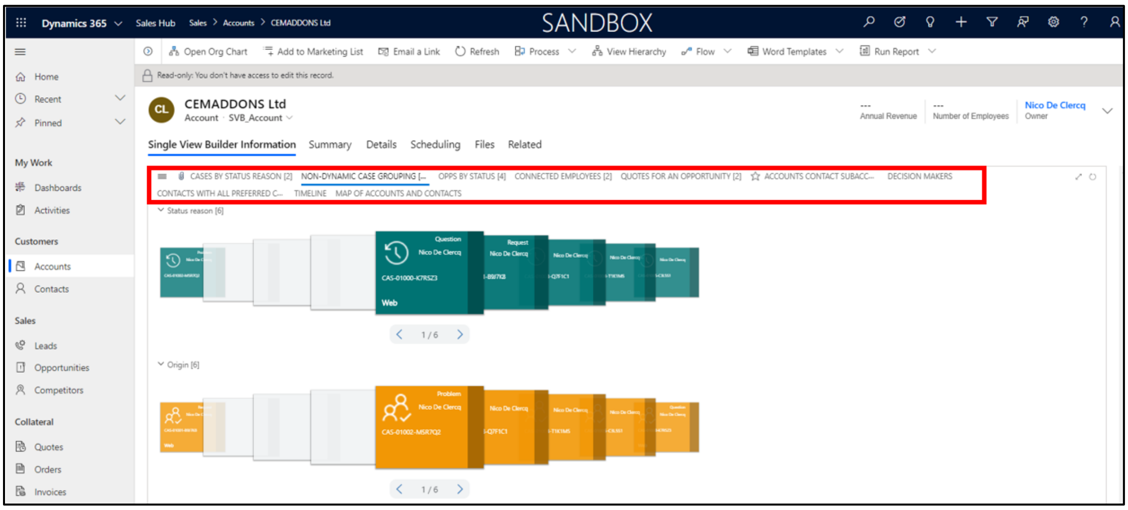


2. Select and click to open the account **CEMADDONS Ltd.**

3. Make sure you select the correct form from the form selector, which is in our case the form **SVB_Account**.



4. From here you can navigate through the different TABs to visualize the power of Single View Builder.



From left to right, on the Form **Single View Builder Information** for the selected account:

(a) The TAB **CASES BY STATUS REASON** shows use of the **TILE VIEW**, with a tile for all cases in context of a specific selected account, revealing selective and actionable small & long field information in carousel format, dynamically grouped per status.

(b) TAB **NON-DYNAMIC CASE GROUPING** shows use of the **TILE VIEW**, with a tile for all cases in context of a specific selected account revealing selective and actionable small & long field information in carousel format, without dynamic grouping but separate series ordered per specific field.

(c) TAB **OPPS BY STATUS** shows use of the **TILE VIEW**, using the Process Type visualizing all opportunities of a selected account, dynamically grouped by status.

(d) TAB **CONNECTED EMPLOYEES** shows use of the **TILE VIEW**, using the Tile Type visualizing all contacts connected to an account with the employee connection role, dynamically group using the Tile Type

(e) TAB **QUOTES FOR OPPORTUNITY** shows use of the **TILE VIEW**, using the New Tile Type visualizing all quotes of a selected account creating starting from an opportunity of that account.

(f) TAB **ACCOUNTS CONTACT SUBACCOUNTS CONTACT** shows use of the **TREE VIEW**, to visualize both contacts and subaccounts of a selected main accounts, as well as the contacts of the subaccounts using actionable nested tree view nodes.

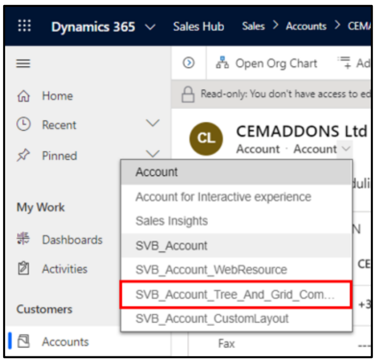
(g) TAB **DECISION MAKERS** shows use of the **TREE VIEW** to visualize all contacts that are connected to a specific selected account in a decision maker role.

(g) TAB **CONTACTS WITH ALL PREFERRED CONTACT METHODS** shows use of the **GRID VIEW** visualizing all contacts for a specific account while highlighting per row or specific cells using conditional coloring.

(h) TAB **TIMELINE** shows the use of the **TIMELINE VIEW** visualizing chronologically events or activities (phone call, meetings, …) in horizontal swimming lanes, revealing selective and actionable info per shown time item.

(i) TAB **MAP OF ACCOUNTS AND CONTACTS** shows the use of the **MAP VIEW** visualizing all contacts related to a specific account in one single map view, revealing relevant information on the account and contacts in the map details.

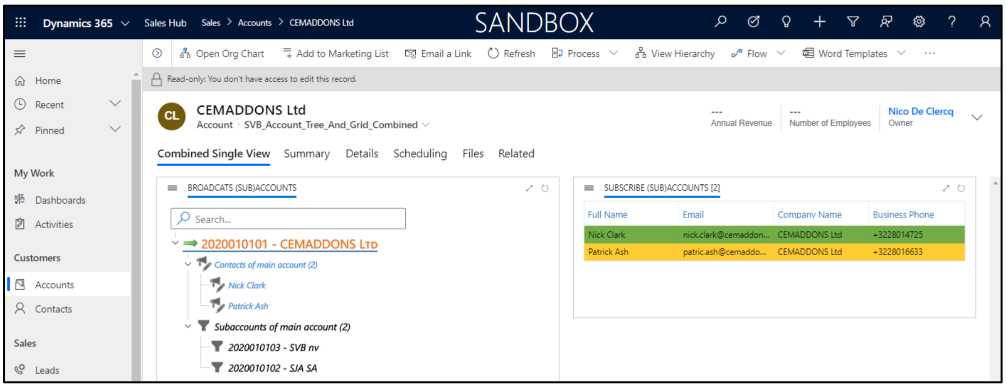
5. Switch to another for using the form selector, to go to **SVB_Account_Tree_And_Grid_Combined.**



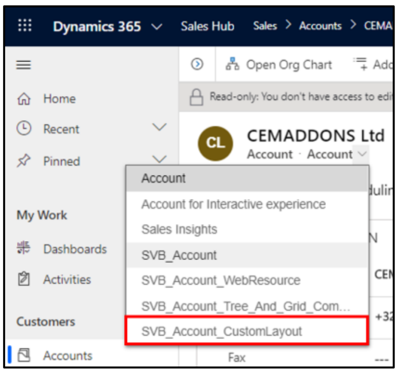
6. Broadcast and subscribe: this Form demonstrates one of the strongest features of the Single View Builder, namely using multiple single views together rom here you can navigate through the different TABs to visualize the power of Single View Builder.

   (a) The TAB **COMBINE SINGLE VIEW** shows how you can combine multiple **SINGLE VIEWS**

   (b) The TREE VIEW is **broadcasting** to 1 or multiple other views, in other words **subscribed**, in this example a GRID VIEW showing contacts of the selected account in the grid view at the left, while keeping the specific behavior and actionable character of each defined Single View.



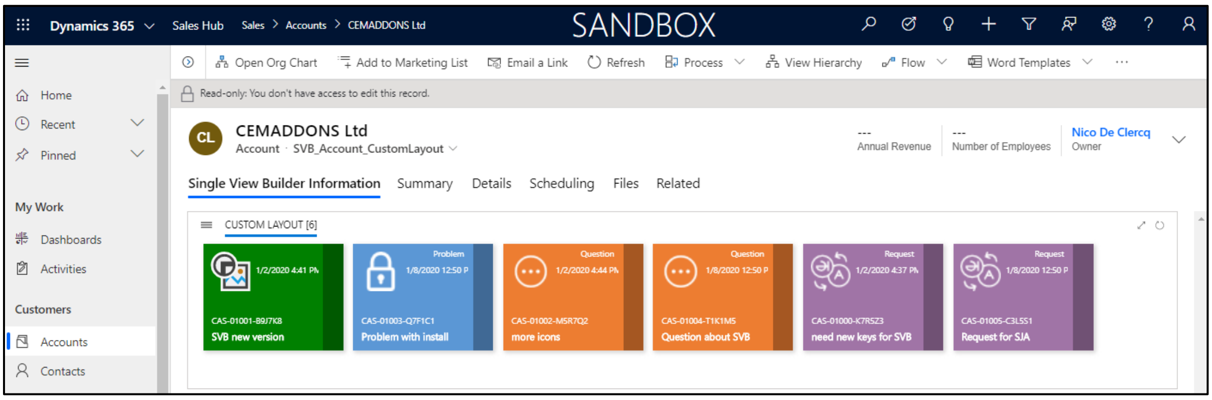
7. Switch to another for using the form selector, to go to **SVB_Account_CustomLayout.**



8. Dynamic custom layouts:

   (a) In most the view configurations as in the above examples, the colors (fore or font color, back color,…) are defined a static color

   (b) Referencing a field contain the color code or icon, is particularly convenient when you want to differentiate elements of the same type: cases based on case types, case origins, … The TAB **SINGLE VIEW BUILDER INFORMATION** in the SVB_Account_CustomLayout is showing the power of using dynamic custom layouts.



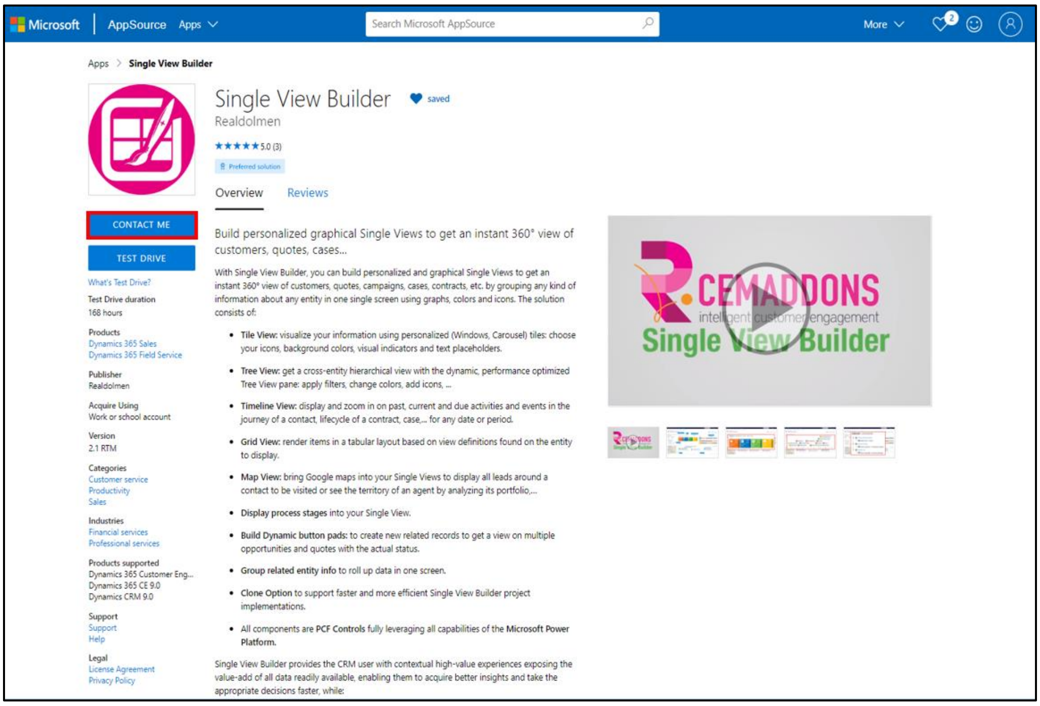
This QuickStart guide provides you:

(a) a description of the different components included in Single View Builder in **Chapter 1**

(b) additionally detailing the logical usage flow of the Single View Builder and what is needed to refresh meta data when configuring SVB components, in **Chapter 2 and 3**

(c) **Chapter 5** focusses on the Tile View component, whereas you can review Online tutorials through **Chapter 4** to show how and what to configure for the Tile View component

There is much more than meets the eye. If the QuickStart documentation isn't sufficient to support your need, if you want to elaborate more on your need our presales team will be happy to work and assist you, we encourage you to connect with us through the **CONTACT ME**-button on the landing page of Single View Builder within AppSource.

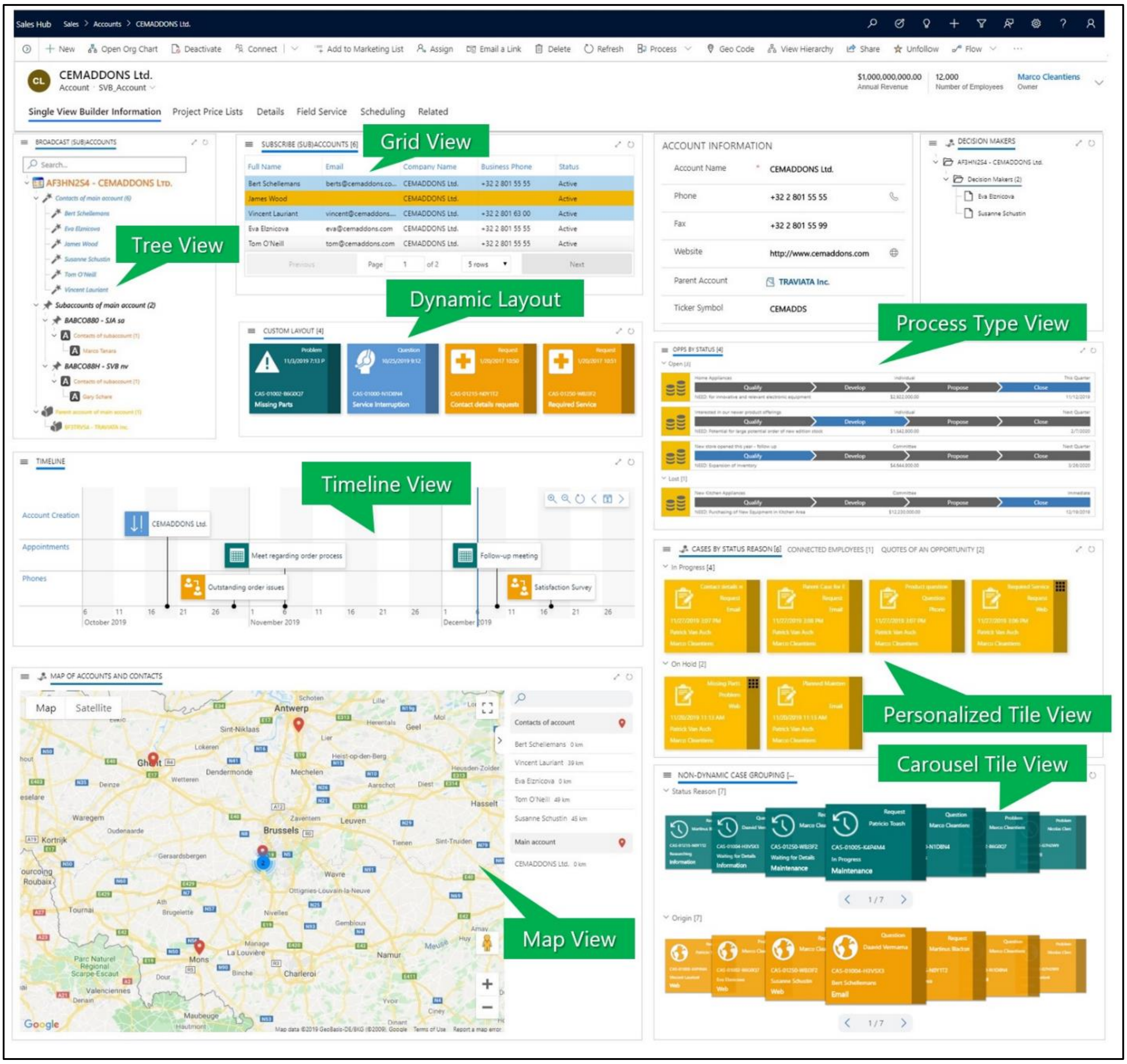


Our **Full Product Manual** - *not published through AppSource* - details configuration and usage options for all components (Tile View for all Types [Tile, Carousel, Process], Tree View, Grid View, Timeline View and Map View, installation procedure of the solution, step-by-step migration instructions for SVB installation prior to version 2.1, how components can be integrated in any View or Dashboard, the new Clone-Option for efficient SVB implementation, …

Chapter 1
Definition and components of the Single View Builder

1. Definition

Realdolmen CRM Single View Builder [abbreviated as ***SVB***] is an add-on to Microsoft Dynamics CRM that provides tools and features to create 360° views on entity items.



Some examples can be the following:

1. **General overview of an account**
   All opportunities of an account, appointments on a timeline, the hierarchy structure of an account, …The above screenshot show all opportunities of an account e.g.

2. **General overview of a contact**
   Relationships with other contacts (e.g. wife, children, friends, …), relevant events (e.g. birthday)

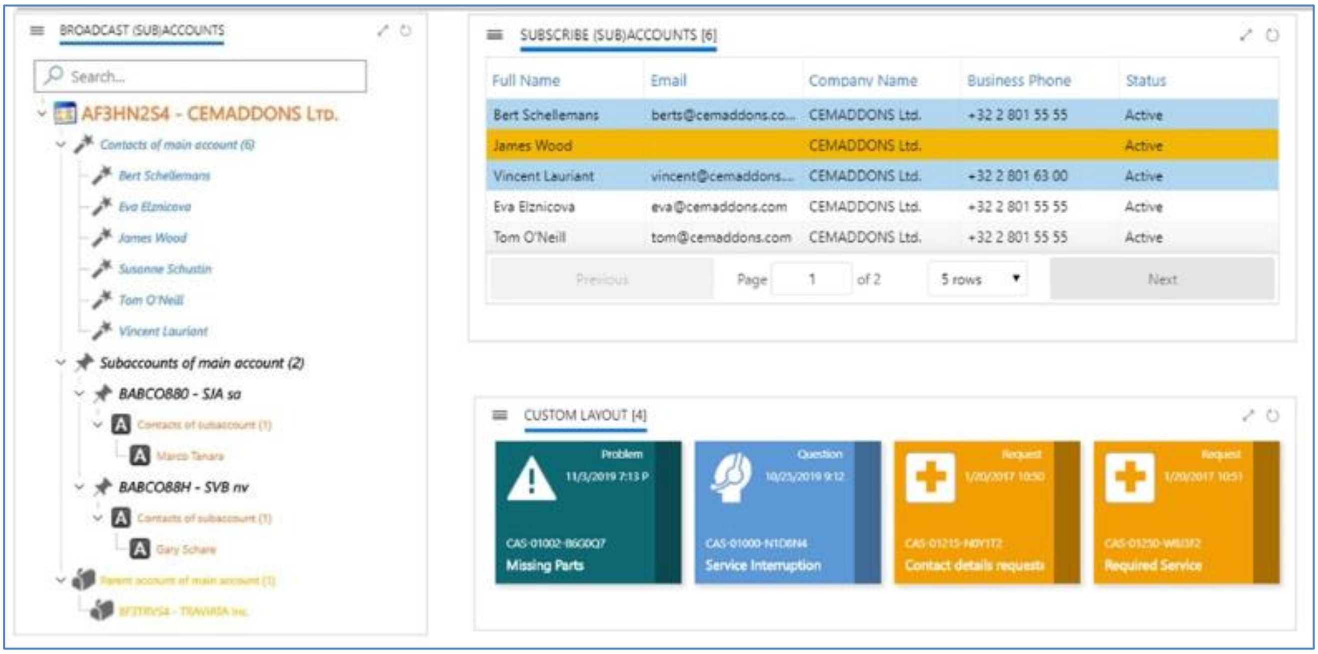
REALDOLMEN
a Gfi Group company

2. Components of the Single View Builder

The SVB has several components. We will list them up in this section, but they will be explained in much further detail during the rest of the chapters of this manual.

