

Role Based CustomView

Table of Contents

<u>Introduction</u>	<u>2</u>
<u>Working model.....</u>	<u>2</u>
<u>Current model:.....</u>	<u>2</u>
<u>New model:.....</u>	<u>2</u>
<u>Design Details.....</u>	<u>4</u>
<u>Developer's approach.....</u>	<u>4</u>
<u>Implementation Details.....</u>	<u>4</u>

Introduction

Role Based Custom view or Custom filters has been introduced for easier management of custom views and sharing custom views with other users.

Design Details

Current model:

Any Custom filter created by any user, is accessible to any other users.

Disadvantages -

1. As and when new filters are added, since all of them are accessible for all the users, it becomes hard to manage the filters and not easily usable.
2. Users cannot have any private custom filters which they would not like other users to view or use it.

New model:

Any user can create Custom filter. By default any created custom filter will be marked as 'Private'.

'Private' -> Only the user who created the custom filter and the users above this user in the role hierarchy can view this custom filter.

Users can mark the custom filter as 'Public' during custom filter creation or even after creation by editing the custom filter through a check box 'Set as Public'

'Public' -> Any user of the CRM irrespective of his position/role in the organization can view it. (Under 'Public' section)

Contacts > New Custom view

Details

*View Name:	<input type="text" value="My Private Contacts"/>	<input type="checkbox"/> Set as Default	<input type="checkbox"/> List in Metrics	<input checked="" type="checkbox"/> Set as Public
-------------	--	---	--	---

Choose Columns

Salutation	First Name	Last Name *	Assigned To *
None	None	None	None
None			

When a user marks a custom filter as 'Public', it doesn't directly become a public filter. Rather it has to be approved by the admin and only then is publicly available. Till then it will be in 'Pending' state.

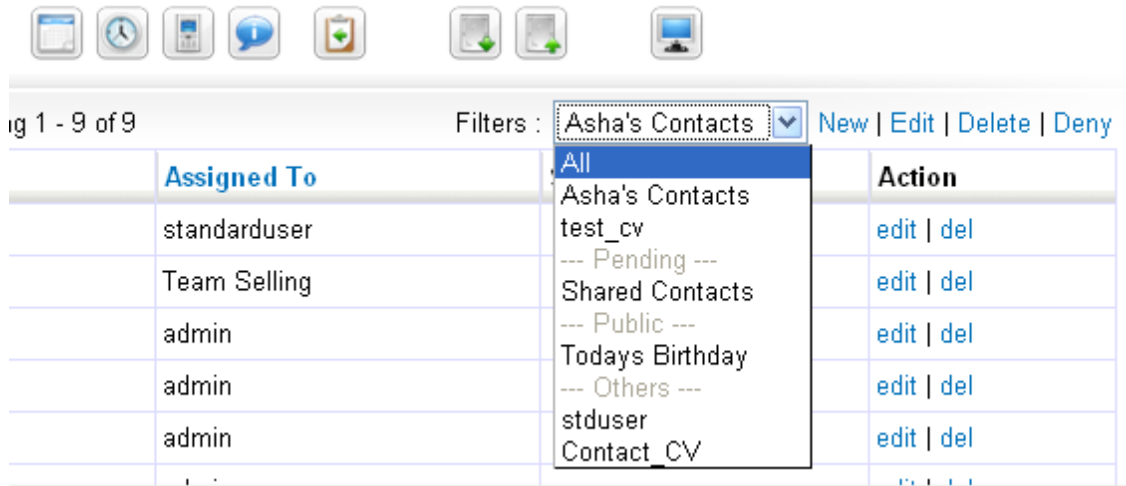
'Pending' -> Custom filter is marked as 'Public' by the created user, but waiting for approval from admin. This will be shown for admin user under 'Pending' section.

Once a custom filter is approved by the admin, it will be publicly available for all users. Admin can still go ahead and deny the access by which the custom filter will no more be publicly available and will enter 'Pending' state again.

Usage

Admin users can see following types of Custom views (refer to the snapshot) :-

1. Default or those created by the logged in user
2. Those which are created by Other users and set as public, but waiting for approval (Under 'Pending' section)
3. Those which are created by Other users and set as public and also approved to be public (Under 'Public' section)
4. Those which are private but created by the users who are under the logged in user in role hierarchy of the organization. (Under the section 'Others')



The screenshot shows a CRM interface with a table of contacts. The table has columns for 'Assigned To' and 'Action'. A dropdown menu is open over the 'Filters' field, showing various filter options.

