

Make a Field Read Only in SuiteCRM (uneditable)

You can make it so users cannot edit records by using SecuritySuite, but if you want to make it so they cannot edit specific fields:

You can make it so users can't edit certain fields by going to :
/custom/modules/<NAMEOFMODULE>/metadata/editviewdefs.php In this file, add 'type'=>'readonly', to the definitions of the file you want to make readonly.

i.e:

```
array (
    'name' => 'testdropdown_c',
    'studio' => 'visible',
    'label' => 'LBL_TESTDROPDOWN',
),
```

Would become:

```
array (
    'name' => 'testdropdown_c',
    'type'=>'readonly',
    'studio' => 'visible',
    'label' => 'LBL_TESTDROPDOWN',
),
```

Afterwards, run a Quick Repair & Rebuild and clear your Browser's cache to view changes.

you have to turn off inline edit or they'll be able to edit inline

Make Field Read Only Based on Role

Step 1:

As it is SugarCRM's module, copy modules/Contacts/views/view.edit.php to custom/modules/Contacts/views/view.edit.php or edit if it already exists at custom/modules/Contacts/views/view.edit.php

In function display() add following piece.

```
function display(){ // add this if function display doesn't exist
    global $current_user;
    // check if current user is in specific role
```

```
// code taken from thread
$IS_AM = in_array("<ROLE NAME>",
ACLRole::getUserRoleNames($current_user->id));
if($IS_AM)
    $this->ev->ss->assign('readOnly', 'readonly =
"readonly"');
else
    $this->ev->ss->assign('readOnly', '');
parent::display(); // add this if function display doesn't
exist
} // add this if function display doesn't exist
```

Basically this sets a value for “readonly” variable which can be called in the edit view.

Step 2:

Open custom/modules/Contacts/metadata/editviewdefs.php, if doesn't exist, go to Studio > Contacts > Layouts > Edit View and press Save and Deploy.

Give custom code to the field, phone_work,

```
'customCode' => '<input type="text" class="phone" tabindex="0" title=""
value="{ $fields.phone_work.value}" maxlength="100" size="30" id="phone_work"
name="phone_work" { $readOnly}>',
```

This just basically sets the <input> html value to readonly. So it is not editable on the screen.

Repeat step 2, for Quick Create.

Do Quick Repair and rebuild! See it working smoothly, every time!!!

If you wish this to happen only while editing a record, add a condition to

```
if($IS_AM && !empty($this->bean->id))
```

Note: disable inline edit or they will be able to edit. Also, this just disables editing in the HTML via “readonly” if someone removes this value from the HTML they can edit the field.

Make Field Read Only Based on Role (and hide it)

custom/modules/Contacts/views/view.edit.php

```
global $current_user;
```

```

        // check if current user is in specific role
        // code taken from thread
        $IS_AM = in_array("Agent",
ACLRole::getUserRoleNames($current_user->id));
        if($IS_AM){
            $this->ev->ss->assign('readOnly', 'readonly =
"readonly"');
            $this->ev->ss->assign('disabled', 'display:none;');
        }else{
            $this->ev->ss->assign('readOnly', '');
            $this->ev->ss->assign('disabled', '');
        }
        parent::display();

```

custom/modules/Contacts/metadata/editviewdefs.php

```

array (
    'name' => 'phone_work',
    'comment' => 'Work phone number of the contact',
    'label' => 'LBL_OFFICE_PHONE',
    'customCode' => '<input type="text" class="phone" tabindex="0"
title="" value="{ $fields.phone_work.value}" maxlength="99" size="30"
id="phone_work" style="{ $disabled}" name="phone_work" { $readOnly}>',
    ),

```

Hide the Value Completely and Make it Uneditable by Role:

You could also assign a variable in view.detail.php using the same method above like:

```

global $current_user;
        // check if current user is in specific role
        // code taken from thread
        $IS_AM = in_array("Agent",
ACLRole::getUserRoleNames($current_user->id));
        $fieldvalue= $this->bean->phone_work;
        if($IS_AM){
            $this->ev->ss->assign('readOnly', 'readonly =
"readonly"');
            $this->ev->ss->assign('blankView', '');

        }else{
            $this->ev->ss->assign('readOnly', '');
            $this->ev->ss->assign('blankView', $fieldvalue);
        }
        parent::display();

```

in detailviews.php you could do something like....

```
'customCode' => '<input type="text" class="phone" tabindex="0" title=""  
value="{ $blankView}" maxlength="99" size="30" id="phone_work"  
"name="phone_work" { $readOnly}>',
```

that should hide it completely on the front end including the value. No one can see it on the front end even if you try to inspect the field. You'd also have to do this for **detailview** and **quickcreate** add to hide it there too to make sure it's hidden everywhere.

References

<https://community.suitecrm.com/t/hide-field-based-on-role-change-data-based-on-role/16454/2>

<https://community.suitecrm.com/t/how-to-make-user-edit-field-readonly-in-suitecrm/66991/6>

<https://stackoverflow.com/questions/18585612/make-custom-field-readonly-based-on-user-role>

<http://urdhva-tech.blogspot.com/2013/08/readonly-field-conditionally.html>

From:

<https://igo.myqnapcloud.com:8085/dokuwiki/> - iGoWIKI

Permanent link:

https://igo.myqnapcloud.com:8085/dokuwiki/doku.php?id=production:sugarcrm:read_only_fields



Last update: **2021/10/15 05:38**