

Enhanced Studio Manual

3.1



dispage

Enhanced Studio Manual
Release 3.1.04 – Edition I, 2010
Copyright © 2008 - 2010 Dispage - Patrizio Gelosi, All Rights Reserved.

This document is subject to change without notice.

Disclaimer

The Enhanced Studio software and all related documents are distributed on an “**AS IS**” basis, **WITHOUT WARRANTY OF ANY KIND**, either express or implied.

Contents

Preface	5
Intended Audience	5
Overview	6
Compatibility	8
Supported Languages.....	8
Feature Matrix	9
System Administration.....	10
Code Fields in SugarCRM Studio	11
How to create a “Hello World !” Code Field	12
EditArea tool	14
PHP Online Syntax Checking (New in 3.1.04).....	15
Advanced Code-type field management	16
How to modify an existing Code-type field.....	22
Usage of Logic Hook Programming Interface (FULL)	23
LHPI: A working Example	26
Code Fields and Dispage Extension Manager Framework.....	27
Code Fields in Module Builder	28
How to create a new Code-type field	28
How to modify an existing Code-type field.....	28
Advanced Code-type field management	28
AJAX Callback fields (New in 3.1.04)	29
Enhanced Studio Configuration (New in 3.1.04)	31
Technical Notes.....	32
General.....	32
Code Preservation.....	32
Samples / Scenarios	33
1. Progressive Page ID Code.....	33
2. Opportunities warning on amount	33
3. Multiple File Upload	35
4. SugarCRM standard text field	37
5. Google Map automatic link.....	37
6. Generic AJAX Call (using YUI libraries)	39
7. Generic AJAX Call (using jQuery).....	40
8. Autocomplete AJAX country field (New in 3.1.04)	41
9. Generic SugarCRM DB / External DB Query.....	42
10. Contact List in Account Module	43

Webography..... 44

- Enhanced Studio on Dispage website 44
- Enhanced Studio on SugarForge 44
- Enhanced Studio on SugarExchange (2.0 and 3.0 rel.) 44
- Enhanced Studio on Sugar Forums 45

Preface

Enhanced Studio is a powerful tool for SugarCRM™ Administrators.

It aims to provide SugarCRM **Studio** tool and **Module Builder** with new features to meet any customization requirement.

With the fundamental aid of Enhanced Studio, **SugarCRM can be easily customized without touching any line of SugarCRM code.**

Intended Audience

Enhanced Studio, as well as this Guide, is targeted to a wide range of audience.

A non-developer Administrator will find easy-following procedures applied to many scenarios at the [Samples / Scenarios](#) of this Document or on the web (to this end, it is accompanied by a rich [Webography](#)).

On the other side, a developer will be pleased to explore the Enhanced Studio capabilities and fit them to his specific demands: from the Database to the graphic interface design.

This Guide is written for the users of **BASIC**, **DEMO** and **FULL** versions of Enhanced Studio: version-specific sections are marked with (BASIC), (DEMO) or (FULL).

Overview

Thanks to the Enhanced capabilities, **the Administrator can add any element he desires to all the Views available in SugarCRM (List View, Edit View, Quick Create View, Detail View, Search View and Dashlet View) of any SugarCRM Module or custom Module.**

An element (hereby referred to as **“Code field”**) might be one of the followings :

- Images
- Buttons
- AJAX-based fields doing any action
- AJAX-based autocomplete fields
- jQuery actions
- Fields automatically evaluated from others
- Embedded elements (Flash, Java, etc.)

...and whatever the Administrator has in mind (there's virtually no limitation)

Code fields, which are basically SugarCRM fields, can be **static** or **dynamic** at Administrator's choice.

Static fields are mainly graphic elements (like Images, Buttons, etc.), whereas **dynamic** fields are elements that can store/retrieve values to/from Database, like any other SugarCRM field does.

Administrator can freely customize the Database to store the data of the dynamic fields: the only limitation to the DB field creation is obviously the syntax of the DB Engine used.

After deployed, Code field values are available to be:

- Created
- Viewed in a List
- Viewed in Detail
- Edited
- Searched for
- Ordered by
- Added to a Dashlet
- Viewed in Subpanels
- Imported from a file
- Added to a Report of SugarPRO/ENT Report Module

A Code field can be given a different behaviour for each SugarCRM View. For instance a standard SugarCRM text field can be built using Enhanced Studio: the value is simply displayed on the page when List-viewed or Detail-viewed whereas it can be edited in an input field when Edit-viewed (see Sample **“SugarCRM standard text field”**).

From Enhanced Studio ver. 3.1 the new **Logic Hook Programming Interface** is available to program SugarCRM logic hooks without manually changing the files.

Each logic hook can be programmed separately. Moreover, field-independent logic hooks such as “after_ui_footer” , “server_round_trip” or “after_login” can be either programmed from Code fields.

(New in 3.1.04) AJAX elements are now fully supported with the new **AJAX Callback** field type. Starting from Enhanced Studio 3.1.04, even scripts called by AJAX requests can be managed from inside the SugarCRM application.