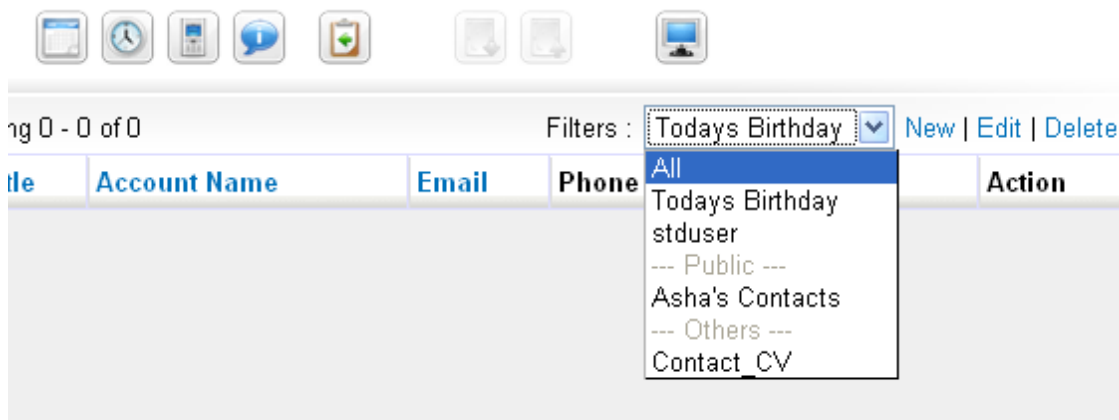
Assigned To	Action
standarduser	edit del
Team Selling	edit del
admin	edit del
admin	edit del
admin	edit del

Filters: Asha's Contacts

- All
- Asha's Contacts
- test_cv
- Pending ---
- Shared Contacts
- Public ---
- Today's Birthday
- Others ---
- stduser
- Contact_CV

Non-Admin users can see the following types of Custom views (refer to the snapshot) :-

1. Default or those created by the logged in user
2. Those which are created by Other users and set as public and also approved to be public by admin user (Under 'Public' section)
3. Those which are set as private but created by the users who are under the logged in user in role hierarchy of the organization. (Under the section 'Others')



The screenshot shows a CRM interface with a table of accounts. The table has columns for 'Account Name', 'Email', 'Phone', and 'Action'. A dropdown menu is open over the 'Filters' field, showing various filter options.

Account Name	Email	Phone	Action
--------------	-------	-------	--------

Filters: Today's Birthday

- All
- Today's Birthday
- stduser
- Public ---
- Asha's Contacts
- Others ---
- Contact_CV

Implementation Details

A new column called “status” needs to be included in 'vtiger_customview' table which holds the states of a custom view. This column should be an integer type of size 1. The possible values for this column are 0, 1, 2 and 3. The default value for 'status' will be 0. The value 1 for status means that the custom view is in 'Private' state. The values 2 and 3 are for the states 'Pending' and 'Public' respectively.

Also, add a column in 'vtiger_customview' called “userid” which holds the id of the user who created the current custom view.

File Changes:

include/js/ListView.js	Added ChangeCustomViewStatus() api to handle approval of public custom view's by Admin
Include/language/en_us.lang.php modules/CustomView/language/en_us.lang.php	Added required language strings
Include/Utils/Utils.php	Included the library RecyclebinUtils.php. Also added the definition for the Recyclebin status flags.
modules/Accounts/ListView.php modules/Calendar/ListView.php modules/Contacts/ListView.php modules/Campaigns/ListView.php modules/Campaigns/DetailView.php modules/Campaigns/CallRelatedList.php modules/Faq/ListView .php modules/HelpDesk/ListView .php modules/Invoice/Invoice.php modules/Leads/ListView .php modules/Notes/ListView .php modules/Potentials/ListView .php modules/PriceBooks/ListView .php modules/Products/ListView .php modules/PurchaseOrder/ListView .php modules/Quotes/ListView .php modules/SalesOrder/ListView .php modules/Vendors/ListView .php	Added a necessary security check for Custom views (Visibility, Edit and Delete options).
modules/CustomView/ChangeStatus.php	Added this new file to handle the action when Admin Approves/Deny's a User's request to make a Customview as public
modules/CustomView/CustomView.php	Changed the following api's to handle the security check while picking or viewing Custom views. getViewId(), getCustomViewByCvid(), getCustomViewCombo(), Added two new api's isPermittedCustomView() and isPermittedChangeStatus() to check the user permissions for respective actions.
module/CustomView/CustomViewAjax.php	Created the file as a gateway for Ajax request

	in CustomView module
modules/CustomView/EditView.php	Added necessary security checks (For Edit and Delete permission of Custom View) and also added a flag to mark the CustomView as public.
modules/CustomView/ListViewTop.php	Added the necessary security check while listing the customview's on homepage
modules/CustomView/Save.php	Taking care of status and userid field of the CustomView to enable security and support role based access.
Fresh Installation :- schema/DatabaseSchema.xml modules/CustomView/PopulateCustomView.php Migration:- modules/Migration/DBChanges/504_to_510.php	<ul style="list-style-type: none"> • Added two fields userid and status in vtiger_customview table • Updated the default entries for demo customview's to have proper status and userid values. • Handled the column addition in vtiger_customview table along with proper updates to userid and status field for all existing custom views in the CRM during migration.
Smarty/templates/CustomView.tpl	Added the necessary place holder to make a CustomView as public
Smarty/templates/ListViewEntries.tpl	Added necessary checks and options to Edit and Delete the CustomView based on the permissions. Also added the options to Approve/Deny Customview for an admin user.