Custom Uitypes

You can add new types to the existing ones and manage them completely without changing other code. The procedure for creating a new one is:

1. Create a new custom field with the new type (nnn)
2. Create the files:
   a. nnn.php in modules/SDK/examples
   b. nnn.js in modules/SDK/examples
   c. nnn.tpl in Smarty/templates/modules/SDK/examples
   These files manage the behavior of the new field depending on the context (list, detail, and so on)
3. Register the new type with the class method SDK::setUitype.

In modules/SDK/examples/VTE-SDK-2.php there are various examples of field creation and uitype registration.
In modules/SDK/doc/VTE-SDK-2.pdf under **Uitypes List** you can find the list of the main standard uitypes.

```
SDK::setUitype($uitype, $src_php, $src_tpl, $src_js, $type='', $params='');
```

$uitype : the number of the new type; it must be nnn (the name of the files)
$src_php: the path of nnn.php
$src_tpl: the path of nnn.tpl (without Smarty/templates/ at the beginning)
$src_js : the path of nnn.js
$type : the type in webservice format (‘text’, ‘boolean’, and so on)
$params : not used yet

To remove the uitype:

```
SDK::unsetUitype($uitype);
```

$uitype : it is the number of the uitype to remove (the files associated with it will not be deleted)

> We recommend using uitype with a value greater than 2000, to avoid conflicts with the future releases of **vtenext**.

The php script has several variables available, the first one is:

$sdk_mode : the views that can be customized for the new uitype (“insert”, “detail”, “edit”, “relatedlist”, “list”, “pdfmaker”, “report”, and so on)

Depending on the type of $sdk_mode, various variables can be read and modified.

detail
to manage the display of the field in DetailView
INPUT
$module : the current module name
$fieldlabel: the label of the field
$fieldname : the name of the field
$col_fields: (array) the values of the fields
OUTPUT
$label_fld[] : the label translated
$label_fld[] : the value to be displayed

edit
to manage the display of the field in EditView
INPUT
$module_name : the current module name
$fieldlabel : the label of the field
$value : the value of the field
OUTPUT
$editview_label[] : the label translated
$fieldvalue[] : the value to be displayed

relatedlist, list, pdfmaker
to manage the display of the field in ListView, RelatedList and PDFMaker
INPUT
$sdk_value : the value of the field
OUTPUT
$value : the value to be displayed

report
to manage the display of the field in Report
INPUT
$sdk_value : the value of the field
OUTPUT
$fieldvalue : the value to be displayed

> If the value to be displayed from the interface **is formatted differently** than the value saved in the database (e.g. number 1.000,25 which must be saved as 1000.25) then the following methods must also be managed to save the value in the correct format and search for it.

insert
to convert the value to the format to be saved in the database
INPUT

$this->column_fields : (array) the values of the fields
$fieldname : the name of the field
OUTPUT
$fldvalue : the value to save in the database

formatvalue
to convert the value coming from the $_REQUEST into the format saved in the database (used in the new management of Conditional Fields)
INPUT
$value : the value of the field
OUTPUT
$value : the value converted to database format

querygeneratorsearch
to convert the value searched in the lists and filters by the user
INPUT AND OUTPUT
$fieldname : the name of the field
$operator : the comparison operator
$value : the value searched

customviewsearch
to manage the conversion of the value in the popup filters for the reference fields
INPUT AND OUTPUT
$tablename : the table of the field
$fieldname : the name of the field
$comparator : the comparison operator
$value : the value searched

popupbasicsearch
to manage the conversion of the value in the popup search for the reference fields
INPUT
$table_name : the table name
$column_name : the column name
$search_string : the value searched
OUTPUT
$where : the condition of the query
e.g. $where = "$table_name.$column_name = '".convertToDBFunction($search_string)."'";

popupadvancedsearch
to manage the conversion of the value in the advanced popup search for the reference fields
INPUT AND OUTPUT
$tab_col : the table and the column of the field
$srch_cond : the comparison operator
$srch_val : the value searched

reportsearch
to convert the value searched in the reports by the user
INPUT AND OUTPUT
$table : the table of the field
$column : the column of the field
$fieldname : the field name
$comparator : the comparison operator
$value : the value searched

Hooks
data/CRMEntity.php
include/ListView/ListViewController.php
include/utils/crmv_utils.php
include/utils/EditViewUtils.php
include/utils/DetailViewUtils.php
include/utils/ListViewUtils.php
include/utils/SearchUtils.php
include/QueryGenerator/QueryGenerator.php
modules/PDFMaker/InventoryPDF.php
modules/Reports/ReportRun.php
modules/Users/Users.php
modules/CustomView/Save.php
modules/CustomView/CustomView.php
Smarty/templates/DisplayFieldsReadonly.tpl
Smarty/templates/DisplayFieldsHidden.tpl
Smarty/templates/DetailViewFields.tpl
Smarty/templates/EditViewUI.tpl
Smarty/templates/DetailViewUI.tpl

Revision #16
Created 2020-02-19 17:50:41 UTC by ddalmaso
Updated 2025-03-04 16:03:44 UTC by m.maporti