Compatibility

Enhanced Studio is currently compatible with **all SugarCRM 5 and 6 Versions / Editions (with the exclusion of the betas)**.

The last Enhanced Studio releases for each SugarCRM version can be found in the following table:

SugarCRM Versions	SugarCRM Editions	Enhanced Studio Releases	Enhanced Studio Versions
SugarCRM 5.0.x	CE / PRO / ENT	Enhanced Studio 2.0	DEMO, FULL
SugarCRM 5.1.x SugarCRM 5.2.x	CE / PRO / ENT	Enhanced Studio 3.0	DEMO, FULL
SugarCRM 5.5.x SugarCRM 6.0.x	CE / PRO / ENT	Enhanced Studio 3.1	BASIC, DEMO, FULL

Table 1

More detailed and up-to-date info on SugarCRM compatibility can be found at

http://www.dispage.com/products/enhanced-studio#tech_info

Supported Languages

Enhanced Studio has a multi-language support from 3.0 version.

The following languages are currently supported:

- English
- Spanish
- Italian
- Dutch
- German
- Brazilian Portuguese

Feature Matrix

The following table reports which features are present in each Enhanced Studio Release / Version.

Enhanced Studio Rel. / Ver.	2.0 DEMO	2.0 FULL	3.0 DEMO	3.0 FULL	3.1 BASIC	3.1 FULL
Features						
SugarCRM Studio Code type support	✓	✓	✓	✓	✓	✓
Module Builder Code type support	-	✓	-	✓	✓	✓
Dynamic DB-types	✓	✓	✓	✓	✓	✓
Support for Create / QuickCreate Code fields	✓	✓	-	✓	✓	✓
PHP / Javascript / HTML / CSS language specific Editor	-	-	✓	✓	✓	✓
Detail/Edit/List View Code type support	✓	✓	✓	✓	✓	✓
Basic/Advanced Search Code type support for Dynamic DB-types	✓	✓	✓	✓	✓	✓
Dashlet List / Search Views Code type support	-	-	✓	✓	✓	✓
Popup View Code type support (SugarCRM 6)	-	-	-	-	✓	✓
Importable Code Dynamic Values	-	-	✓	✓	✓	✓
Sortable Code Dynamic DB-types in List View	-	-	✓	✓	✓	✓
Subpanels Code type support	-	-	-	-	✓	✓
Logic Hook Programming Interface	-	-	-	-	-	✓
Report Generation Supported for Code Dynamic DB-types	-	-	✓	✓	✓	✓
Online Syntax Checking of PHP Code (New in 3.1.04)	-	-	-	-	-	✓
AJAX Callback Field Type (New in 3.1.04)	-	-	-	-	-	✓
Enhanced Studio Configuration (New in 3.1.04)	-	-	-	-	✓	✓
Exportable / Importable customizations	✓	✓	✓	✓	✓	✓
Limit of Code fields for each Module	Unlimited	Unlimited	Unlimited	Unlimited	1	Unlimited

Table 2

Table above is available online at

<http://www.dispage.com/products/enhanced-studio/overview#comparison>

Some important notes about Enhanced Studio versions:

- **Before 3.1 release** there were only two versions for each Enhanced Studio release: **DEMO** and **FULL**. DEMO had no time limitation but had less features than FULL.
- **From 3.1.04 release** three versions are published for each release: **BASIC**, **DEMO** and **FULL**.

The differences between BASIC and FULL are reported in the last two columns of [Table 2](#).

The DEMO has the same features of the FULL and two limitations:

1. 30 Days of evaluation
2. An upper limit of **4 total Code type fields**

System Administration

From 3.1 release, Enhanced Studio is managed as a **Dispage Extension** from the **Dispage Extension Manager Tool** (<http://www.dispage.com/index.php/products/extension-manager>).

Enhanced Studio can be now seamlessly managed as a Mozilla Firefox Add-on, with additional info on SugarCRM compatibility versions, supported languages, expire date and whatever is useful to schedule SugarCRM and Enhanced Studio upgrades.

The procedures to **install**, **uninstall** and **upgrade** Enhanced Studio are explained in the **Extension Install Guide** downloadable from the following link:

http://www.sugarforge.org/frs/download.php/6509/Generic_Extension_Install.1.2.pdf

Code Fields in SugarCRM Studio

Once installed Enhanced Studio, Administrator finds a new field type (“Code”) in the SugarCRM Studio field list. This is the basic element from which to start to customize SugarCRM. The first step is to create a Code-type field in any SugarCRM module, then the field is available to be added to any View.