2.1 Single View component

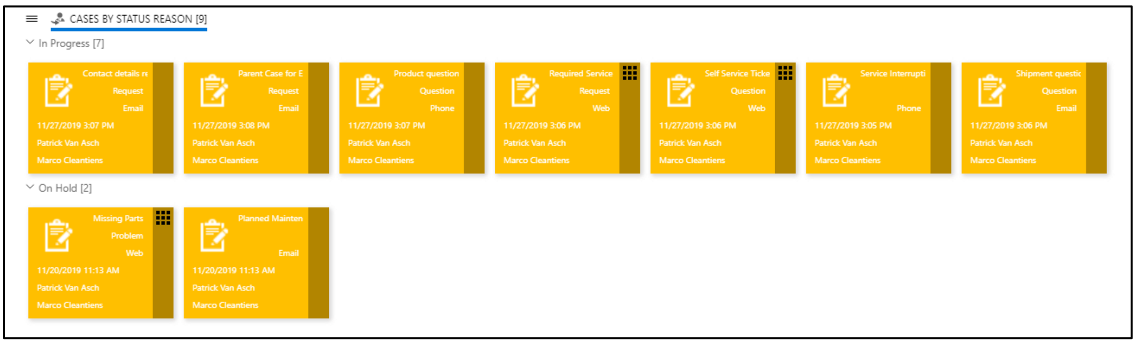
The single view component is the main component of the solution. It acts as a container which is placed on forms in order to display tree views, timelines, tiles, maps, grids, … in a tabbed view or individually.



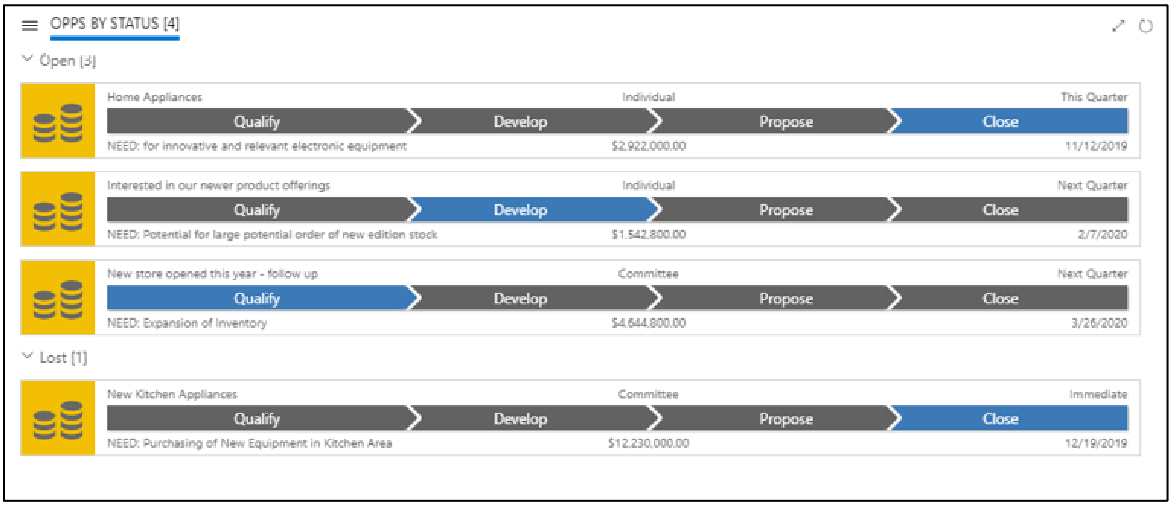
2.2 Tile View component

The tile component can be used to display information about related items (i.e.: opportunities, cases …).
Tiles are available in three flavors:

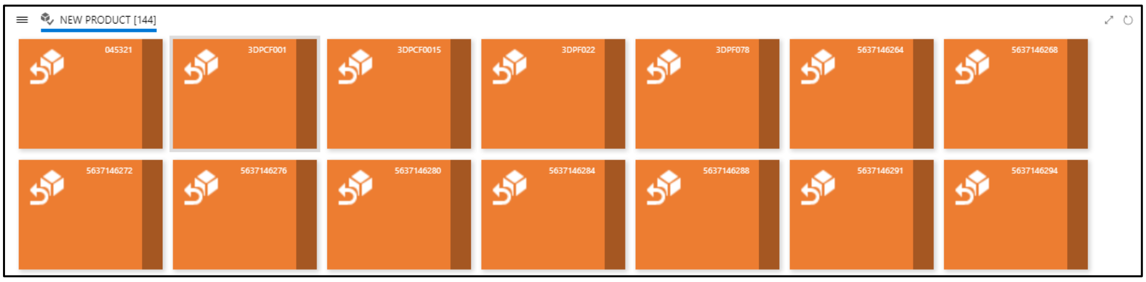
1. **Tile Type**
   Information tiles are used to provide an overview of the items with visual elements and small facts.



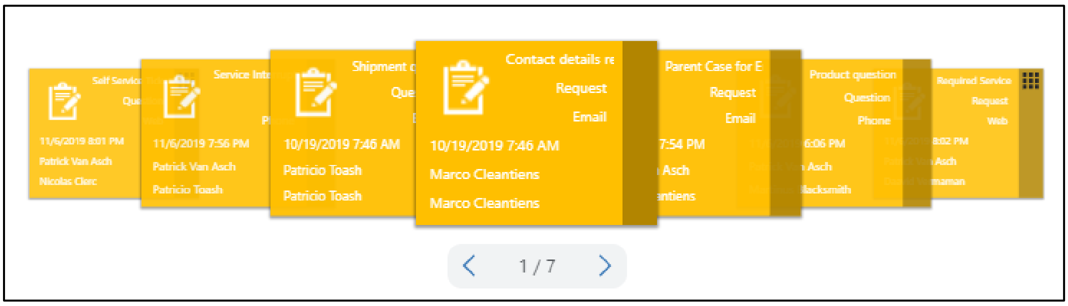
2. **Process Type**
When business process flows are applied on these items, tiles can also give users a visual indication about the current stage in the process:



3. **New Type**
This is for creating new objects from out a form. In this manual we will use the example that you can create from the contact form a new case and on that case a product will be filled up automatically, based on the product that you have double clicked on the tile view on the contact form.



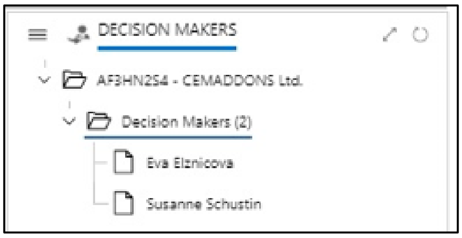
4. **Carousel visualization for Tile view components**
With version 2.1 we've added a new 2D-visualization option for the Tile View component, i.e. Carousel.

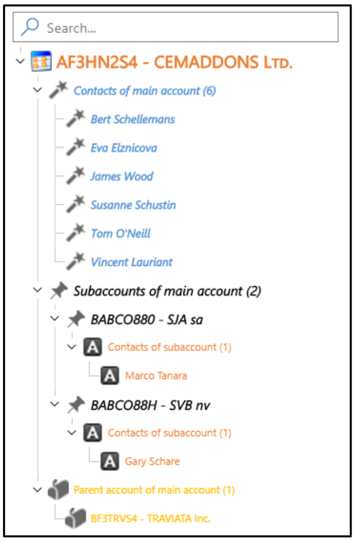


When using the carousel visualization default it will auto-operate as a circular infinite loop. Additionally, it will behave fully responsive through an auto-scaling layout. In today's version the horizontal Carousel-slider can be steered to navigate through the tiles shown using the left or right buttons underneath the tile view component. The counter in the middle, indicates which item is focused on and shown in the center of the carousel and relatively to the total amount to be shown (e.g. 1 / 7 or first out of a total of seven). This counter and the arrow navigator is positioned underneath the carousel tile view component.
We consider to add future improvements for more configuration flexibility (e.g. vertical next to horizontal carousel, position of the arrow navigator (above/under, left/right) as well as navigation options (e.g. keyboard support using left and right arrows, driving left and right direction using mouse wheel scrolling when hovering over a specific carousel, left/right swiping for touch devices).

2.3 Tree view component

The tree view component is used to visualize lists of items that are related to an entity item and that might also be related to each other hierarchically.





A tree view is typically used to display:

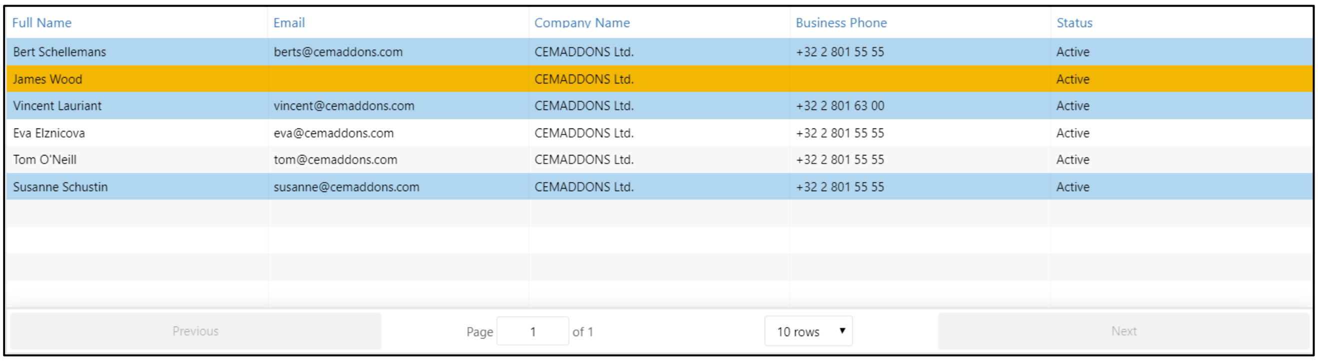
1. An enumeration of items by category

2. A hierarchical structure of items

This component can be used to navigate in the CRM: users can click on items and access their detailed information.
The tree view also benefits from common features usually available in regular tree view components: Expand and collapse, multiple selection …

2.4 Grid view component

This component lets you render items in a tabular layout. The definition of the grid relies on view definitions found on the entity to display.



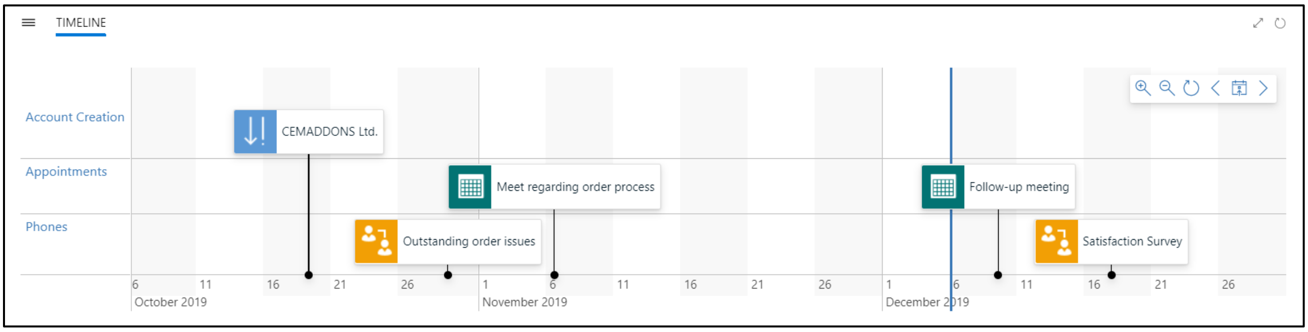
Full Name	Email	Company Name	Business Phone	Status
Bert Schellemans	berts@cemaddons.com	CEMADDONS Ltd.	+32 2 801 55 55	Active
James Wood		CEMADDONS Ltd.		Active
Vincent Lauriant	vincent@cemaddons.com	CEMADDONS Ltd.	+32 2 801 63 00	Active
Eva Elznicova	eva@cemaddons.com	CEMADDONS Ltd.	+32 2 801 55 55	Active
Tom O'Neill	tom@cemaddons.com	CEMADDONS Ltd.	+32 2 801 55 55	Active
Susanne Schustin	susanne@cemaddons.com	CEMADDONS Ltd.	+32 2 801 55 55	Active

Previous Page 1 of 1 10 rows Next

Grid View is very strong in combination with other Single View Builder components, especially when you use the so called **broadcast** functionality, which we will explain further on. In the above example you see a grid combined with a tree view where you can select an office that will display the employees and agreements of that office. You can even work with colors in your grid.

2.5 Timeline component

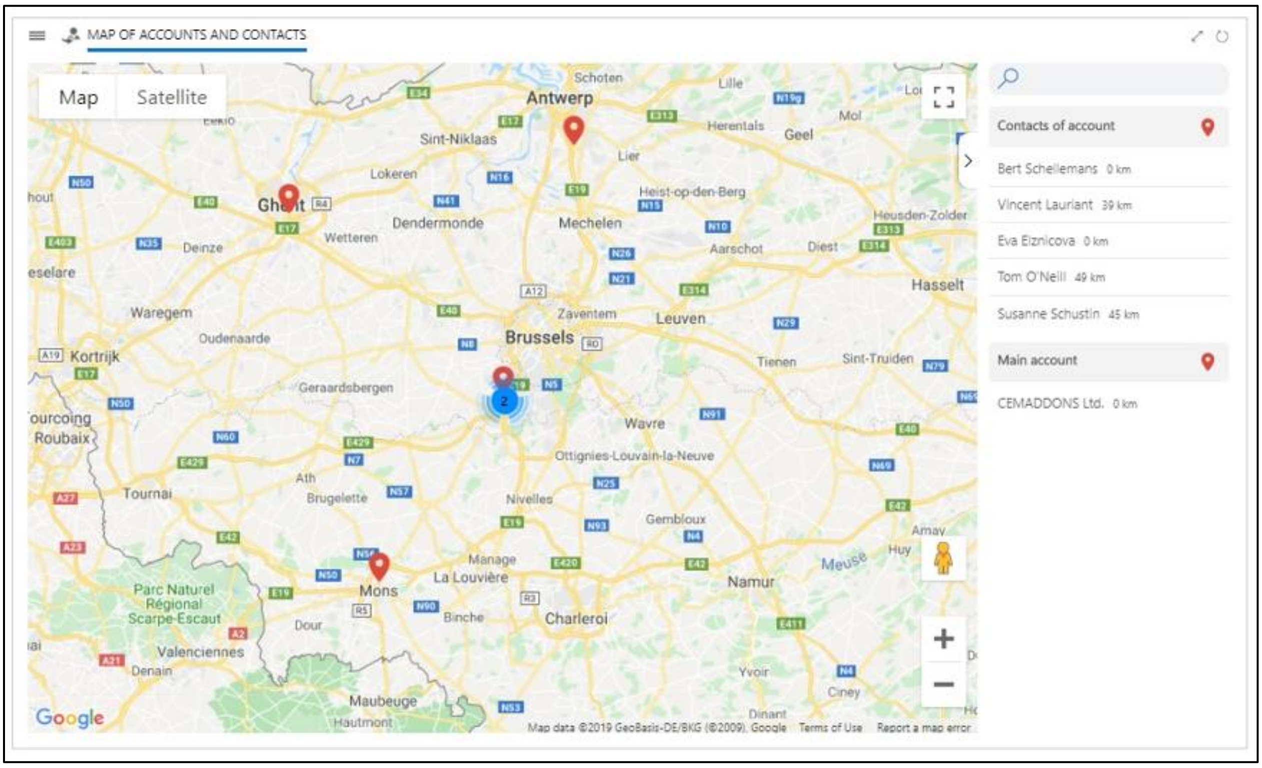
The timeline component is used to chronologically visualize events or activities related to an entity item. For example, a timeline could display all created cases related to a customer or all the made phone calls or appointments.



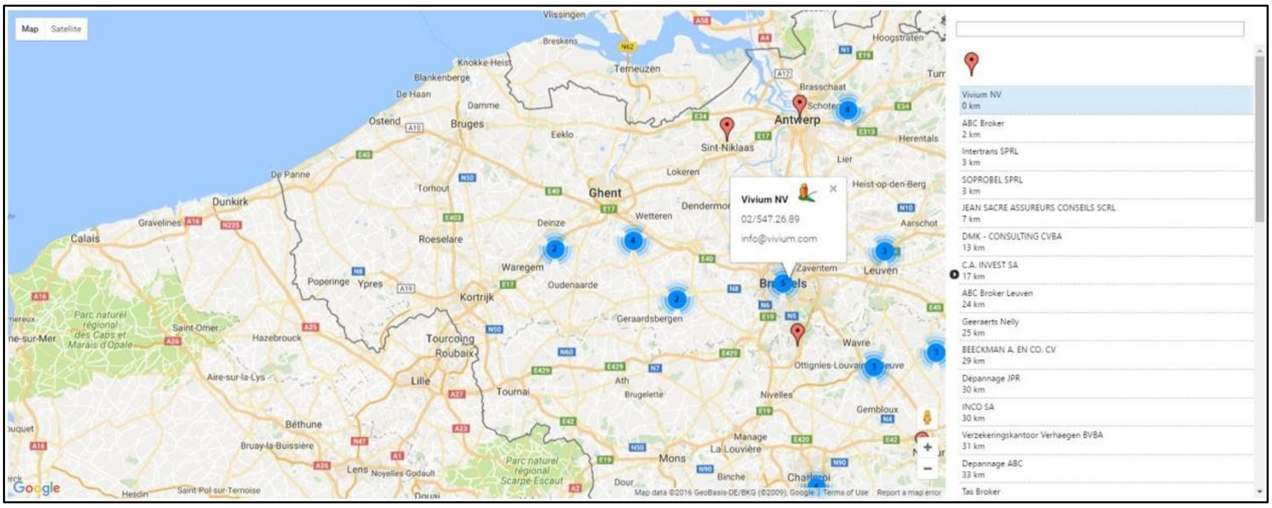
Types of activities can be mixed in order to show a more elaborated timeline (i.e.: the component dynamically rearranges the elements depending on the number of elements to display).
Users can use the timeline to navigate between elements displayed on the timeline. Furthermore, controls such as zoom in / out facilitate browsing in the timeline.

