

SDK

- [Installing a new signature field](#)

Installing a new signature field

General premises

During the steps you can replace the names of the classes and the names of the files and/or the relative paths

If during the steps you change the names of the classes or the names of the files and/or the relative paths remember to change them also in the scripts that we are going to create and in the installation script `InstallSignatureField.php` (last step)

You can install a single signature field per module

The installation is valid for vtenext versions **18.05, 18.12, 19.10**

Installation

- Create new file `TouchCustom.php` in `modules/SDK/src/MyCustomizations/modules/Touch` folder. Copy&paste the following code in the file:

```
<?php

require_once('modules/Touch/Touch.php');

class TouchCustom extends Touch { // You can replace "Custom" with whatever you want
[]
[]public function __construct() {
[]parent::__construct();
[]$this->webservices['GetRecord'] = array('file' => 'overrides/TouchGetRecordCustom.php', 'class' =>
'TouchGetRecordCustom'); // You can replace "Custom" with whatever you want
[]$this->webservices['SaveRecord'] = array('file' => 'overrides/TouchSaveRecordCustom.php', 'class' =>
```

```
'TouchSaveRecordCustom'); // You can replace "Custom" with whatever you want
```

```
    }  
    }  
}
```

- Create new file `TouchGetRecordCustom.php` in `modules/Touch/vtws2/overrides` folder.
Copy&paste the following code in the file:

```
<?php
```

```
require_once('modules/Touch/vtws2/TouchGetRecord.php');
```

```
class TouchGetRecordCustom extends TouchGetRecord {
```

```
      
    function process(&$request) {  
        $module = $request['module'];  
        $record = parent::process($request);  
          
        if ($record && $record['success'] && $recordid == 0) {  
            if (in_array($module, array('ModuleCustom1', 'ModuleCustom2', 'ModuleCustomN'))) { // ModuleCustom* are the  
modules where you want to enable the signature  
                $no_image_lbl = getTranslatedString('NO_SIGNATURE_IMAGE', $module);  
                if (isset($record['signature']) && $record['signature'] == $no_image_lbl) {  
                    $record['signature'] = '';  
                }  
            }  
        }  
        $record['success'] = true;  
        return $record;  
    }  
      
    function retrieveRecord($module, $recordid, $prodblock = 0) {  
        $record = parent::retrieveRecord($module, $recordid, $prodblock);  
          
        if ($recordid > 0) {  
            if (in_array($module, array('ModuleCustom1', 'ModuleCustom2', 'ModuleCustomN'))) { // ModuleCustom* are the  
modules where you want to enable the signature  
                $no_image_lbl = getTranslatedString('NO_SIGNATURE_IMAGE', $module);  
                if (!empty($record['signature']) && $record['signature'] != $no_image_lbl) {  
                    $img_path = $record['signature'];  
                }  
            }  
        }  
    }  
}
```

```

        $type = pathinfo($img_path, PATHINFO_EXTENSION);
        $data = file_get_contents($img_path);
        $base64 = 'data:image/' . $type . ';base64,' . base64_encode($data);
        $record['signature'] = $base64;
    } else {
        $record['signature'] = '';
    }
}
}
}
}
return $record;
}
}
}
}

```

- Create new file `TouchSaveRecordCustom.php` in `modules/Touch/vtws2/overrides` folder.
Copy&paste the following code in the file:

```

<?php

require_once('modules/Touch/vtws2/TouchSaveRecord.php');

class TouchSaveRecordCustom extends TouchSaveRecord {
}

public function preSaveRecord($module, $crmid, $mode, &$values) {
    parent::preSaveRecord($module, $crmid, $mode, $values);
}

if (in_array($module, array('ModuleCustom1', 'ModuleCustom2', 'ModuleCustomN'))) { // ModuleCustom* are the
modules where you want to enable the signature
    if (isset($values['signature'])) {
        $path = 'storage/signatures/';
        if (!file_exists($path)) {
            mkdir($path);
            chmod($path, 0755);
        }
        $img = $values['signature'];
        $img = str_replace('data:image/png;base64,', '', $img);
        $img = str_replace(' ', '+', $img);
        $data = base64_decode($img);
        $file = $path . 'signature_' . md5($crmid) . '.png';
        $success = file_put_contents($file, $data);
    }
}
}
}

```

```
if ($success) {  
    $values['signature'] = $file;  
  
}  
  
}  
  
}  
  
}
```

- Create new file `Custom.js` in `modules/Touch/overrides`. The app will automatically download it to extend/overwrite the default behavior. Copy&paste the following code in the file:

```
(function() {  
  if (Vtecrm && Vtecrm.app) {  
    Ext.define("Vtecrm.view.ShowRecordCustom", { // You can replace "Custom" with whatever you want  
      override: "Vtecrm.view.ShowRecord",  
      signatureModules: ["HelpDesk", "ModuleCustom1", "ModuleCustom2", "ModuleCustomN"], // ModuleCustom* a  
        the modules where you want to enable the signature  
    });  
  }  
})();
```

If you already have app customizations in a file, you just need to add the class definition (`Vtecrm.view.ShowRecordCustom`).

If you have already extended/overwritten the class `Vtcrm.view.ShowRecord` you just need to overwrite the variable `signatureModules` adding the modules where you want to enable the signature

- Create new file `InstallSignatureField.php` (or whatever you want) in `plugins/script` folder and copy&paste the following code:

```
<?php
die(); // Uncomment this to run the script

require('.././config.inc.php');
chdir($root_directory);
require_once('include/utills/utills.php');
require_once('vtlib/Vtecrm/Module.php');
$Vtiger_Utills_Log = true;
```

```

global $adb, $table_prefix;
VteSession::start(); // crmv@128133

SDK::setClass('Touch', 'TouchCustom', 'modules/SDK/src/MyCustomizations/modules/Touch/TouchCustom.php');
// You can replace "Custom" with whatever you

$moduleName = 'ModuleCustom'; // ModuleCustom is the module where you want to enable the signature

$signatureBlock = new Vtiger_Block();
$signatureBlock->label = 'LBL_SIGNATURE_BLOCK';
$moduleInstance = Vtiger_Module::getInstance($moduleName);
$moduleInstance->addBlock($signatureBlock);

$fields = array();
$fields[] = array(
    ['module' => $moduleName,
    ['block' => 'LBL_SIGNATURE_BLOCK',
    ['name' => 'signature',
    ['label' => 'signature',
    ['uitype' => '1016',
    ['columnntype' => 'VARCHAR(255)',
    ['typeofdata' => 'V~O'
    );
include('modules/SDK/examples/fieldCreate.php');

SDK::setLanguageEntries($moduleName, 'LBL_SIGNATURE_BLOCK', array('it_it' => 'Signature Information',
'en_us' => 'Signature Information'));
SDK::setLanguageEntries($moduleName, 'signature', array('it_it' => 'Firma', 'en_us' => 'Signature'));
SDK::setLanguageEntries($moduleName, 'NO_SIGNATURE_IMAGE', array('it_it' => 'Nessuna Firma', 'en_us' =>
'No Signature'));

```

- Execute the file `InstallSignatureField.php` and logout/login from vtenext on both the web and app side

You can run the script in the browser or through the CLI

```
cd plugins/script && php -f InstallSignatureField.php > InstallSignatureField.log
```

Screenshots