WARNING

From 3.1 release, **Code Fields have the same functionalities in SugarCRM Studio and Module Builder.**

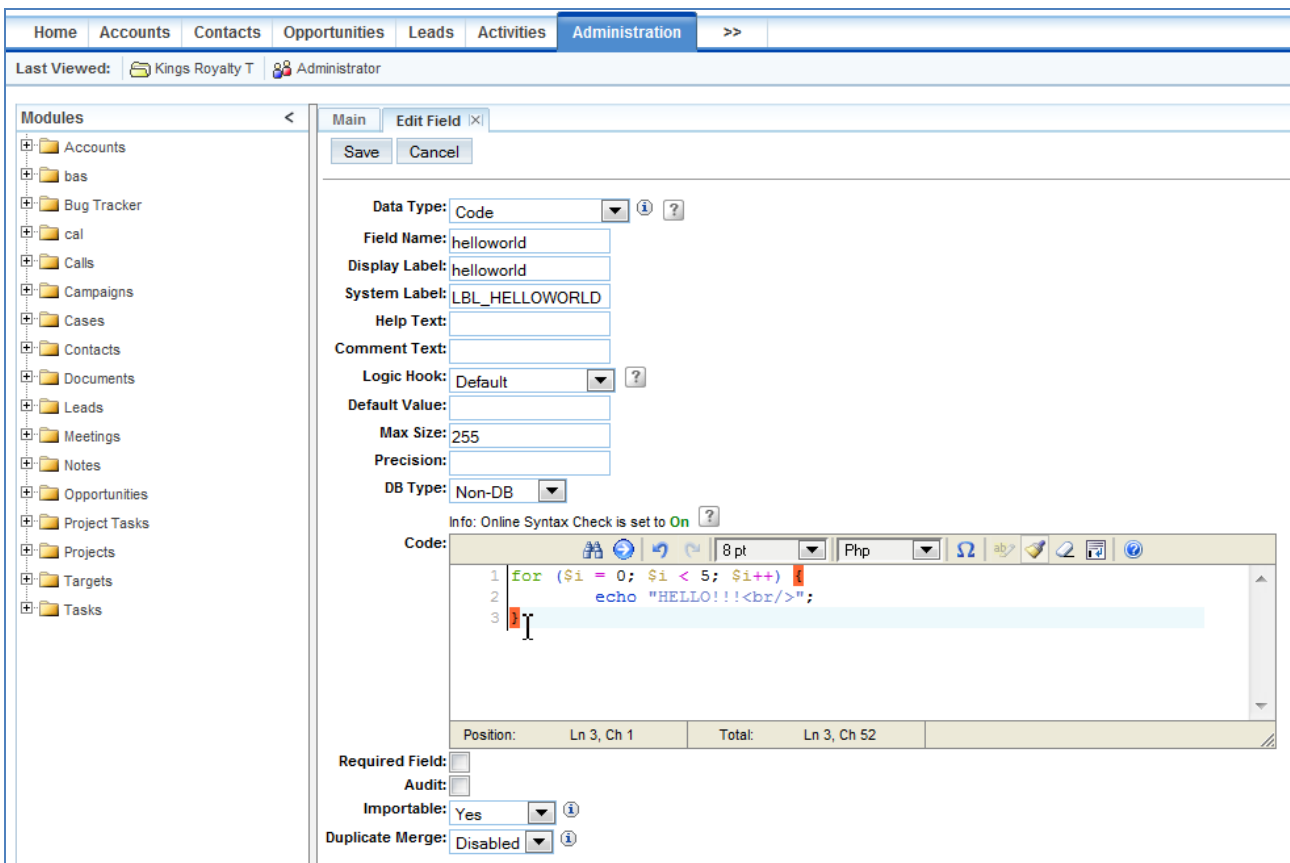


Figure 1

How to create a “Hello World !” Code Field

1. In the “Admin” → “Studio” tool, click the “Fields” link of the module to customize (“Cases” in the example in [Figure 1](#)).
2. Select “Code” in the “Data Type” dropdown at the top of the field page. A mask like the one in Figure 1 will appear.

The mask contains all the standard SugarCRM type options, plus the followings:

- **Logic Hook** selector (FULL)
- **Default Value** input field
- **Max Size** input field
- **Precision** input field
- **DB Type** selector
- **Code** Edit Area Tool

(FULL) The **Logic Hook** selector has been added to specify a Logic Hook to assign the code to, as explained in paragraph [Usage of Logic Hook Programming Interface](#). To have a Code field PHP-eval()’d each time it is accessed to (as in the old Enhanced Studio releases), “**Default**” option must be selected.

The **DB Type** selector allows to choose whether the Code field must be created in the DB or not and its type. To create a **static** Code field (with no DB value to store, as explained in the Overview section), the DB Type input field must be set to “**Non-DB**”.

If DB Type has a value different from “Non-DB” a **dynamic** Code field with the selected type will be created in SugarCRM Database.

The **Default Value**, **Max Size** and **Precision** fields come in use to further customize the field as it is created in the DB.

The **Code** EditArea Tool allows to enter the PHP code that is run-time eval()’d as the field is accessed for being viewed or edited, or when the related logic hooks is triggered.