2.6 Map view component

Map view is the component that visualizes your geolocation abled data on a map by using Google maps. While using standard functionality on latitude and longitude you can see where your accounts, leads, contacts are situated.



Map view is not limited to one record (as the standard Bing Map integration) but let you combine multiple information of different entities on one map view. So you can see all of your accounts on one map. In some examples we have set up Map view to display the cases and their occurrence. Map view can be extended with a list to show the record on the map.



The indication of the record allows you to go to street view or go directly to the record by double clicking.

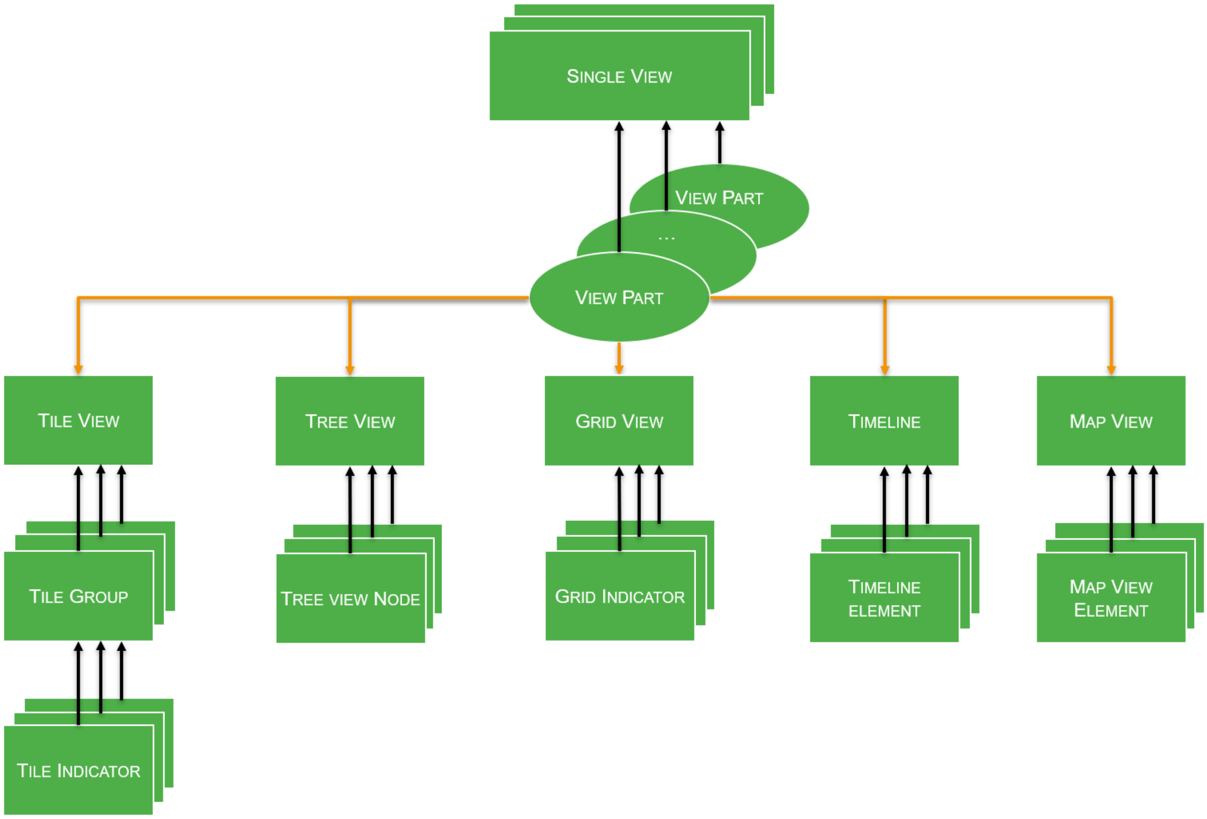
2.7 Clone Option

Your SVB Architect together with the business will configure the suite of visual components to enrich the graphical user interfaces. Working with and configuring- tree views, timelines, tiles, map and grid views and other will be part of his core task. To facilitate and support the SVB architect realizing SVB implementations more efficiently and less prone to errors, we have added a **Clone Option**.

This new Clone Option for Single View Builder is available on each component (e.g. Tile View component) as well as on the child level of each component (e.g. Tile Group, Tile Indicator for that Tile View component).

3. Logical Data Model

As explained above SVB offers several Single View Components, you can configure and combine.



The logical data model reveals:

1. you can combine one or more Single Views

2. a Single View can contain multiple View Parts

3. a View Part can be a tile, a tree, a grid or a timeline or map view

4. each view type has specific configuration settings

All configurations are done through point and click screens, full details of the hierarchical logical data model can be found in Chapter 18.

4. Color palettes

4.1 RGB and Hex Triple color models

Customizing your Single Views involves choosing colors for background, font, icons,etc. Both the RGB and hex triplet color model – commonly used in HTML, CSS, SVG and other computing applications to represent colors.- are supported to define your preferences.

(a) the **RGB color model** is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors

(b) **Hex Triplet** is a six-digit, three-byte hexadecimal number representing the red, green and blue components of the color. One byte represents a number in the range 00 to FF (in hexadecimal notation), or 0 to 255 in decimal notation. This represents the least (0) to the most (255) intensity of each of the color components. Thus web colors specify colors in the 24-bit RGB color scheme. The hex triplet is formed by concatenating three bytes in hexadecimal notation

4.2 Think ahead – define a color palette for consistency

If you plan to align the colors used in your Single View, think ahead and define your preferred color palette beforehand or keep your corporate color palette next to you, thus creating consistency in your SVB experiences. Example of some color palettes (*horizontally*) in Hex Triplet color palette format.

Chapter 2
The logical usage flow of the Single View Builder

1. Introduction

If you want to use the Single View builder (SVB), you must follow certain steps to get the job done. If you don't follow these steps, you will have problems in configuring the SVB.

2. The flow itself

2.1 Refresh metadata

Retrieve info concerning all entities that you plan to manipulate

2.2 Create the view components

You can create now different components in a random desired order based on the needs of the customer. The components that can be created during this phase in the process are:

- Tile view
- Tree view
- Timeline
- Map view
- Grid view

2.3 Bundle those elements into single view component

Once you have created the components mentioned in the above step, you have to assemble the desired ones within a single view component.

2.4 Create single view form

Before you can actually benefit from the SVB, you need to setup a good working form with some adjustments.

3. Important remark for the tutorial

This quick start guide will be focusing on one example with the Tile View, a more elaborated tutorial can be found in our full manual for Single View Builder using multiple lifetime examples. Within both tutorials we can't follow 100% the above logic, because you need to see fast some results. In this tutorial we will use the following process:

1. Refresh metadata
2. Build a component (of type Tile view, Tree view, timeline view, map view, grid view)
3. Create a single view component
4. Add the created component of step 2 into the single view component of step 3
5. Create a form (or modify an existing one) and add the single view component to the form
6. Test all modifications
7. Build a component (of type Tile view, Tree view, timeline view, map view, grid view)

8. Add this to the single view component of step 3

9. Test all modifications

10. Continue the steps 6-9
If there are too many components on one single view component, we will create a new single view component.

11. Test all modifications

Chapter 3
Refresh metadata

1. What?

Retrieve information concerning all entities that you plan to manipulate through one of the components. The configuration of all controls of the Single View Builder rely on entities and their attributes. For technical reasons, that information is stored in a “cache” in order to facilitate the configuration afterwards.

2. When?

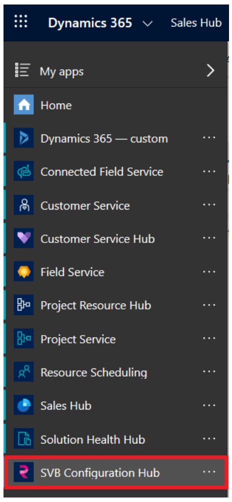
1. Necessary to start working with the single view builder
2. When an entity has been modified

REMARK:

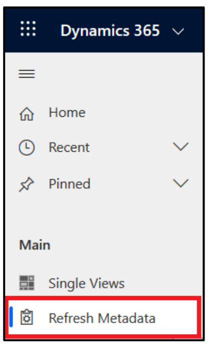
Refreshing all the metadata may take some time

3. How?

1. Navigate via the App Switcher to the **SVB Configuration Hub** App.



2. Select “**Refresh Metadata**” from the sitemap.



3. In the upcoming screen, you select the entities that you need and you click afterwards on the button **REFRESH**. For our tutorial, we need the following entities:

   (a) Account

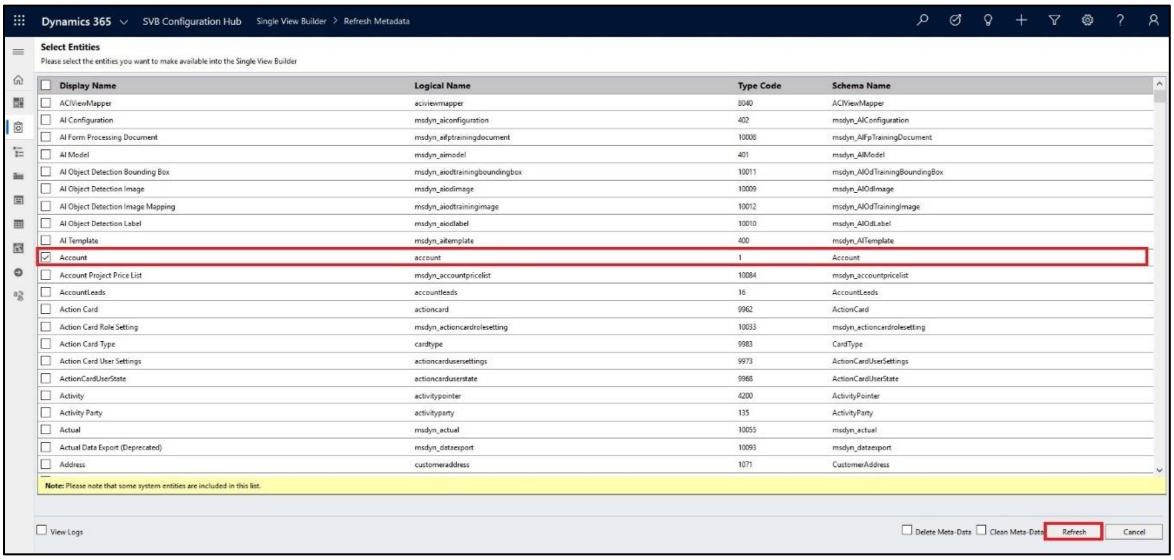
   (b) Appointment

   (c) Case

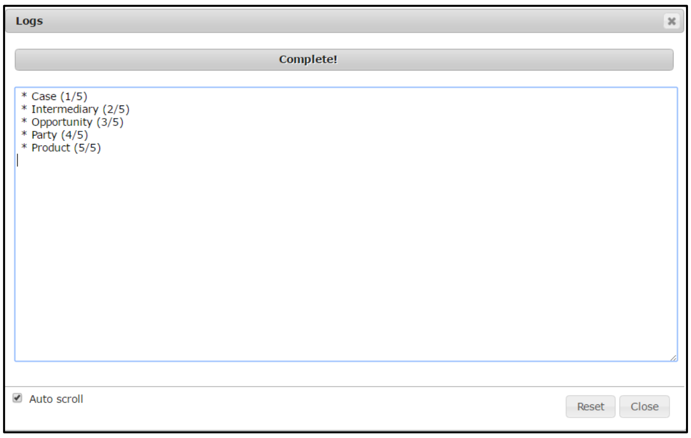
   (d) Contact

   (e) Opportunity

   (f) Products



4. When finished, you will get the following window, in which you click on the button **CLOSE**

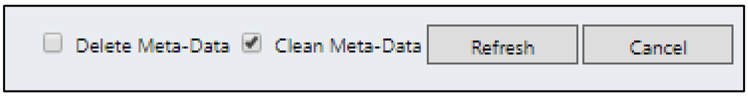


5. If you want to open this window again, you can use the option **VIEW LOGS** on the main screen of the **REFRESH METADATA**

4. Clean and delete metadata

It is of course also possible to delete metadata. On the main screen, you will find two options for that.
When refreshing metadata; the SVB is actually creating also kinds of relationships between the selected entities. So if you add e.g. Accounts and Case to the refresh metadata, SVB is creating a relation between those entities.

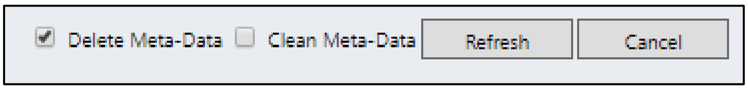
1. If you choose **CLEAN METADATA**:



   (a) SVB will refresh all related metadata and also verifies whether the relationships linked to the entity – that you are cleaning the metadata for – are still in use, if they are not they are removed.

   (b) the cleaned metadata then remains available for the SVB configurations.

2. If you choose **DELETE METADATA**:



   (a) the entities Accounts and Cases will be effectively delete

   (b) and they will not be available anymore for the SVB configuration

Chapter 4
Online Tutorials

TUTORIAL

How to visualize SVB components on an account form.

http://ior.ad/6MAl

TUTORIAL

How to visualize SVB components using WebResources

http://ior.ad/6MAB

TUTORIAL

How to create a Tile Group

http://ior.ad/6Mw4

TUTORIAL

How to create a Tile Indicator

http://ior.ad/6Mwy

TUTORIAL

How to create extra filters using Fetch XML

http://ior.ad/6MAK

TUTORIAL

How to create New Single View and View Part (Type = Create New)

http://ior.ad/6MGN

TUTORIAL

How to create New Tile Group (full fetch XML)

http://ior.ad/6MHa

TUTORIAL

How to Create New Tile Group (full fetch XML)

http://ior.ad/6MHb

TUTORIAL

How to Create New View Part (full fetch XML)

http://ior.ad/6MHb

TUTORIAL

How to create new Tile Group (Process)

http://ior.ad/6MBu

TUTORIAL

How to create new Tile Group (show connected account contact employees)

http://ior.ad/6MGd

TUTORIAL

How to create new Tile Group (Type = Create New)

http://ior.ad/6MGD

How to create new tile group NonDynamic 1 (Status Reason)

http://ior.ad/6MAR

How to create new tile group NonDynamic 2 (Origin)

http://ior.ad/6MAU

How to create New Tile View (full fetch XML)

http://ior.ad/6MH7

How to create New Tile View (Process)

http://ior.ad/6MBf

How to create new Tile View (Process)

http://ior.ad/6MFW

How to create new Tile View (Type = Create New)

http://ior.ad/6MGw

How to create new Tree View Node

http://ior.ad/6MKS

Svb Config - How to create New Treeview

http://ior.ad/6MHI

How to create new View Part for Non Dynamic

http://ior.ad/6MB4

How to create Single View and add a View Part

http://ior.ad/6MwL

How to create SVB Form on Contact

http://ior.ad/6MGS

How to create Tile View (show contacts connected to account)

http://ior.ad/6MG6

TUTORIAL	**How to create Tile View (Show related Cases)** http://ior.ad/6MvO
TUTORIAL	**How to create Tile View without dynamics grouping** http://ior.ad/6MAP
TUTORIAL	**How to create View Part (show account contact employee connections)** http://ior.ad/6MGj
TUTORIAL	**How to download FetchXML using Advanced Find in CRM** http://ior.ad/6MAN
TUTORIAL	**How to set Tile Group Display Color & Icon** http://ior.ad/6MwE
TUTORIAL	**How to set Tile Group to Carousel** http://ior.ad/6MAy

Chapter 5
Tile View component

1. Definition

The tile component can be used to display information about related items (i.e.: opportunities, cases …). At creation, you will need to create a tile view, which consists of one or more different tile groups.
Tiles are available in three flavors :

1. **Tile Type**
   Information tiles are used to provide an overview of the items with visual elements and small facts.

2. **Process Type**
   When business process flows are applied on these items, tiles can also give users a visual indication about the current stage in the process:

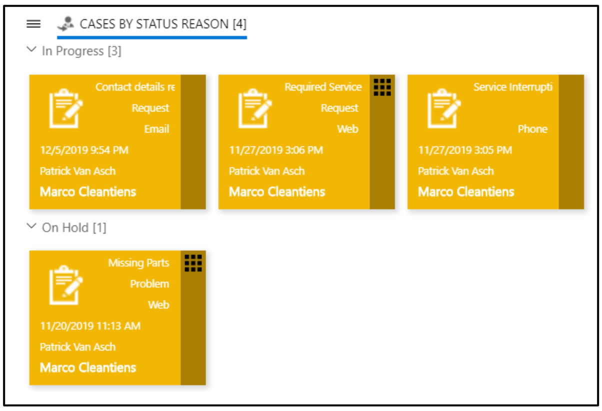
3. **New Type**
   This is for creating new objects from out a form. In this manual we will use the example that you can create from the contact form a new case and on that case a product will be filled up automatically, based on the product that you have double clicked on the tile view on the contact form

2. Practical cases

2.1 A tile view of tile type that shows you cases for an account

2.1.1 Desired result

We want to create a tile view of the tile type which shows all cases of a selected account, grouped by status reason. Within each group, the tiles must be ordered on the field **Case Title** in ascending order. The tile view must look as follows:



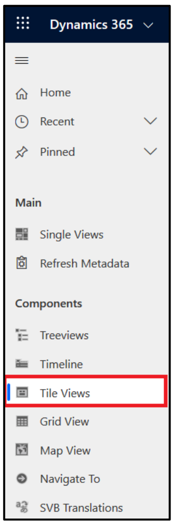
You see that the cases in this tile view are grouped on status reason (*On hold* , *In progress*) and within each group the tiles are ordered ascending on the Case Title (*CRM SVB Problem*, *CRM synch issue, …*).