In this “Hello world !” example a static Code field named “helloworld” (which will be automatically renamed into “helloworld_c” by SugarCRM Studio) is being creating with the simple Code

```
echo "HELLO !";
```

No change involves the Database, the only action performed as the field is viewed, listed or edited (depending on the Layout it is put into) is to write the string “HELLO !”.

PHP Code can be developed directly from inside the SugarCRM application thanks to the [EditArea tool](#).

(FULL) Additionally, a [PHP Online Syntax Checking \(New in 3.1.04\)](#) is performed on the PHP code before saving the Code field.

3. Save the field

4. Add the new Code field to any View or Subpanel.

Available Layouts are “EditView”, “DetailView”, “ListView”, “QuickCreate”, “Search”, “Sugar Dashlet” and “Popup View” (the last is supported from SugarCRM ver. 6).

WARNING

Only **dynamic** fields should be added to the Search Layouts since **static** fields are not created in the Database and so they cannot be searched for.

If the “helloworld” field of this example is added, for instance, to the “ListView” Layout, the Cases Module looks like the one in the [Figure 2](#).

Num.	Subject	Account Name	helloworld	Priority	Status	Assigned to
1	Having Trouble Plugging It In	Doggie Diner Co Ltd 418071	HELLO !	Medium	Pending Input	sarah
2	Having Trouble Plugging It In	Dirt Mining Ltd 401795	HELLO !	Medium	Assigned	chris
3	Having Trouble Plugging It In	Rhyme & Reason Inc 193967	HELLO !	Low	New	sally
4	System is Performing Too Fast	Nimble Technologies Inc 884020	HELLO !	Medium	New	max
5	Need to Purchase Additional Licenses	Slender Broadband Inc 656789	HELLO !	Low	Closed	will
6	Warning message when using the wrong browser	TJ O'Rourke Inc 339632	HELLO !	High	Assigned	sarah
7	System is Performing Too Fast	COMPLETE HLDNG 465534	HELLO !	Low	Pending Input	sally
8	System is Performing Too Fast	Tri-State Medical Corp 344389	HELLO !	Low	Rejected	will

Figure 2

EditArea tool

The EditArea Tool ¹ includes some features very useful to developers such as :

- PHP specific syntax highlight
- Parenthesis highlighting.
- Line numbers
- Search & Replace Tool
- Undo & Redo
- Resizable Editor Panel
- Extra Javascript, CSS and HTML syntax-highlighting support
- Graphic HTML Symbol tool
- Word-wrap line option

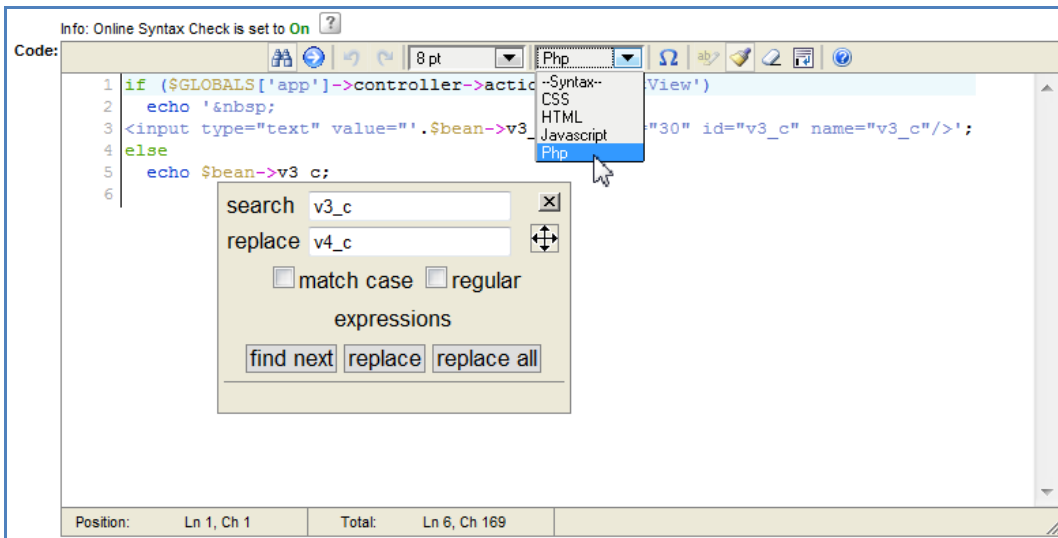


Figure 3

¹ © 2008-2010 Christophe Dolivet.
Licensed under LGPL, Apache and BSD licenses.

PHP Online Syntax Checking (New in 3.1.04)

(FULL) Before saving Code Type fields (as well as [AJAX Callback fields \(New in 3.1.04\)](#)) an online checking is automatically performed on PHP syntax of the Default code and all the Logic Hooks in the field.

If any syntax error/warning occurs, the field is not saved and the error text is reported in both an alert popup ([Figure 4](#)) and an error mark placed on the left of the code ([Figure 5](#)).

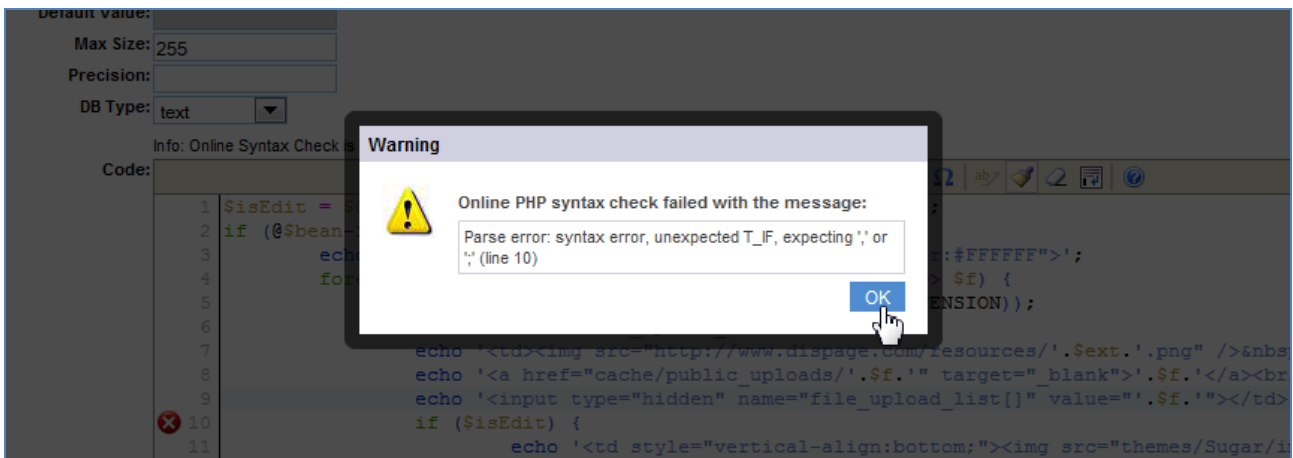


Figure 4

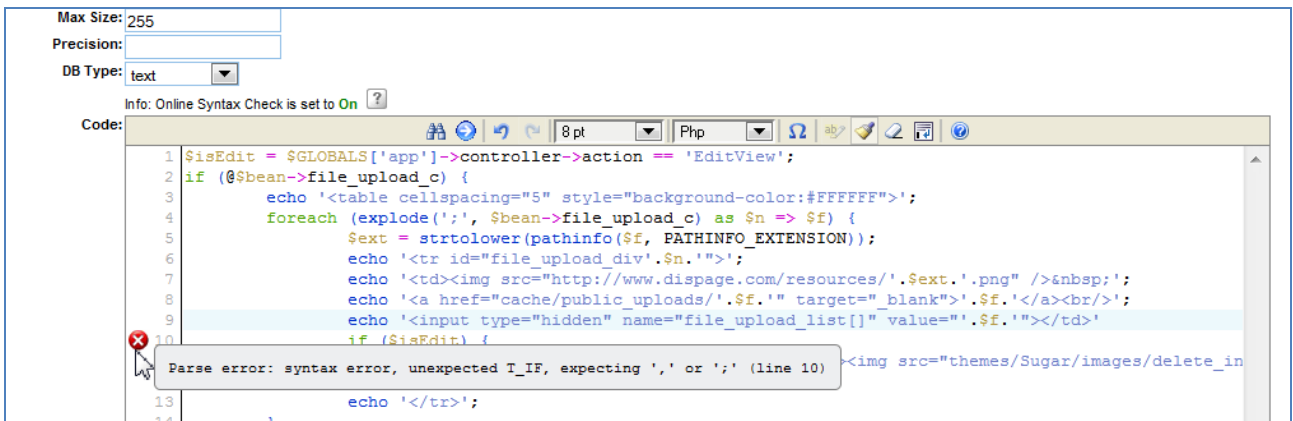


Figure 5

Notes:

- If multiple errors occur, only the first error of the first Logic Hook encountered is returned.
- PHP code is submitted to dispage server **with the sole purpose of validating the PHP syntax**, then the result is sent back to client. **Dispage do not keep any copy of the PHP code.**

- Total size of PHP code must be less than 64 Kbytes for each submission.
- Only **Compile-time parse errors and warnings** are detected. Run-time errors and warnings depend on the execution context and are excluded from syntax checking.
- Online Syntax Checking can be disabled from [Enhanced Studio Configuration \(New in 3.1.04\)](#).

WARNING

Dispage does not warrant that the PHP Online Syntax Checking Service will be **error-free, uninterrupted or will meet any particular criteria of accuracy, completeness or reliability of information, performance or quality.**

Advanced Code-type field management

As explained in the previous paragraph, when a Code field type is selected in SugarCRM Studio a Code specific mask appears. The mask has the standard input fields in common with other SugarCRM field types, and in addition it has the specific fields listed below:

1. Logic Hook selector (FULL)

The logic Hook Selector is the switch to access the new **Logic Hook Programming Interface (LHPI)**. Postponing this topic to the next Paragraph, “**Default**” option causes the Code field to be PHP-eval()'d each time it is accessed to.