2.1.2 Solution

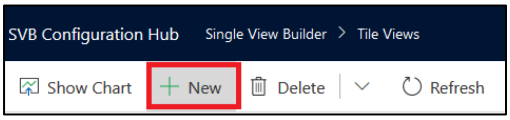
2.1.2.1 Create the tile view component

1. Navigate via the App Switcher to the **SVB Configuration Hub** App.

2. Select **TILE VIEWS** from the sitemap



3. Click the **NEW**-button in the command bar



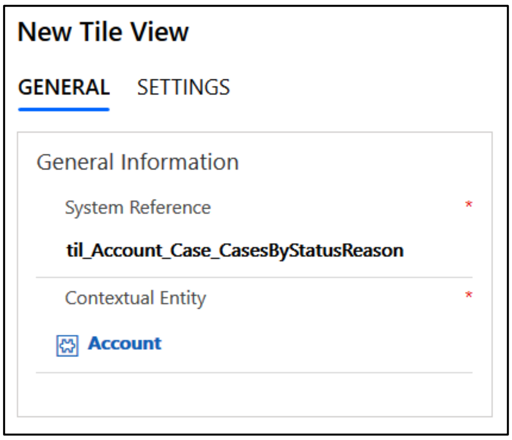
4. Fill in the data as explained here beneath and start with the **GENERAL** tab and it's three sections:

   (a) **General information** section

   (b) **Relationship** section

   (c) **Tile groups** section

GENERAL TAB - **GENERAL INFORMATION** SECTION



(a) **System Reference**
This is the name of the tile view configuration. We advise you the following naming convention, which is a bit

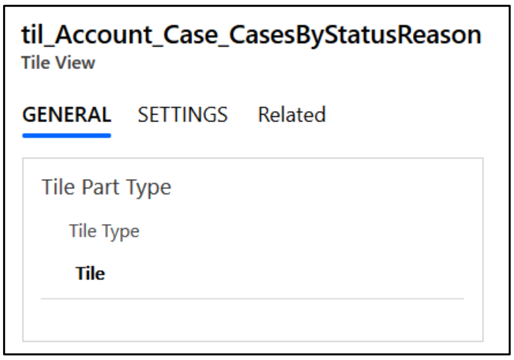
similar to the Hungarian naming convention that is often used when working with objects. The name has four parts, which will be separated by an underscore

A three character abbreviation to indicate the object type	As we are creating a tile view, we could use **til**
The name of the contextual entity. This is the entity on which you will put the tile view	In our example this will be the Account entity
The name of the related entity. This is the entity for which you want to see the tiles	In our example this will be the Case entity
A description of the tile view	In our example we want to the Cases by Status reason

The final result for our example is then til_Account_Case_CasesByStatusReason

(b) **Contextual entity**
Reference to the entity on which the tile view will be placed.
In our example we use Account

TILE VIEW - GENERAL TAB - **TILE PART TYPE** SECTION

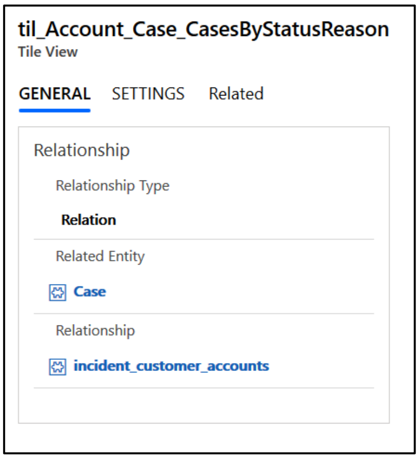


(a) **Tile type**
With this field, you can select which type of tile view you would like to create. As mentioned before, three types are available:

- Tile
- Process
- New

We will give an example of each type further on in this manual.
In our example we use Tile

TILE VIEW - GENERAL TAB - **RELATIONSHIP** SECTION



(a) **Relationship type**
Type of link between the contextual entity and the related entity. The type can be:

(1) **Relation**
Contextual and related entities are linked together via a [1:N] relationship. In our example this means that we would select all cases that are really related to the accounts.

(2) **Connection**
Contextual and related entities are linked together by using CRM connections. In our example this means that we would select all cases that are connected to the accounts.

In our example we use Relation

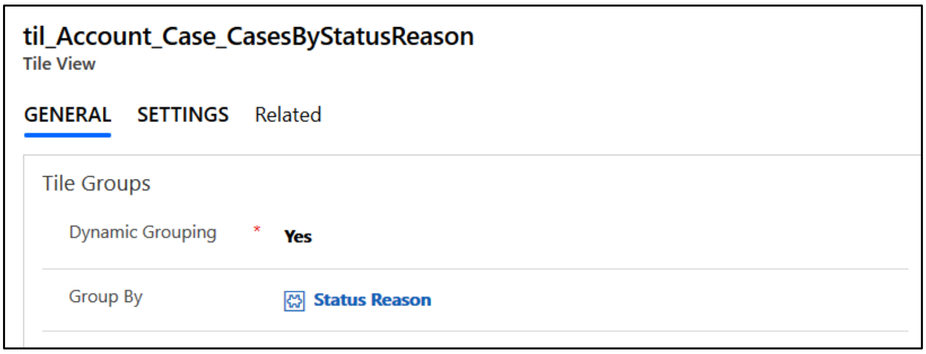
(b) **Related entity**
Name of the related entity
This is the entity for which you want to see the tiles.
In our example we use Case

(c) **Relationship**
The relation you want to use. The relations are filtered according to the contextual and the related entities.
In our example we use incident_customer_accounts

Important Remarks on Relationship

- This is one of the fields that you have to select carefully. Very often people take the wrong relationship, which will give you no or very strange results. Keep that in mind!!!
- If you have selected in the field **Relationship type** the option **Connection**, you will have to select a **Connection role to** instead of **Relationship**. We will give an example later in this course (practical case: "A tile view of tile type that shows all contacts connected to an account with the role "employee").

TIEL VIEW - GENERAL TAB - **TILE GROUPS** SECTION

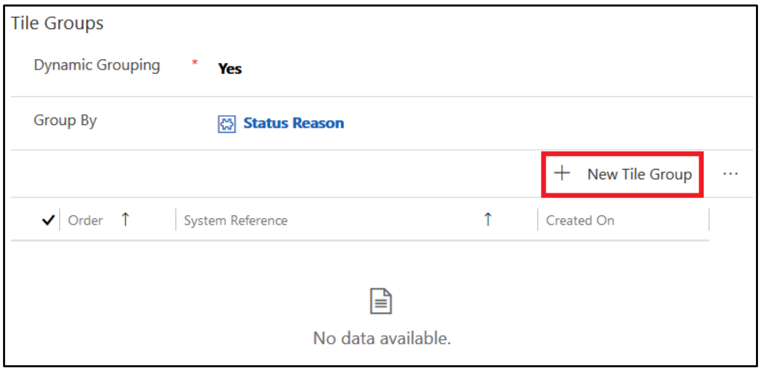


(a) **Dynamic grouping**
With this field you can define if groups of tiles should be dynamically created or if each group should be individually defined.
If you put the field on **yes**, this means you are going to use a grouping. By selecting **yes**, you will see an extra field **Group by**, in which you have to indicate on which field you want to group.
Keep in mind that if you select **yes**, you can only add one tile group to the tile group view.
If you put the field to **no**, you don't group the tiles and it will be possible to add more than one tile group. If you do that, the field **order** on the tile group form will determine in which order you see the different tile groups. We will give an example of that later in the course (practical case: "A tile view of tile type that shows you cases for an account without dynamic grouping")
As long as you don't add a tile group, you will see no results.
In our example we use Yes

(b) **Group by**
If you have chosen for **yes** in the previous option, you must specify here on which field you want to group.
In our example we use Status Reason

5. Before you can add a tile group, you must save the tile view by using the **SAVE**-button in the command bar.

6. Add a tile group to the tile view by clicking on the +-sign on the right side. With a tile group, you specify in fact which tiles you want to see and how they have to appear on the tile view.



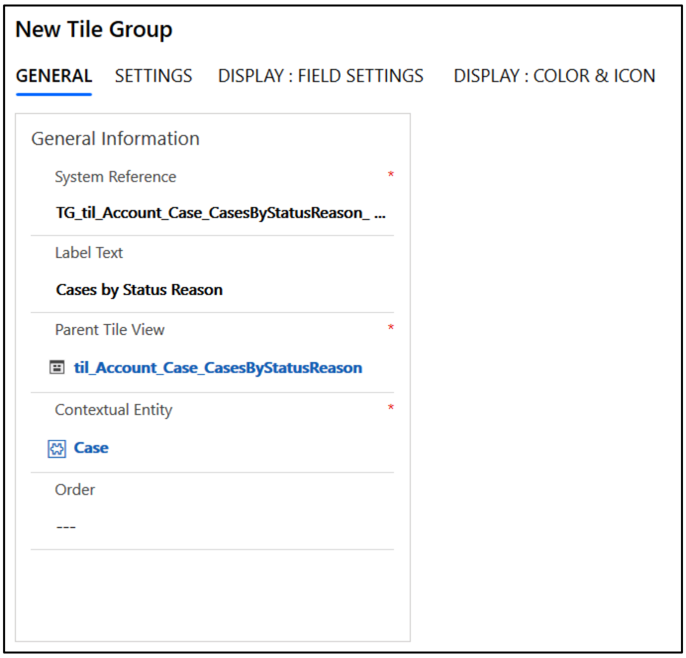
7. Fill in the data as follows, by going through the **GENERAL** tab and it's sections:

   (a) **General information** section

   (b) **Others** section

   (c) **Indicator** section

TILE GROUP – GENERAL TAB - GENERAL INFORMATION SECTION



(a) **System Reference**
This is the name of the tile group configuration. We advise you the following naming convention, which is a bit similar to the Hungarian naming convention that is often used when working with objects. The name has different parts, which will be separated by an underscore

An abbreviation to indicate the object type	As we are creating a tile group, we use **TG**
The name of the tile view on which you put the tile group.	Our tile view was called **til_Account_Case_CasesByStatusReason**
A short description of the tile group	As we visualize the cases by status reason we could take something as **CaseByStatR**

The final result for our example is then TG_til_Account_Case_CasesByStatusReason_CaseByStatR

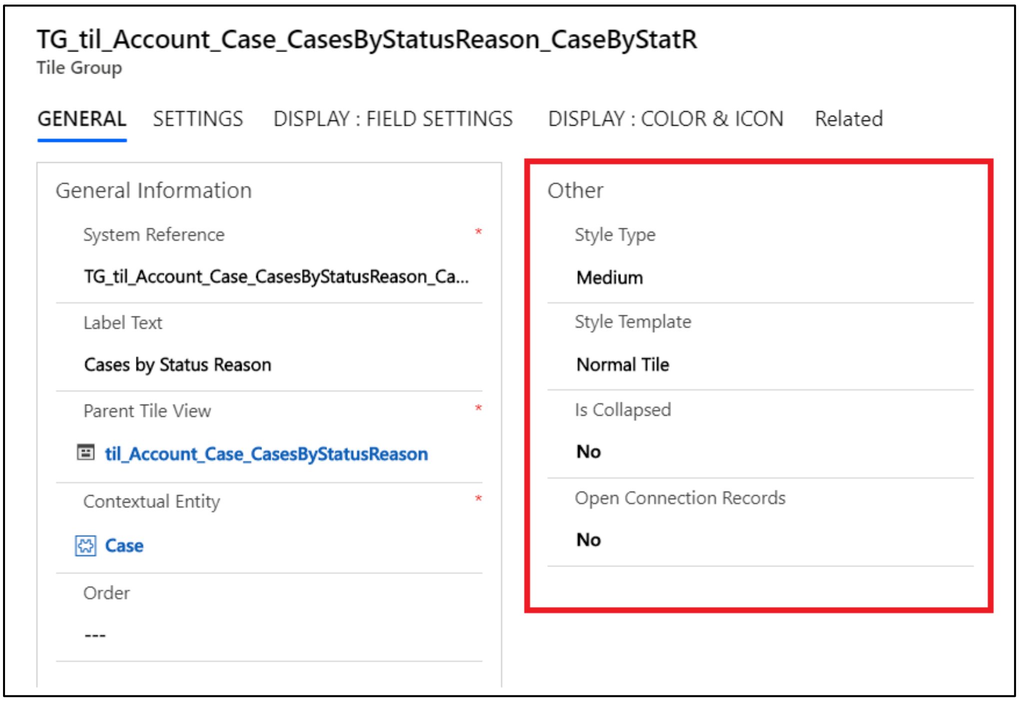
(b) **Label text**
This is the label of the tile group. If you have chosen a tile view without dynamic grouping, this label will be used as label for the tile grouping. So it is only of use if you don't use dynamic grouping.
In our example we use Cases by Status Reason, and we will see later where that comes up (when we create an example without dynamic grouping.)

(c) **Parent tile view**
Reference to the parent tile view, which is filled up automatically.
In our example it is thus til_Account_Case_CasesByStatusReason

(d) **Contextual entity**
Reference to contextual entity, which is filled up automatically.
In our example it is thus Case

(e) **Order**
If you don't use dynamic grouping, it is for ordering the tile groups. You will do that by filling in a number starting from 1 for the first, 2 for the second, …
In our example we leave it blank

TILE GROUP – GENERAL TAB - **OTHER** SECTION

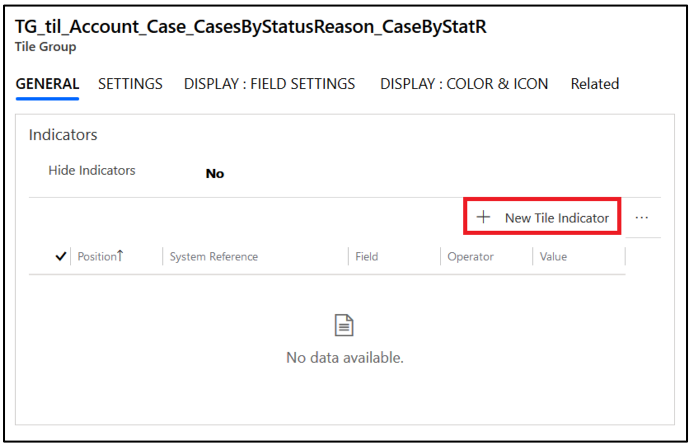


(a) **Style type**
Size of the tile: small, medium, large (only for "Tile" type)
In our example we take Medium

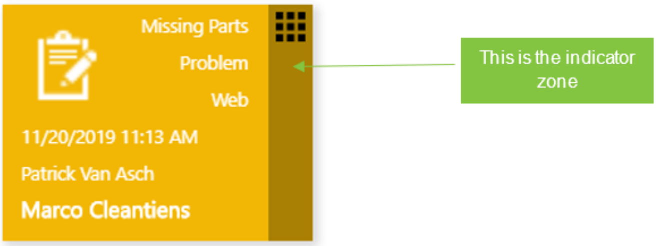
(b) **Is collapsed**
Define if the tile group should be collapsed or not.
In our example we take No

(c) **Open connection record**
If you had selected *Connection* as relationship type, you could ask if the connection record must be opened when you double-click on it
In our example we take No

TILE GROUP – GENERAL TAB - **INDICATOR** SECTION



You have the possibility to show extra indicators on the right side of a tile view of the tile type as you can see on the following image.

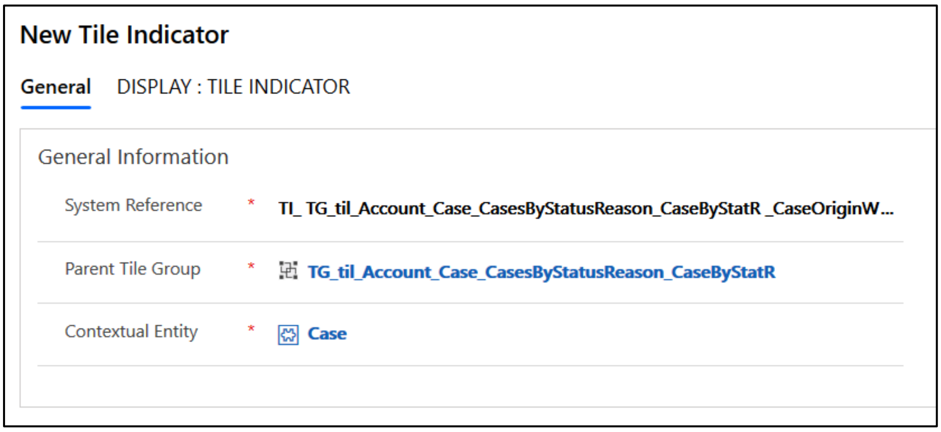


(a) **Hide indicators**
You can ask to hide the created indicators or not
In our example we take No

(b) **Create indicators**
Therefore you click on the +-sign on the following part of the screen. Keep in mind that the data of the tile group must be saved before you can do that.

Now you can start filling up the necessary data

TILE INDICATOR – GENERAL TAB - **GENERAL INFORMATION** SECTION



(a) **System Reference**
This is the name of the tile indicator configuration. We advise you the following naming convention, which is a bit similar to the Hungarian naming convention that is often used when working with objects. The name has different parts, which will be separated by an underscore

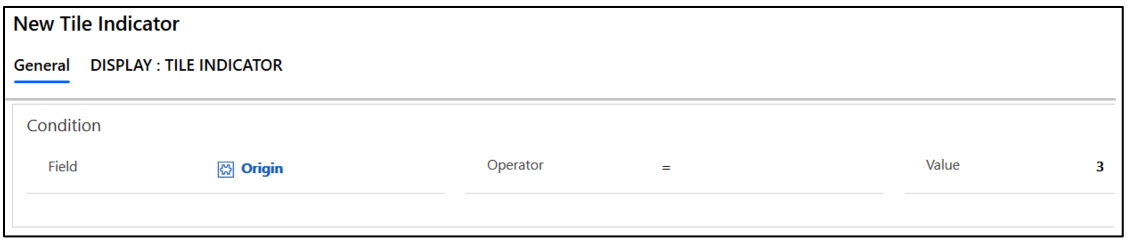
An abbreviation to indicate the object type	As we are creating a tile group, we use **TI**
The name of the tile group on which you put the tile indicator.	Our tile view was called **TG_til_Account_Case_CasesByStatusReason_CaseByStatR**
A description of what the indicator shows	In our example we will create an indicator for a case origin coming from the web so we take something as **CaseOriginWeb**

The final result for our example is then
TI_ TG_til_Account_Case_CasesByStatusReason_CaseByStatR _CaseOriginWeb

(b) **Parent tile group**
This is the tile group on which you create the indicator. It is filled up automatically
In our example it is TG_til_Account_Case_CasesByStatusReason_CaseByStatR

(c) **Contextual entity**
The entity for which you are creating tiles and tile indicators It is filled up automatically
In our example it is Case

TILE INDICATOR – GENERAL TAB - **CONDITION** SECTION

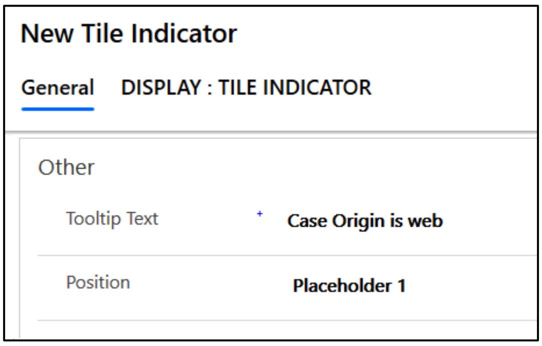


(a) **Field**
The field that you want to add as indicator
In our example it is Origin

(b) **Operator**
This is a classic operator as we have in e.g. advanced find: =, contains data, …
In our example it is =

(c) **Value**
What is the filter of the field that you want to use?
Watch out because you will need the value of the option set as it is determined within the settings of the environment
In our example we want to see an indicator for all cases with an origin of type Web, so we need the value 3

TILE INDICATOR – GENERAL TAB - **OTHER** SECTION



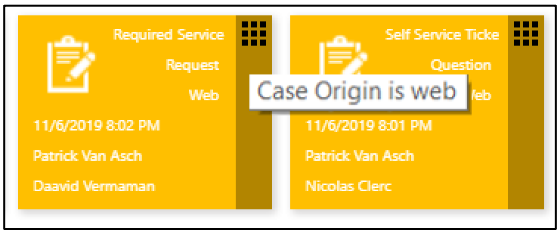
(a) **Tooltip text**
If you will hover over the indicator on the tile, you can have a tooltip which explains what the indicator is showing
In our example we will use Case Origin is web

(b) **Position**
You can add maximum 5 indicators. With this field you specify on which position you want to see the indicator
In our example we will use Placeholder 1

TILE INDICATOR - DISPLAY:TILE INDICATOR-TAB - **ICON** SECTION

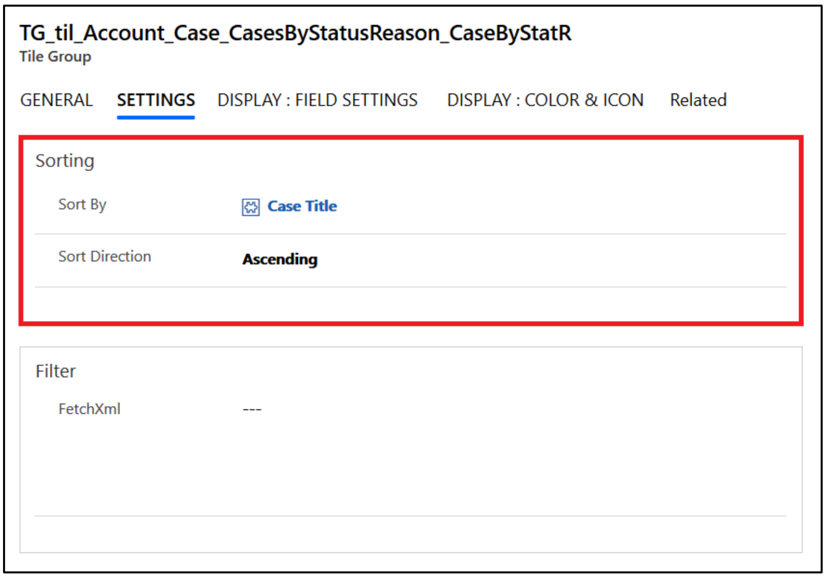
(a) **Icon**
Icon displayed in the indicator. The URL can point to an internal web resource (relative path) or an external image (absolute URL).
When a dynamic icon field is mentioned, the default icon is only displayed when no value has been found in the dynamic field.
You don't need to type the URL : simply click on the desired image in the icon library below the URL. The library is showing all the icons with a 16x16 size, saved as web resources in your CRM.
In our example we take whatever icon we like.