This means that the PHP Code entered in the “Code” entry while Logic Hook is set to “**Default**” is run-time executed when a record containing that field is edited or viewed from any SugarCRM Layout.

WARNINGS

- **Default** Logic Hook option is the basic Code option and grants the backward compatibility with the Code fields developed with Enhanced Studio rel. < 3.1.
- The Logic Hook is a switch that allows to view/edit the code of all the Logic Hooks: if changed it simply switches the entries of the Panel from one selection to the other. **Each entry value is kept in memory and retrieved automatically when the Logic Hook selection changes.**

2. DB Type selector

The “DB Type” value switches the kind of Code field generated:

2.1. Static (Non-DB) field

To select a static field just leave the “**Non-DB**” option selected.

This kind of field is executed only at run-time and does not need to be stored/retrieved to/from DB, so **when a static field is created no field is physically added to the Database.**

It can be used for many kinds of elements that do not need to store a value to the Database, for example buttons that do any action or fields containing operations on the content of other fields, that need to be executed each time they are viewed.

All the Code fields of the Enhanced Studio releases prior to 2.0 belong to this category.

An example a bit more useful than the “Helloworld” is the “[Opportunities warning on amount](#)” reported in the Samples Section.

2.2. Dynamic field

From the Enhanced Studio 2.0 release, Code fields can store/retrieve values of any type to/from Database, exactly like other SugarCRM fields.

The PHP code assigned to these fields is also eval()'d at run-time and the field values retrieved from the DB can be used from inside the PHP code itself (a standard SugarCRM Text field is reproduced in the [SugarCRM standard text field](#) sample for teaching purposes).

Dynamic fields are created in the Database according to the DB type selected and the DB Engine in use, as detailed in [Table 3](#).

DB-Type	MySQL	MSSQL	Oracle
Non-DB	-	-	-
Int	Int	Int	number
Double	Double	Float	number(30,10)
Float	Float	Float	number(30,6)
Uint	int unsigned	Int	number(15)
Ulong	bigint unsigned	Int	number(38)
Long	Bigint	bigint	number(38)
Short	smallint	smallint	number(3)
Varchar	varchar	varchar	varchar2
Text	text	text	clob
Longtext	longtext	text	clob
Date	date	datetime	date
Enum	varchar	varchar	varchar2(255)
Relate	varchar	varchar	varchar2
Multienum	text	text	clob
Html	text	text	clob
Datetime	datetime	datetime	date
Time	time	datetime	date
Bool	bool	bit	number(1)
Tinyint	tinyint	tinyint	number(3)
Char	char	char	char
Blob	blob	image	varchar2(36)
Longblob	longblob	Image	blob
Currency	decimal(26,6)	Decimal	blob
Decimal	decimal	Decimal	number(26,6)
Decimal2	decimal	Decimal	number(20,2)
Id	char(36)	varchar(36)	number(30,6)

Table 3

3. Default Value entry

A default value can be entered for each Dynamic field, according to the “DB Type” selected (the Default Value is ignored for Static fields).

This entry is disabled for “text”, “longtext”, “multienum”, “blob” and “longblob” DB-Types to follow a constraint of the DB Engines.

WARNING

If “**Varchar**” DB Type is selected the default value is automatically set to 255.

This value can be changed if needed but **in no case must left blank** or the Varchar field will not be created correctly in the DB.

4. Max Size entry

A max character length can be assigned to the Dynamic fields to optimize Database disk space usage. The Max Size is ignored for Static fields and for those fields having a fixed conversion Size by default (see the conversion “DB Type” table above).

5. Precision entry

This entry is used to specify the 2nd parameter in the SQL creation string of the field (it indicates the precision for the “Double”, “Float” and “Decimal” types).

WARNING

Precision is automatically set to 4 for “**Float**” DB-Types. This value can be changed if needed but **should not left blank** or the Float field might not be created correctly when using the MySQL DB Engine.

6. Code entry

The content of the "Code" entry is a PHP code which is run-time eval()'d as the field is viewed or edited.

Any operation can be performed from PHP code, for example adding any **HTML** or any **JavaScript** Code (even using **AJAX** functions or **jQuery**, which is natively supported from 3.1) anywhere in any SugarCRM view.

Dynamic field value can be referred to as

```
$bean-><field_name>
```

in the PHP code. For example the first name of a contact can be easily displayed through a Code field in Contacts Module with this simple PHP code:

```
echo $bean->first_name;
```

The code to refer to a custom “try” field is (note that the “_c” prefix is automatically added to the custom field name by SugarCRM Studio):

```
echo $bean->try_c;
```

Here's a complete list of the variables that can be accessed from inside the PHP Code :

- & \$bean² (the bean object of the specific record)
- \$event (containing useful info about the specific view)
- \$args
- \$GLOBALS

The exact content of the variables should need an in-depth examination that involves SugarCRM application structure and would go beyond the aim of this Manual. However, the following is a useful suggestion to explore the content of these variables while they are runtime eval()'d, and for more general **Debug** purposes.