8. Save and close the modifications of the tile indicator and the final result of the tile with indicator will be the following



9. You are back in the tile group form and can continue with the **SETTINGS** tab, in which there are several sections
   (a) **Sorting** section
   (b) **Filter** section
   (c) **Technical configuration** section

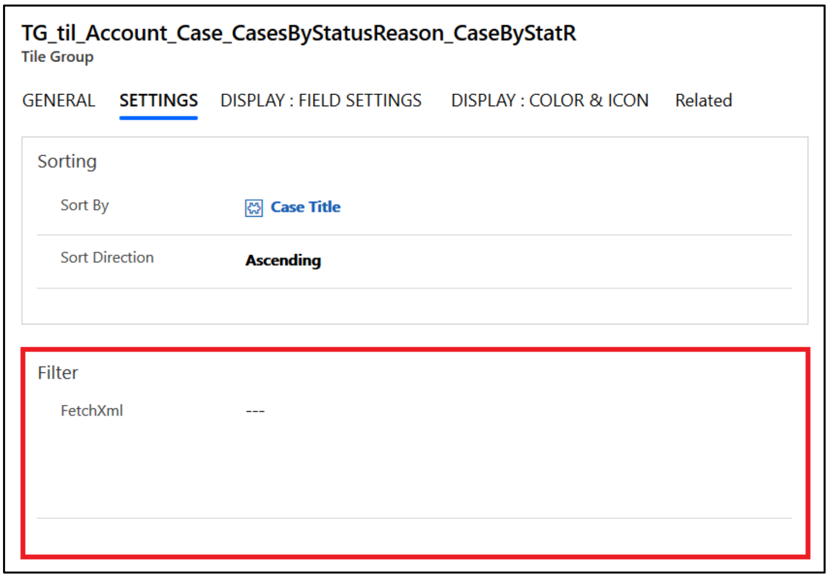
TILE GROUP – SETTINGS TAB - **SORTING** SECTION



(a) **Sort by**
How do the tiles within the groups have to be sorted?
In our example we use Case Title

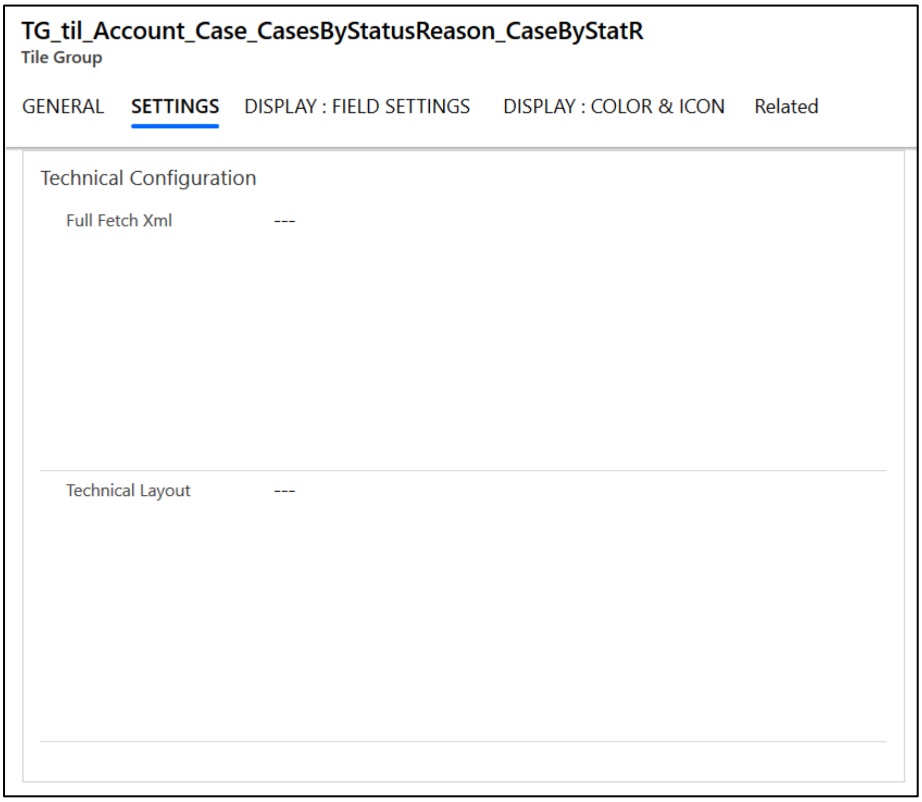
(b) **Sort direction**
Sort order: ascending or descending
In our example we use Ascending

TILE GROUP – SETTINGS TAB - **FILTER** SECTION OF THE TILE GROUP



(a) **FetchXML**
Additional filter to apply on the query, written in FetchXML format. If you have chosen a tile view without dynamic grouping, you may use this filter to create some custom groupings (for example, the cases of the last 6 months). We will create an example later in this course.
In our example we leave it blank

TILE GROUP – SETTINGS TAB - **TECHNICAL CONFIGURATION** SECTION



(a) **Full Fetch XML**
We will explain in a later example what we can do with that.

(b) **Technical layout**
This is about programming stuff and out of the scope for this course

10. Save what you have done in the **SETTINGS**-tab, and continue with the **DISPLAY: FIELDS SETTINGS** tab and it' sections:

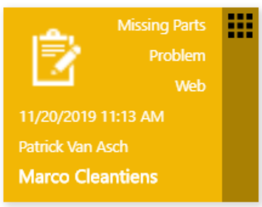
(a) **Small fields** section

(b) **Long fields** section

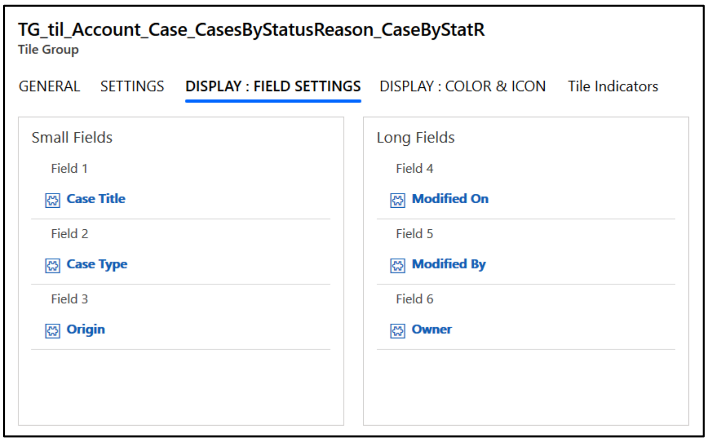
(c) **Settings** section

TILE GROUP - DISPLAY:FIELD SETTINGS-TAB - SMALL & LONG FIELDS SECTION

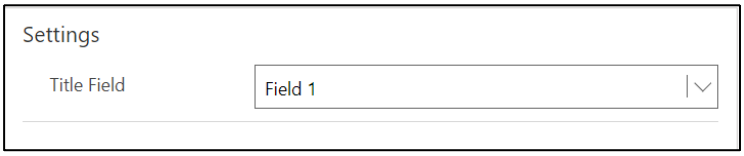
On a tile, you can add the content of six fields as you see on the following image



The small fields are the ones in the upper right corner; the long fields are the ones on the down left corner.
In our example we have specified them as follows



TILE GROUP - DISPLAY:FIELD SETTINGS-TAB SETTINGS SECTION



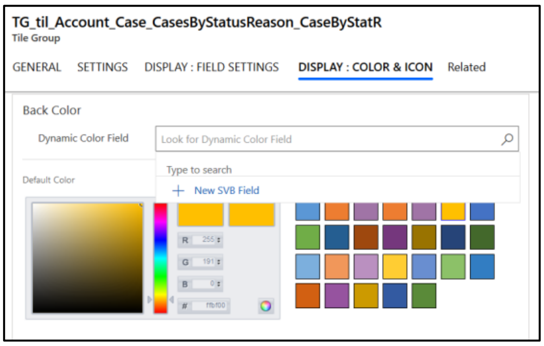
(a) **Title field**
This is what is considered as the most important field with the small/long fields. The field that you select will be visible in bold in the tile view.
In our example we use Case Title

11. Save what you have done in the **DISPLAY: FIELDS SETTINGS** -tab, and continue with the **DISPLAY: COLOR AND ICON** tab with sections tab and it' sections:

    (a) **Back color** section

    (b) **Icon** section

TILE GROUP - DISPLAY:COLOR&ICON-TAB - BACK COLOR SECTION

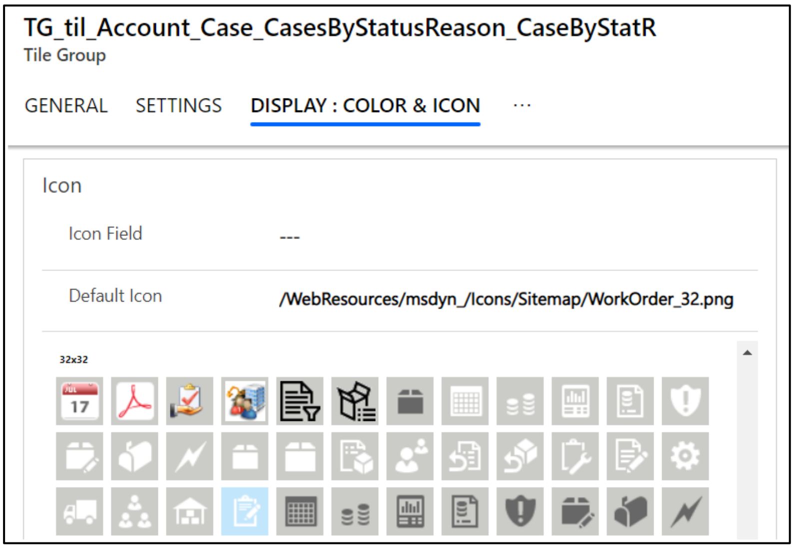


(a) **Dynamic color field**
If the fore color has to be dynamically determined, select one field on the contextual entity, containing the value

of the back color. The color format must be either a web color code (i.e.: #FFFFFF) or a color name (i.e.: red). We will deal later on with dynamic coloring
In our example we leave it blank.

(b) **Default color**
Background color of the icon.
When a dynamic color field is mentioned, the default back color is only used when no value has been found in the dynamic field.
You don't need to type the color code; simply click on the desired color in the color picker.
In our example we take whatever color we like.

TILE GROUP - DISPLAY:COLOR&ICON-TAB - ICON SECTION



(a) **Icon field**
If the icon has to be dynamically determined, select one field on the contextual entity, containing the value of the icon.
In our example we leave it blank.

(b) **Default icon**
Icon displayed in the tile. The URL can point to an internal web resource (relative path) or an external image (absolute URL).
When a dynamic icon field is mentioned, the default icon is only displayed when no value has been found in the dynamic field.
You don't need to type the URL : simply click on the desired image in the icon library below the URL. The library is showing all the icons with a 48x48 size, saved as web resources in your CRM.
In our example we take whatever icon we like.

12. Save the data of the tile group and go back to the **TILE VIEW**, where we need to fill data in the **SETTINGS**-tab which consist of two sections

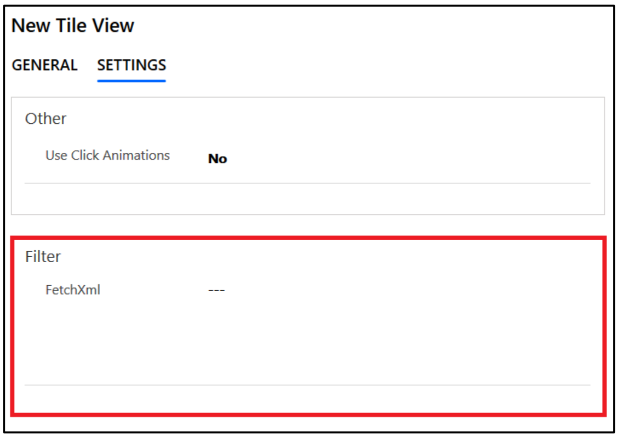
(a) **Other** section

(b) **Filter** section

TILE VIEW - SETTINGS TAB – OTHER SECTION

(a) **Use click animations**
If you click on the sides of a tile, youw ill see a small animation in the form of a kind of min popup of the tile.
In our example we use Yes

SETTINGS – TAB - FILTER SECTION



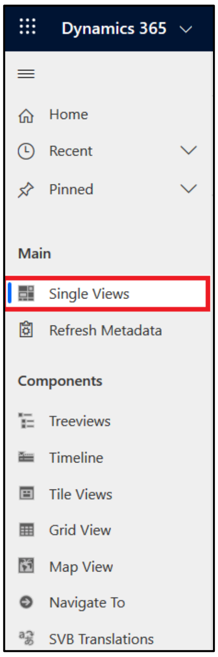
(a) **FetchXML**
Additional filter to apply on the query, written in FetchXML format. We will create another case to give you an example of that later in this tutorial (Practical case:" A tile view of tile type that shows you cases with an origin "web" for an account")
In our example we don't add anything in here

13. Save the modifications of the tile view

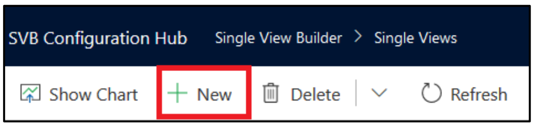
2.1.2.2 Create the single view component

If we want to see results of what we have done, we must first create a single view component to which we add later on our tile view component. These are the necessary steps to do that

1. Select **SINGLE VIEWS** from the sitemap.

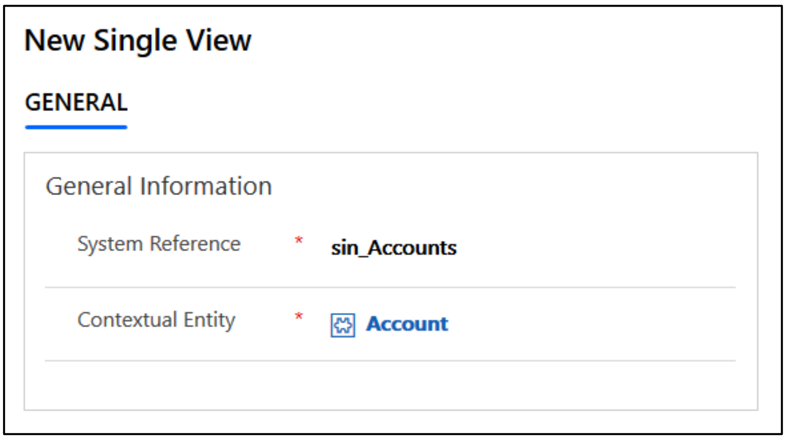


2. Click the **NEW**-button in the command bar



3. Fill in the data as explained here beneath:

SINGLE VIEW - GENERAL TAB - **GENERAL INFORMATION** SECTION



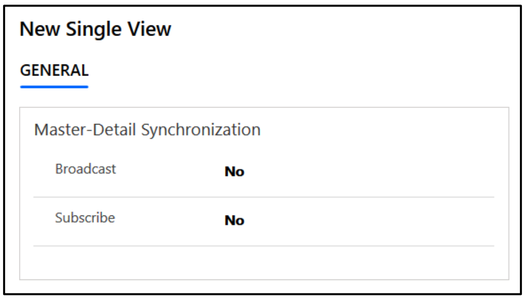
(a) **System Reference**
This is the name of the single view configuration. We advise you the following naming convention, which is a bit similar to the Hungarian naming convention that is often used when working with objects. The name has several parts, which will be separated by an underscore

A three character abbreviation to indicate the object type This will be placed as reference on the contextual entity form and used by the system to retrieve its configuration.	As we are creating a single view, we could use **sin**
The name of the contextual entity. This is the entity on which you will put the single view	In our example this will be the Account entity

The final result for our example is then sin_Accounts

(b) **Contextual entity**
This is the entity on which the single view will be placed.
In our example we use Account

SINGLE VIEW – GENERAL TAB - **MASTER-DETAIL SYNCHRONIZATION** SECTION



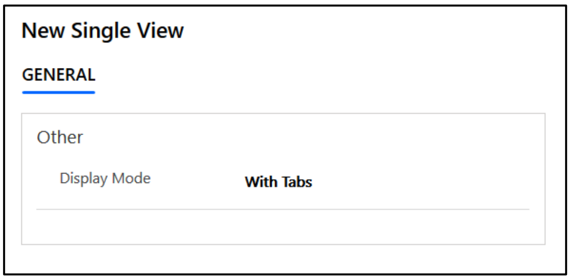
These two options are useful if you want to have two linked single views in one form. In such case, if you click on one record of the first single view, the second will refresh and display the information of the record focused on in the first one.
We will give an example of that when we are dealing with tree view and grid view components.