Simply add the following debugging line to the Code:

```
error_log ( print_r ( <variable>, true ), 3, "<log_file>" );
```

Where:

- <variable> is a variable to watch, chosen among the previous list
- <log_file> is the path to a writable log file (for example "/tmp/watch.txt")

Then, once accessed to any page from which the Code field is called, the content of the variables can be found in the specified log file.

WARNING

Please remember to **delete the error_log line** from the code after finished, or it will carry on appending data to the log file every time the field is accessed.

This Guide cannot go into the specific code language syntax and the SugarCRM application structure. However, it provides an exhaustive "Samples / Scenarios" section with a great wealth of scenarios.

² The meaning of the ampersand ("&") prepended to the variable is that it is passed to the Code function by reference (see <http://www.php.net/manual/en/functions.arguments.php>). This involves that if \$bean is changed from inside the PHP Code, changes persist out of the PHP Code.

After created, fields can be added to almost any Layout and almost any action can be performed on them (depending on their types) as described in [Table 4](#).

Action \ Field DB-Type	Static	Dynamic ("Text" Type)	Dynamic (Other Types)
Quick Create	✓	✓	✓
Detail View	✓	✓	✓
Edit	✓	✓	✓
List View	✓	✓	✓
Basic/Advanced Search	-	✓	✓
Dashlet View	✓	✓	✓
Subpanel View	✓	✓	✓
Popup View (SugarCRM 6)	✓	✓	✓
Import	-	✓	✓
Order By (List View)	-	-	✓
Reported (in Sugar PRO / ENT Report Module)	-	✓	✓

Table 4

Numeric Code DB fields are also recognized by the Enhanced Search plugin (<http://www.dispage.com/products/enhanced-search>) and can be searched using the Enhanced Search numeric operators.

How to modify an existing Code-type field

Both **static** and **dynamic** Code fields can be modified after they have been deployed from the SugarCRM Studio tool according to requirements. The only important limitation comes from the Database constraints and is resumed in the following warnings.

WARNINGS

- DB Type of a Code field **must not be changed from static to dynamic in any case.**
- DB Type must not be modified **to an incompatible Db Type** (i.e. from “text” to “integer”) when the related Database field contains any not-empty value (that might also happen when no row is shown in SugarCRM, as rows are logically but not physically deleted). This obviously leads to an error if such DB values cannot be automatically converted by the DB Engines. **Thus, it is allowed to change dynamic fields even after they have been created, but the related field in the Database must be manually emptied if the specific DB type conversion is not supported by the DB Engine.**

Moreover, if a Dynamic Code field is changed to a Static field after it has been created in SugarCRM Studio, the related DB field is not automatically removed from the Sugar tables, and should be manually deleted in order to save Database space.

Enhanced Studio is a very powerful tool that allows deep after-deployment changes: this feature fully balances the precautions expressed in this paragraph.

Usage of Logic Hook Programming Interface (FULL)

From Enhanced Studio 3.1, Code fields have become a powerful tool to program the SugarCRM Logic Hooks. The main advantages in using Enhanced Studio’s Logic Hooks Programming instead of manually modifying custom files are

- **Ease of managing:** Logic Hooks are logically linked to a Code field and can be managed from inside the SugarCRM Studio or Module Builder thanks to the **Logic Hook Programming Interface (LHPI)**.
- **Ease of programming:** PHP code of AJAX callbacks can be entered from the PHP **EditArea tool** SugarCRM Studio or Module Builder and debugged with the **PHP Online Syntax Checking (New in 3.1.04)**
- **Logic Hooks seamlessly exported / imported:** Logic Hooks made with LHPI are automatically included in any published / exported package they belong to, and they are seamlessly rebuilt in any SugarCRM installation in which the packages are installed.

The key element to pilot the LHPI is the “**Logic Hook**” selector. Actually it is not a common selector, it is rather a switch giving an additional info on which logic hooks are currently valued (as in **Figure 6**). Each Code field can perform 16 distinct actions (with 16 different PHP codes), one for each Logic Hook.