(a) **Broadcast**
The single view will be used as master single view on a form and when one element will be focused, trigger the refreshing of other single views (with subscribe option)

(b) **Subscribe**
The single view will listen to the others (with broadcast option) and refresh when needed.

In our example we put them both on No

SINGLE VIEW – GENERAL TAB - OTHER SECTION



(a) **Display mode**
Defines how the single view components should be accessed:

With tabs: tabs are displayed on top of the view area.

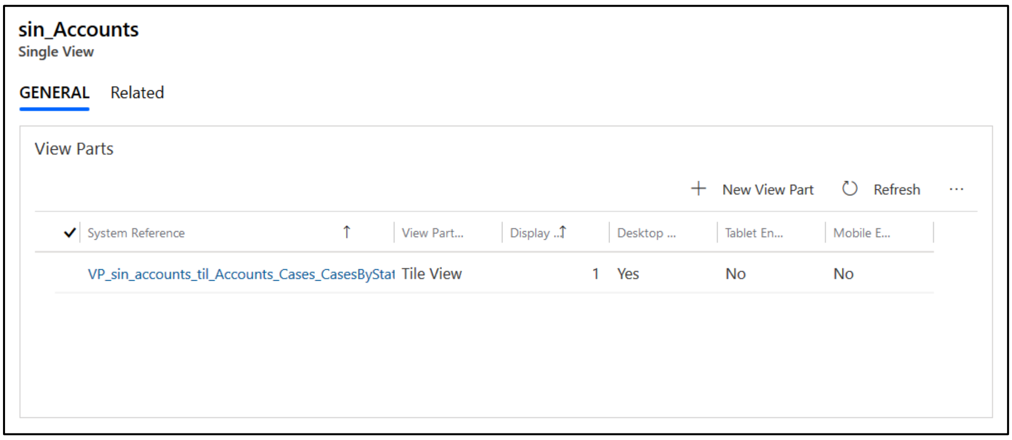
Without tabs: there are no tabs displayed and only the first view part is displayed.

In our example we use With Tabs

4. Save the modifications of the single view component

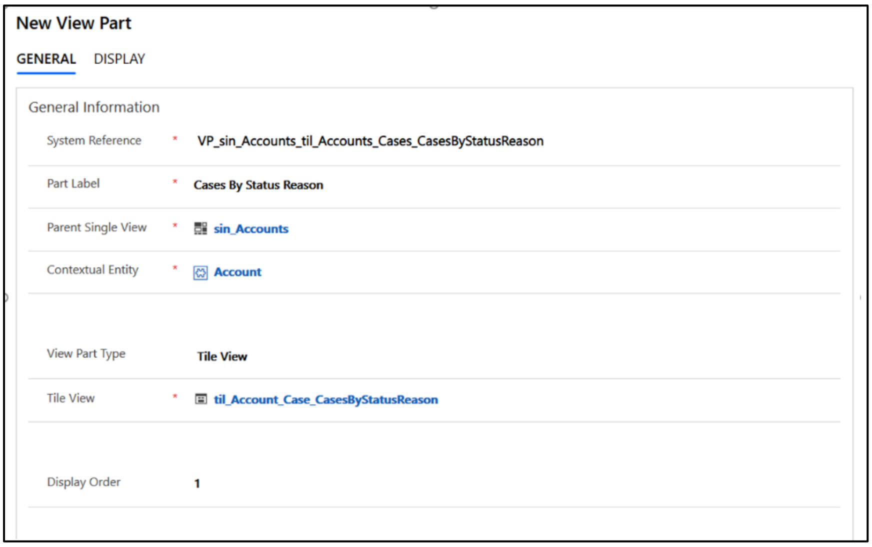
5. A single view can contain several other components such as tile views, tree views,…If you want to see those on the single view, you must add them as view parts on the single view. You can do this by clicking on the +-sign on the right side. If this is not available, then you have to save the single view first.

SINGLE VIEW – GENERAL TAB - VIEW PARTS SECTION



6. Clicking on the +-sign brings you in the screen of view parts, where you can define the following information:

VIEW PART – GENERAL TAB - **GENERAL INFORMATION** SECTION



(a) **System Reference**
This is the name of the view part configuration. We advise you the following naming convention, which is a bit similar to the Hungarian naming convention that is often used when working with objects. The name has several parts, which will be separated by an underscore

An abbreviation to indicate the object type	As we are creating a view part, we could use **VP**
The name of the single view on which the view part is placed.	In our example this will be **sin_Accounts**
The name of the tile view you want to add as view part to the single view	In our example this will be **til_Accounts_Cases_CasesByStatusReason**

The final result for our example is then
VP_sin_Accounts_til_Accounts_Cases_CasesByStatusReason

(b) **Part Label**
This is the label of the view part. If you have chosen to display the title (by asking “with tabs” in the **display mode** option of the single view), this will be used as label for the single view tab.
In our example we use Cases By Status Reason

(c) **Parent Single View**
Reference to the parent single view (= the single view on which you are adding the view part)
In our example it is sin_Accounts

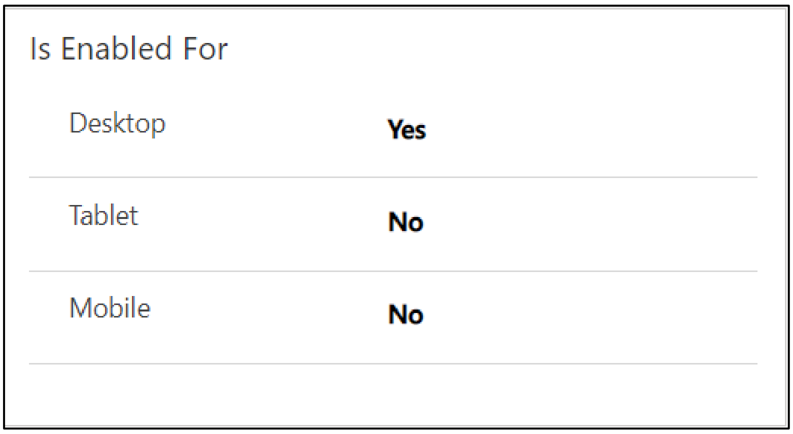
(d) **Contextual entity**
This is the entity on which view part of the single view will be placed
In our example it is Account

(e) **View part type**
Type of the view part : timeline, tile view, tree view, grid view, map view
In our example it is Tile View

(f) **Tile view**
Reference to the tile view that you want to use as view part
In our example it is til_Account_Case_CasesByStatusReason

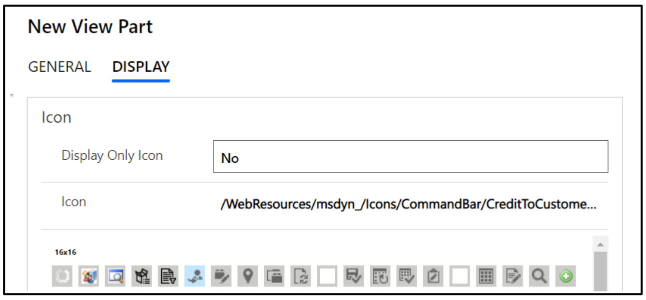
(g) **Display order**
If you are creating a single view with different tabs that represent the different view parts, this field will allow you to order them. In our example this is **1**

VIEW PART – GENERAL TAB – **IS ENABLED FOR** SECTION



You can uses these options to specify on which device you want the view part enabled. You have the coice between **desktop, tablet** and **mobile.** The **Desktop** option is by default enabled.

VIEW PART – DISPLAY TAB - ICON SECTION



(a) **Display only icon**
If you use tabs (by asking “with tabs” in **the display mode** option of the single view) and an icon, you can ask to show only the icon or the icon and label.
In our example we put this on No

(b) **Icon**
Icon displayed in the tabs of your single view.
The URL can point to an internal web resource (relative path) or an external image (absolute URL).
You don’t need to type the URL : simply click on the desired image in the icon library below the URL. The library is showing all icons with a 16x16 size, saved as web resources in your CRM.
In our example we use whatever icon we wish

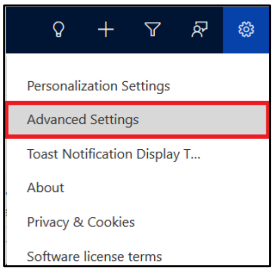
5. Save the modifications of the view part and go back to the single view component

6. Save the modifications of the single view component

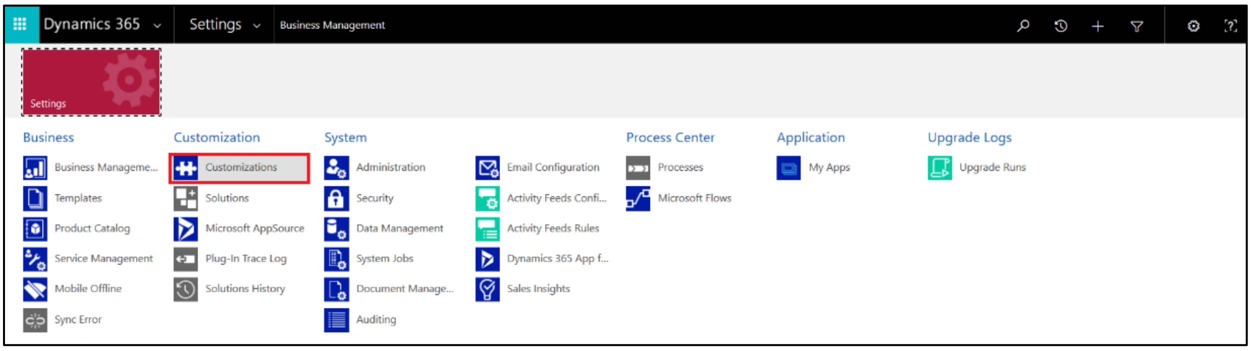
2.1.2.3 Create or modify the form to visualise the Single View Component (using PCF Controls)

We have now created a tile view component and added that one to a single view component. This last one must be visualised now on a form of accounts. We will create therefore a new form for accounts. We will not spend too much time explaining how to create the form, because this is general CRM-knowledge, but we have to explain you the steps for adding the single view component to the form

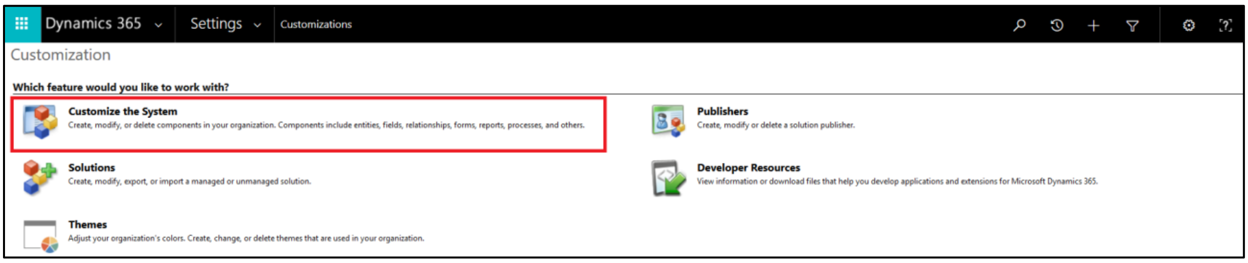
1. Go to the **SETTINGS** module:



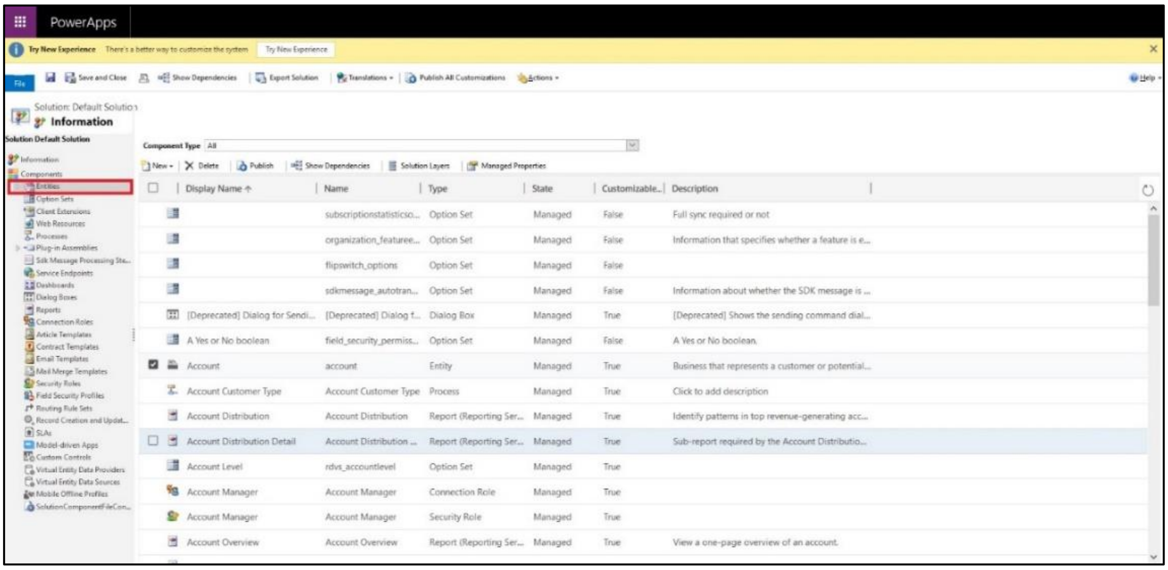
2. Open **CUSTOMIZATIONS**



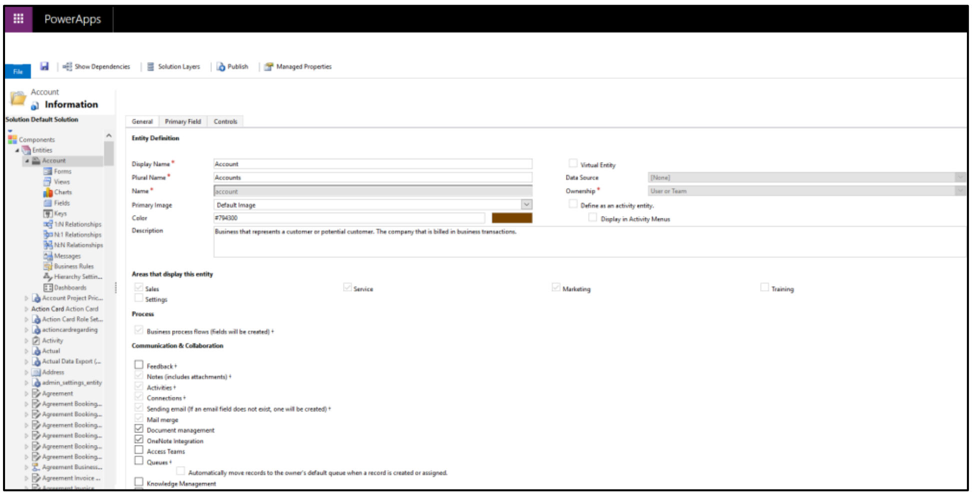
3. Select **CUSTOMIZE THE SYSTEM**



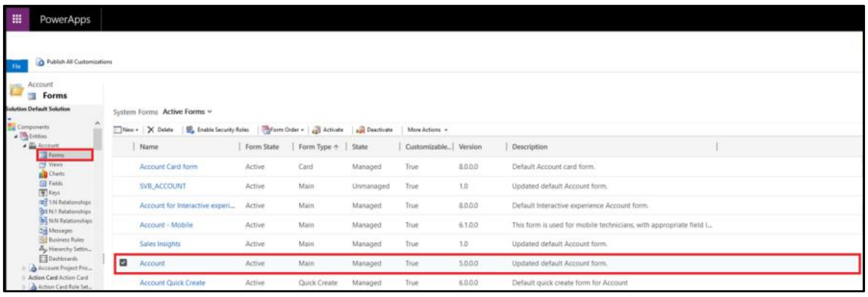
4. and you should have a similar screen like this one



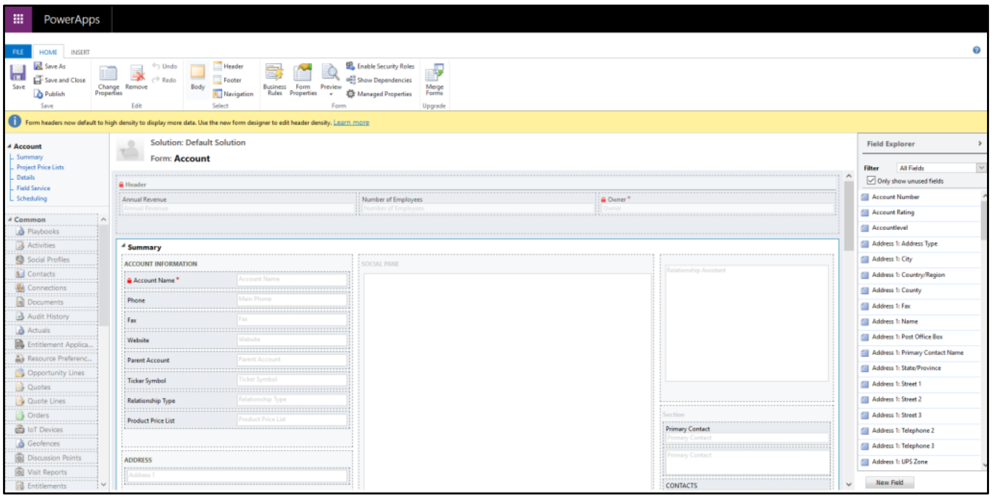
5. Go on the left side of the screen to expand the little arrow-icon  preceding **ENTITIES** and open in the same way the structure of the **ACCOUNT** entity and the screen will change as follows:



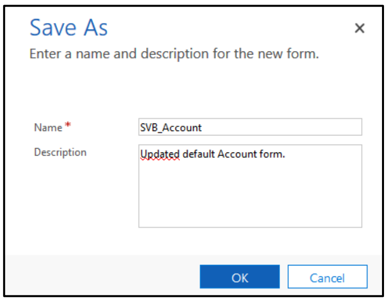
6. Select **FORMS** and the screen changes as follows



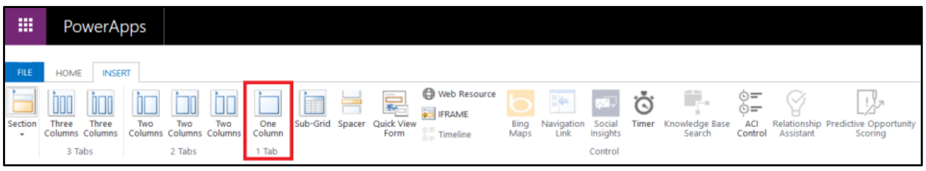
7. Open the **main account form** (the one on the blue line in previous screen), which shows you a similar screen like this one:



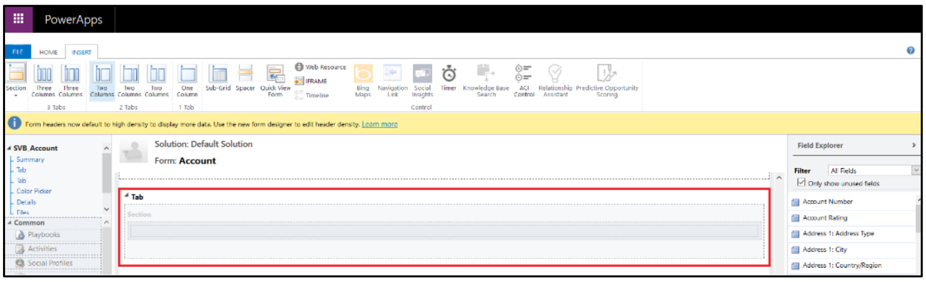
8. Click on **SAVE AS** in the ribbon and give a name to the form. **In our example we use SVB_Account** here



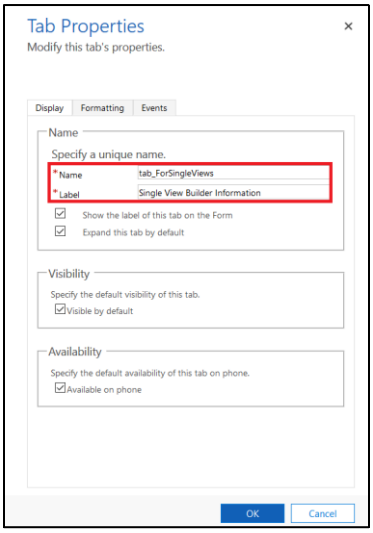
9. In the “INSERT”-Tab of the ribbon bar, select the “One Column” button



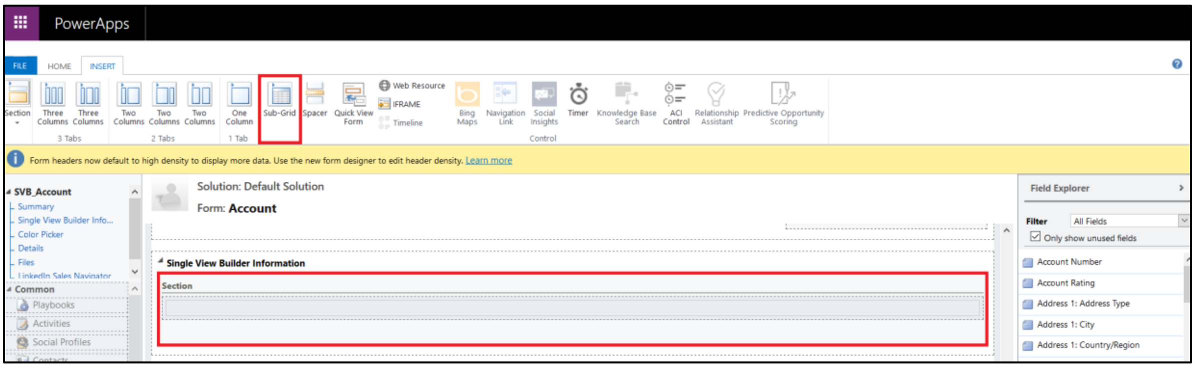
10. DoubleClick on the “One Column” Tab you inserted to change the properties:



11. Name it **tab_ForSingleViews** and set **Single View Builder Information** as Label:

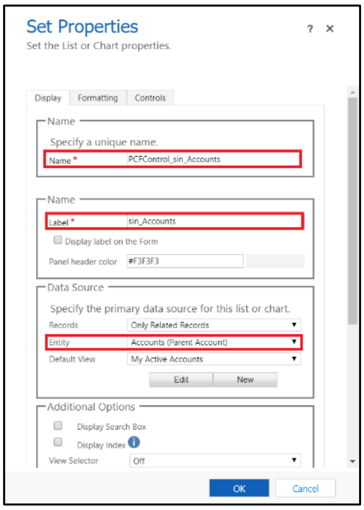


12. Click on the **Section** of that “One Column” and insert a **Sub-Grid** by going to the “INSERT”-TAB of the ribbon bar:



13. Modify the values as follows. We only describe those that are relevant for our example

DISPLAY-TAB



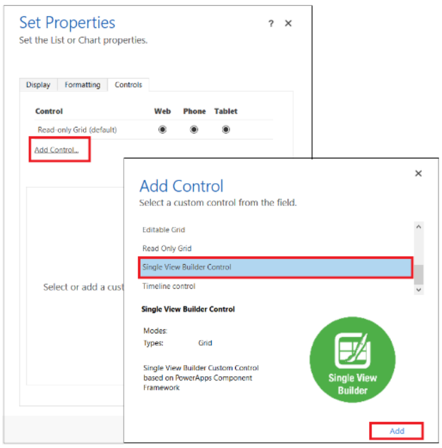
(a) **Name**
ID of the visual element. We always use the following structure: PCFControl_name of single view component.
In our example this is PCFControl_sin_Accounts

(b) **Label**
Name of the single view component
In our example this is sin_Accounts

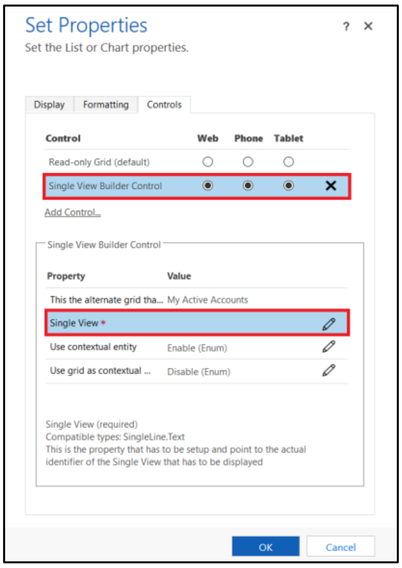
(C) **Entity**
Specify the primary data source
In our example this is Accounts (Parent Account)

CONTROLS-TAB:

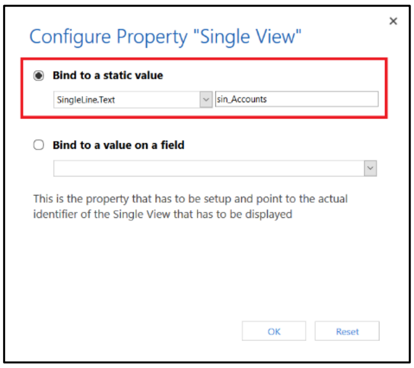
14. Click the **Add Control…**option from the Controls-Tab, select **Single View Builder** from the custom fields listed, and click **Add**



15. Set the Single View Builder Control for **Web, Phone and Tablet**, and select the pen-icon to configure the properties of the “Single View”



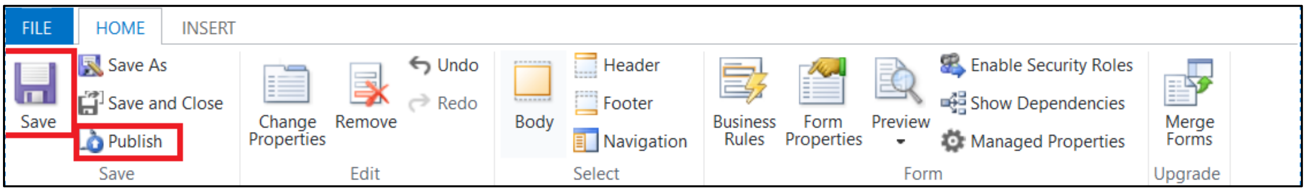
16. Select **Bind to a static value** and set to **sin_Accounts**



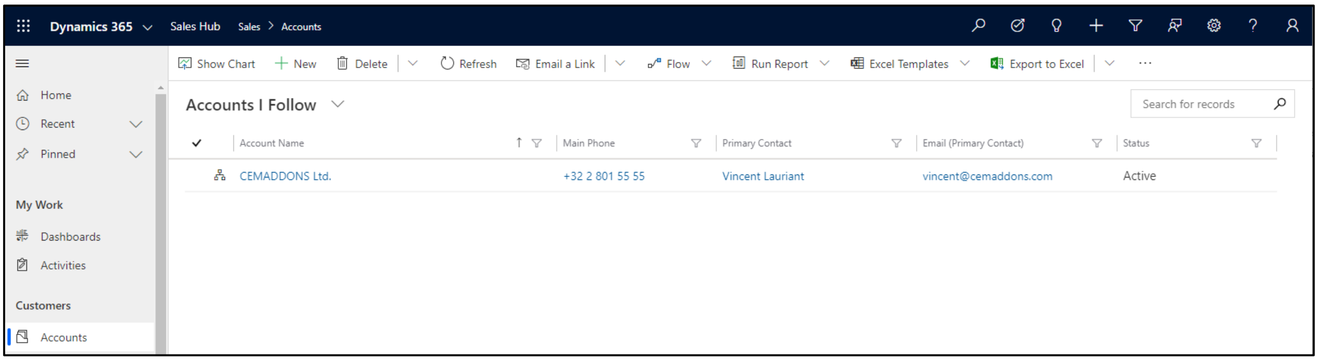
(a) **Bind to a static value**
This is the System Reference of your single view component that must be given to the Single View Builder custom control
In our example this is sin_Accounts

(b) **Bind to a value on a field**
One could also reuse the value of a field to make the setting the Single View Builder custom control dynamically configured
Not for our example here

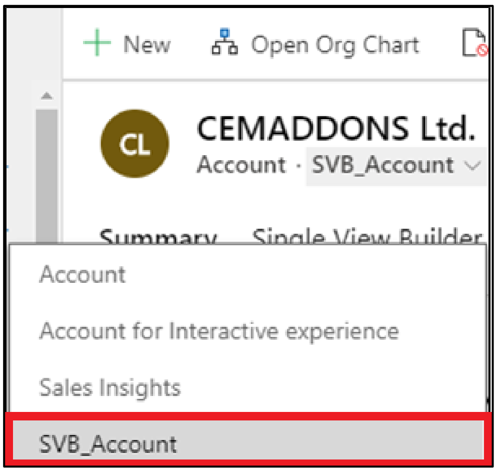
17. **Save** the form and **publish** it



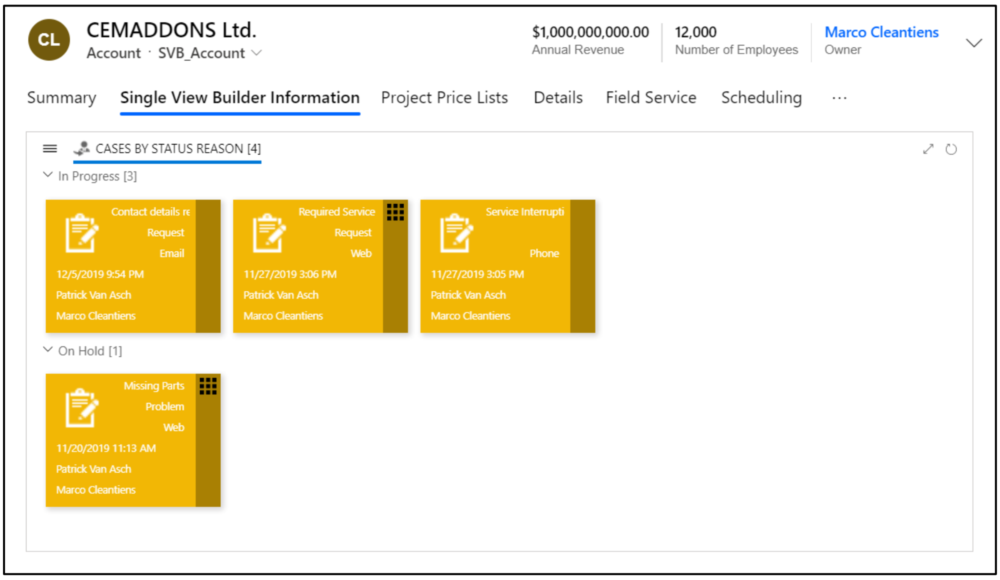
18. Go via the sitemap to **SALES** app**,** go to subarea **ACCOUNTS** and open an account that has cases available. **We take for our example the account CEMADDONS Ltd.**



19. Make sure you select the correct form, which is in our case the form **SVB_Account**. You can do this by going to the left side of the forms, just above the account name

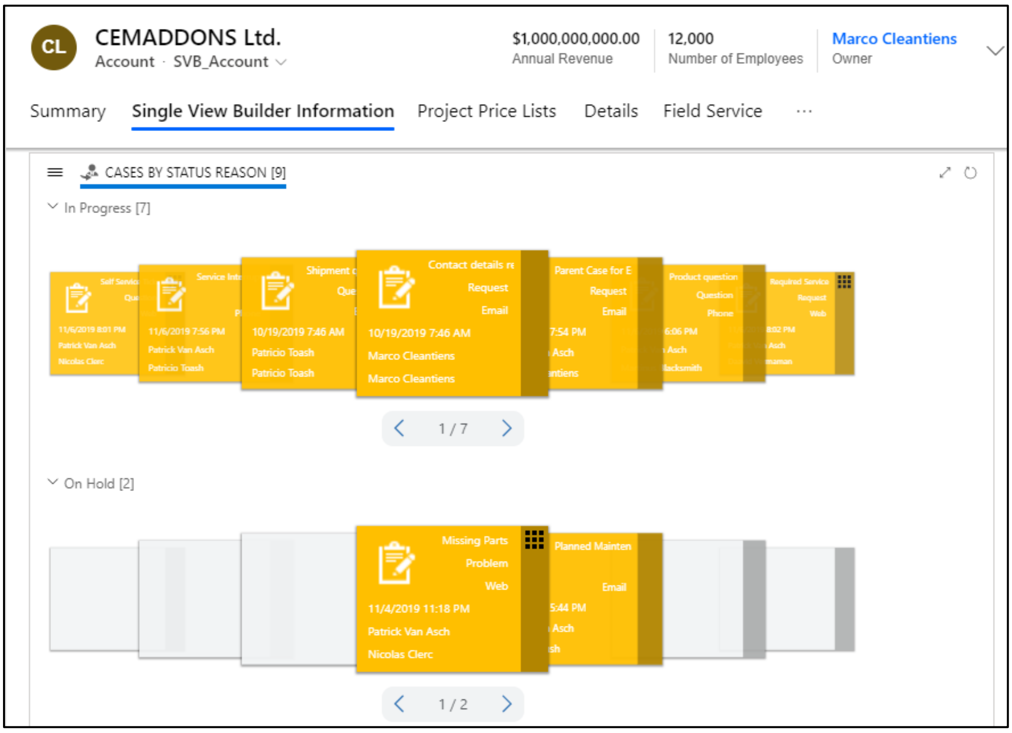


20. The single view component with the tile view on it should now be available. As the system needs to load it, you have to wait a few seconds before you see it. It should look like this:



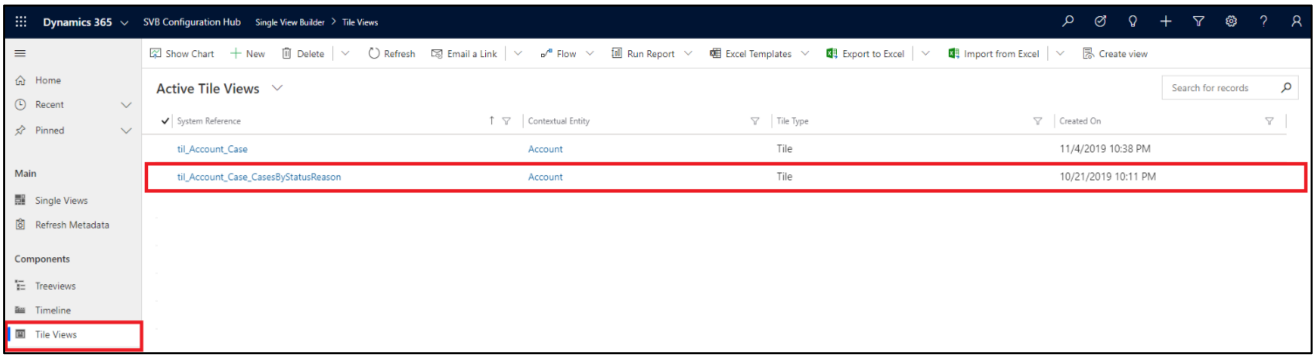
2.1.2.4 Modify the tile view style template to behave as carousel

Till now we have indicated the style template of the tile view in our above exercises. By defining the **Style Template** - either **Normal Tile** or **Carousel** - one can indicate the behaviour of the tile view. You can set this behaviour from the OTHER-Section of the GENERAL-TAB of your Tile Group configuration. If you do not specify explicitly the Style Template, it will default to Normal Tile Style. Next we will show how to configure the tile view style template to get the following desired result. Create some extra cases with different stages to show later on.

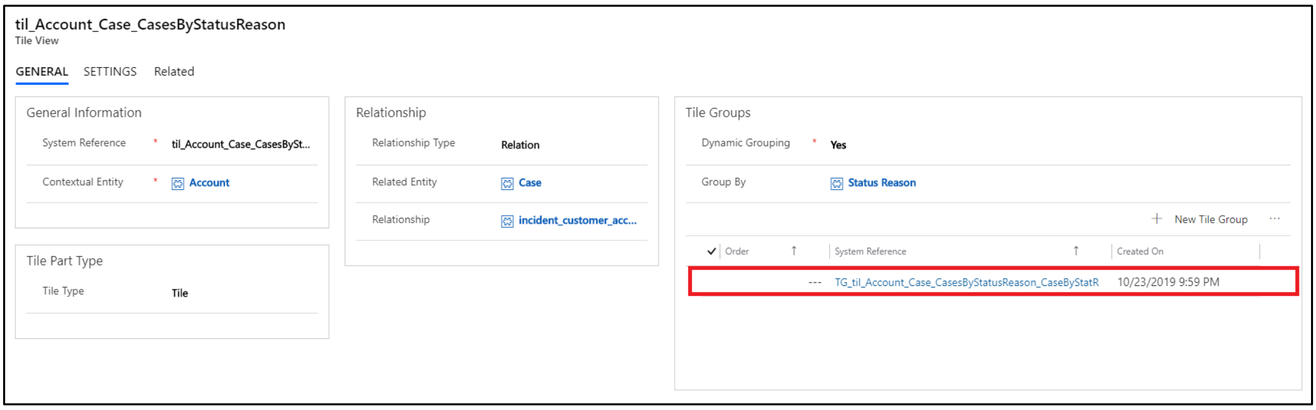


1. Navigate via the App Switcher to the **SVB Configuration Hub** App.

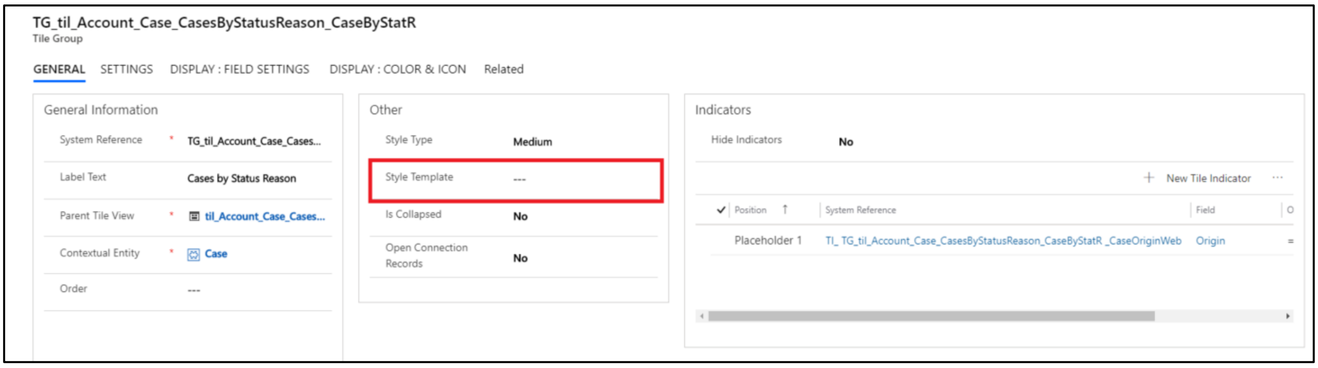
2. Select **TILE VIEWS** from the sitemap and open the tile view **til_Account_Case_CasesByStatusReason** you created earlier, and open



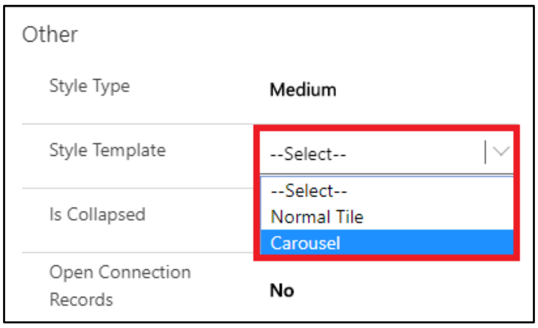
3. Open the Tile Group **TG_til_Account_Case_CasesByStatusReason_CaseByStatR**



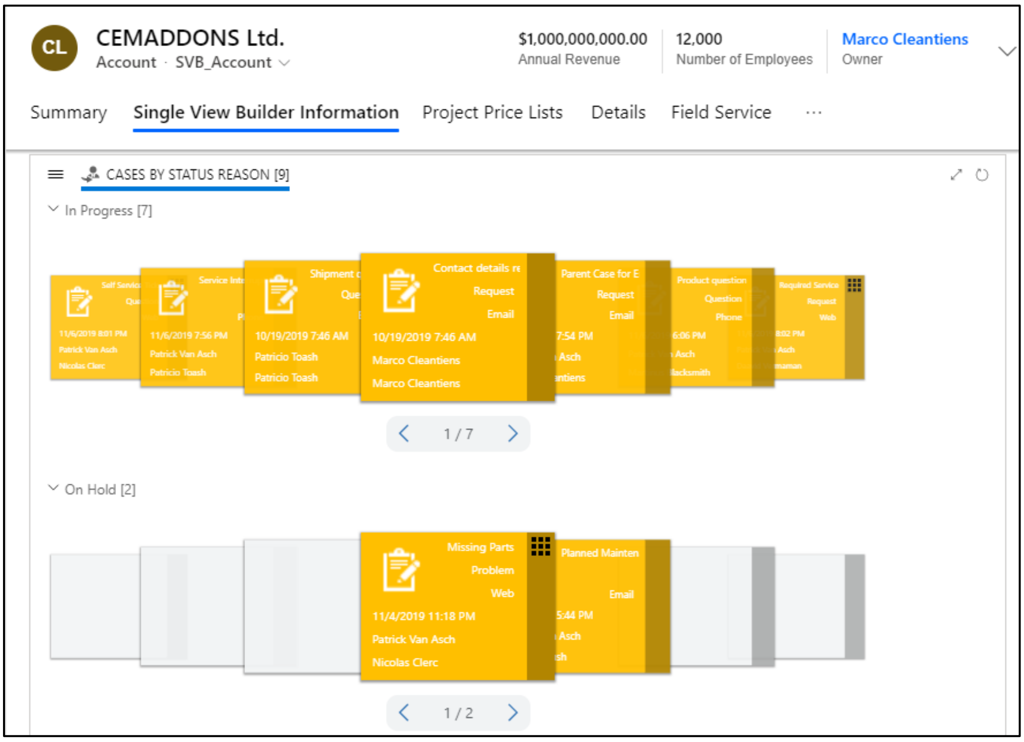
4. Navigate to the **OTHER**-section of the **GENERAL**-TAB to configure the **Style Template**:



5. Choose **Carousel** from the drop list to configure the tile view to behave as Carousel representation::



6. Save and verify the result: go to an account, select the right form SVB_Account. The cases shown dynamically grouped per status are now visualized in carousel style and should look like this:

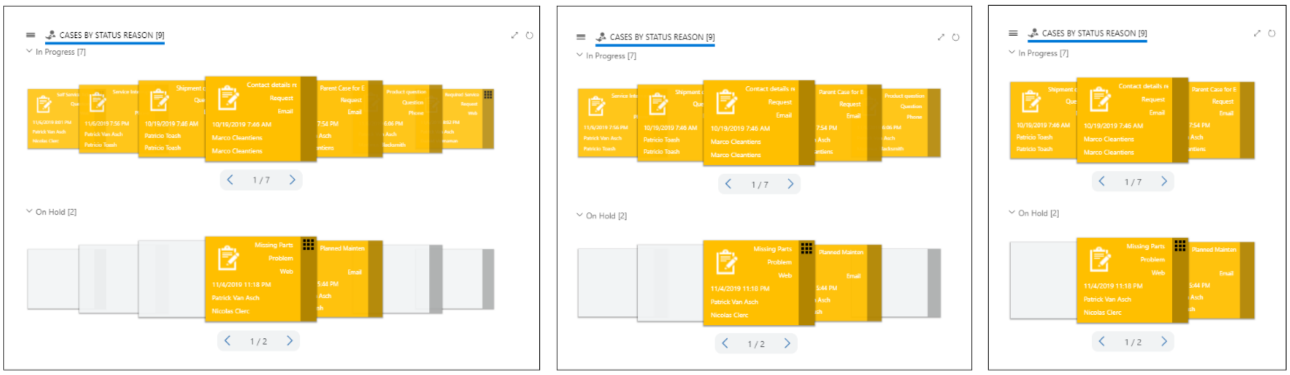


7. Check also the behavior we have built-in the carousel style template for the tile component:

   (a) By design using the carousel visualization will auto-operate as a circular infinite loop

   (b) use left/right arrow buttons  of the horizontal Carousel-slider to navigate through the tiles shown underneath the tile groups, the counter in the middle, will indicate which item you are focusing on in the center of the carousel and relatively to the total amount per tile group

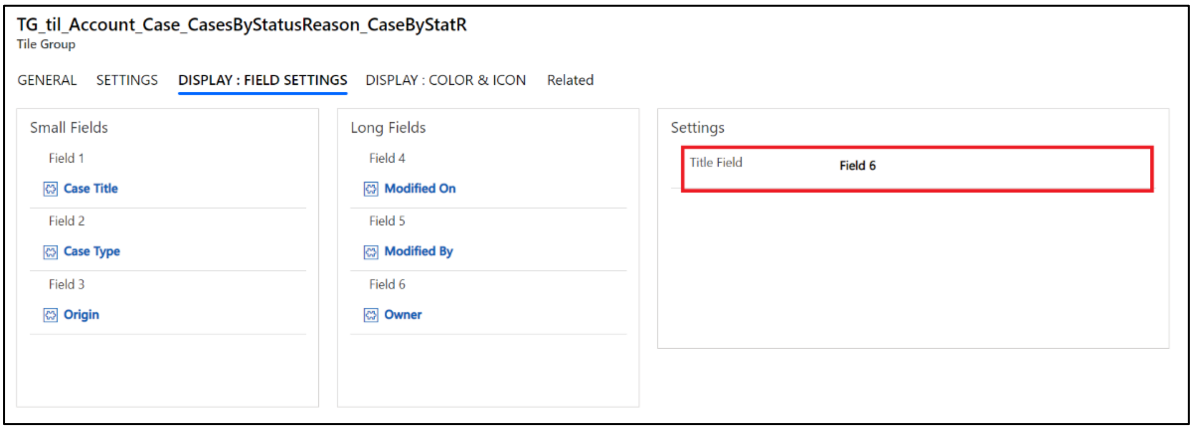
   (c) The carousel style template will behave fully responsive through an auto-scaling layout



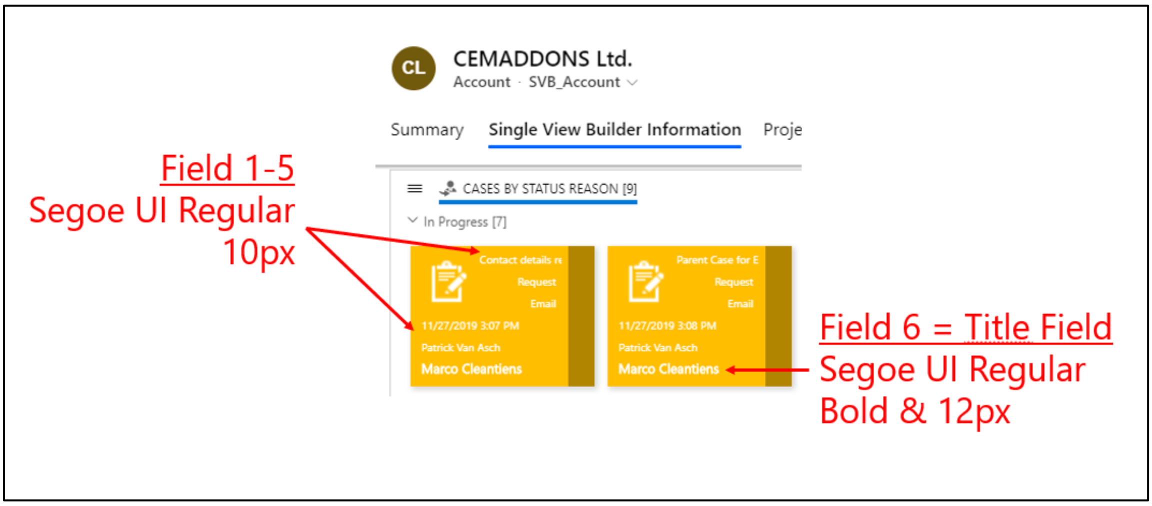
2.1.2.5 Highlight one of the Fields with larger font and displayed in bold

One of the six field you can show per tile can be highlighted slightly different from the others. By default all fields are shown with font *Segoe UI Regular 10px*. The field you choose to stand out from the other 5 -also referred to as *Title Field* – will be shown in a slightly larger and bold font *Segoe UI Regular 12px.*

1. Navigate via the App Switcher to the **SVB Configuration Hub** App.
2. Select **TILE VIEWS** from the sitemap
3. Open the active Tile View **til_Account_Case_CasesByStatusReason** you created earlier
4. Open the Tile Group **TG_til_Account_Case_CasesByStatusReason_CaseByStatR** and navigate to the **DISPLAY:FIELD SETTINGS-**TAB
5. In our example we want to highlight the *Owner*, thus choose **Field 6** in the **SETTINGS-SECTION** as **Title Field**:



6. Navigate to the **OTHER**-section in the **GENERAL**-tab and revert back the **Style Template** to **Normal Tile**
7. Verify the Title Field defined standing out



2.1.3 Web Browser and Mobile Device Support

2.1.3.1 Supported web browsers and mobile devices

Users can access the Dynamics 365 Customer Engagement Web application with the most recent versions of these popular browsers:

(a) Chrome

(b) Edge

(c) Internet Explorer

(d) Firefox

(e) Safari

Supported Devices for Mobile App

(a) iOS 11.4.1 or later (RAM: min 1Gb, recommended 2Gb)

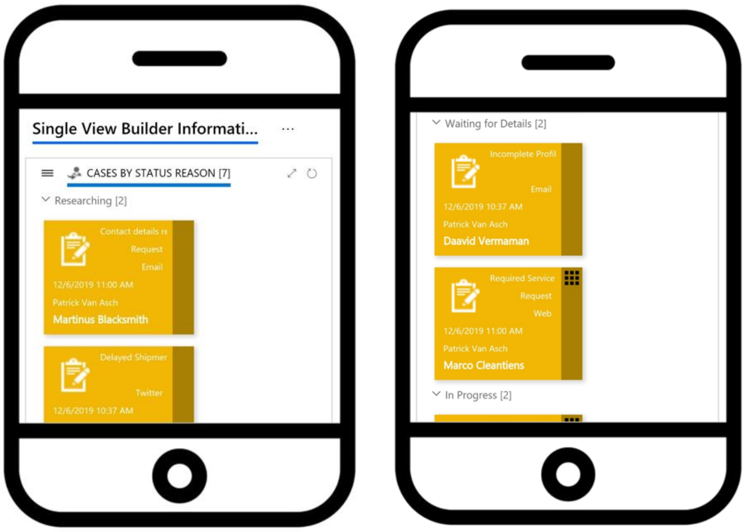
(b) Android 6.0.1 or later (RAM: min 2Gb, recommended 3Gb)

(c) Windows 10 (RAM: min 2Gb, recommended 4Gb)

For more detailed information about supported browsers and supported phones or tablets, see https://docs.microsoft.com/en-us/dynamics365/customerengagement/on-premises/admin/supported-web-browsers-and-mobile-devices.

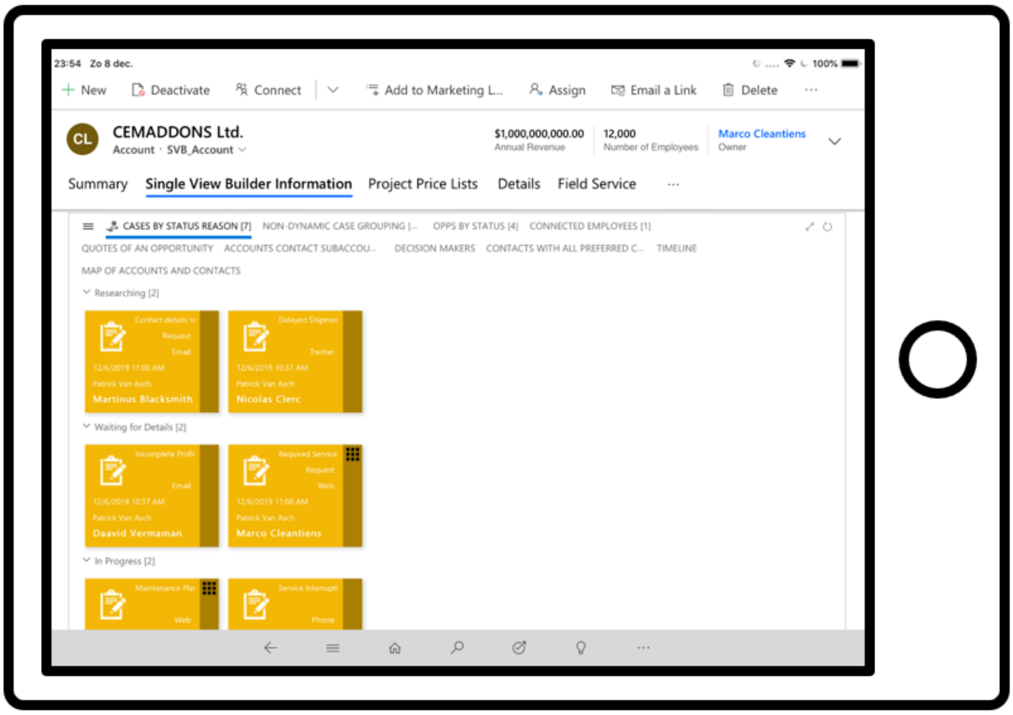
2.1.3.2 Smartphone experience with Dynamics 365 for Phones

The **PowerApps component framework** is the foundation for the new Unified Interface released with Dynamics 365 for Customer Engagement apps version 9.0 which uses responsive web design principles to provide an optimal viewing and interaction experience for any screen size, device, or orientation. **Single View Builder V2.1** fully leverages the capabilities of the Microsoft Power Platform and offers all its custom components as PCF controls next to its Web Resource variants, and with support for mobile devices. Example given using a **Microsoft Dynamics 365 for Phones** with **Android 8.0.0**



2.1.3.3 Tablet Experience with Dynamics 365 for Tablet

Example given using a **Microsoft Dynamics 365 13.19112.7es** with **iOS 12.4.3**



Chapter 6
Assistance

1. Getting assistance

If you need assistance in deploying Single View Builder or encounter any issues you cannot solve yourself, we recommend you to:

(1) contact the support department of your implementing SVB Partner or SVB Reseller in 1st Line Support

(2) contact our Realdolmen Service Desk for implementations by Gfi or Realdolmen or 2nd Line Support

1.1 Realdolmen Service Desk

To streamline all customer and partner support requests and give you a better support experience, we centralize all requests to be handled by our Realdolmen Service Desk. We have foreseen different channels to communicate with the Realdolmen Service Desk.
When asking for assistance and throughout the support process or when providing feedback, you will provide us with contact and identifiable information about yourself and your company. Please read the **Privacy Notice and Policy** (https://www.realdolmen.com/en/privacy-policy-statement) explaining how Realdolmen processes the personal data you submit through this website.

1.1.1 Mail

When using electronic mail **send us an email directly** at info.D365products@realdolmen.com. Please note the Realdolmen Service Desk will be available for you from Monday until Friday between 8.30 AM and 5 PM (Time zone UTC +1 i.e. Central European Time (CET)). In order to speed up the support process, please include the following information in your communication:

(1) Your partner name

(2) Contact details (company, name, email address, phone number)

(3) Product name: *Single View Builder*

(4) Product version

(5) Detailed description of the issue you're experiencing and how to reproduce the issue

(6) Any support logs

1.1.2 Phone

Please call the following phone number: +32 2 78 150 140 and provide the following information:

(1) Your partner name

(2) Contact details (company, name, email address, phone number)

(3) Product Name: *Single View Builder*

(4) Product version

(5) Describe the issue you're experiencing and how to reproduce the issue

The operator will log your inquiry in the system of the Realdolmen Service Desk. The "Microsoft Dynamcis365 CRM" team will contact you as soon as possible.

1.1.3 Website

We can provide you access to our online system, where you can create your own ticket. The advantage of this system is that you have an overview and can follow-up your tickets. If you want to have access, please ask for your credentials by sending an email to at info.D365products@realdolmen.com. The Realdolmen Service Desk will be available for you from Monday until Friday between 8.30 AM and 5 PM (Time zone UTC +1 i.e. Central European Time (CET)).

1.2 Feedback

We are also listening closely to feedback from our customers and partners and we'd like to know how our product and this documentation met your needs, as well as any ways that you would like to improve them.

You can quickly and easily provide your feedback per mail at infoCRM@realdolmen.com.

1.3 Contact us

For any other requests please contact us using the online form at https://www.cemaddons.com/contact.