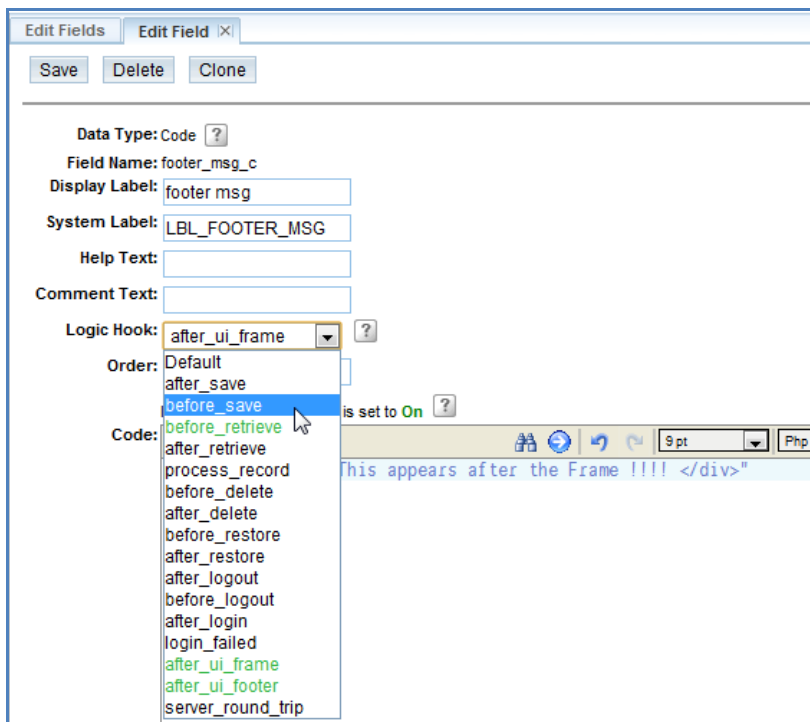


Figure 6

Plus, the “Default” abstract Logic Hook, that is the PHP executed each time the field is accessed from any View ³, is available. This is the option with the entries of the Code fields generated with Enhanced Studio Versions prior to 3.1 .

Thus, to enter the Code for a specific Logic Hook it is enough to select it and enter the PHP script in the Code entry.

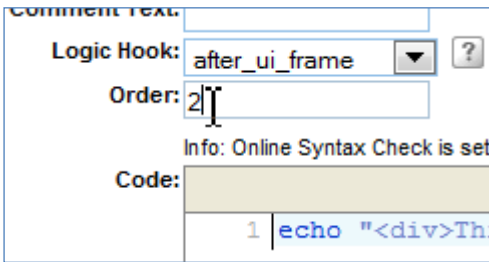


Figure 7

All the SugarCRM logic hooks are supported by LHPI as reported in [Table 5](#).

Since more than one Logic Hook of the same type can exist (taking into consideration that Logic Hooks can be also programmed directly from the SugarCRM code), the LHPI allows to assign directly an **order of execution** to each Logic Hook ([Figure 7](#)) .

Logic Hook	Type	Trigger Level
after_ui_frame	Application	Module
after_ui_footer	Application	Module
server_round_trip	Application	Application
before_delete	Module	Field
after_delete	Module	Field
before_restore	Module	Field
after_restore	Module	Field
before_retrieve	Module	Field
after_retrieve	Module	Field
before_save	Module	Field
after_save	Module	Field
process_record	Module	Field
after_login	Users Module	Application
before_logout	Users Module	Application
after_logout	Users Module	Application
login_failed	Users Module	Application

Table 5

³ Technically the PHP Code of “Default” abstract Logic Hook is linked to “process_record” and “after_retrieve” basic Logic Hooks.

When the Logic Hook selection changes, all the entries of that Logic Hook are kept in browser's memory (and retrieved as the Logic Hook option is re-selected) until they are permanently saved as the Code field is saved.

To help the Administrator to keep under control all the Logic Hooks at a time, Logic Hooks with a non-empty Code entry are green-highlighted in the dropdown menu of the Logic Hook selector.

Even though Logic Hooks are programmed from inside a Code field, they are not necessary bound to the field itself and can be triggered even by events not directly related to the field.

There are three **Trigger Levels** of Logic Hooks, related to the context in which they are fired (see [Table 5](#)):

- **Field:** Logic Hook is fired by an event in which the Code field is involved.
- **Module:** Logic Hook is fired by an event in which is involved the Module that the Code field belongs to.
- **Application:** Logic Hook is fired by an event not related to the Code field.

Logic Hooks are discussed in detail at the following SugarCRM wiki page:

http://www.sugarcrm.com/wiki/index.php?title=Logic_Hooks

LHPI: A working Example

If a Code field is created in the Accounts Module with the code

```
echo "<div>This Appears after the Frame!!!!</div>";
```

in the “after_ui_frame” Logic Hook like in [Figure 6](#), the result is the one in [Figure 8](#).

<input type="checkbox"/>	RR. Talker Co 40183	Ohio	(473) 773-4109	tr2y
<input type="checkbox"/>	Q3 ARVRO III PR 864596	San Jose	(632) 541-5847	tr2y
<input type="checkbox"/>	Q.R.&E. Corp 538389	Denver	(182) 883-4265	tr2y
<input type="checkbox"/>	Pullman Cart Company 65790	Salt Lake City	(823) 790-2506	tr2y
<input type="checkbox"/>	Overhead & Underfoot Ltd. 333126	Sunnyvale	(315) 050-5259	tr2y
<input type="checkbox"/>	OTC Holdings 449585	Denver	(107) 249-1559	tr2y

Select ▾ Selected: 0

Mass Update

Assigned User: Type:

Industry: --None-- ▾

Opt out Primary Email: --None-- ▾

This appears after the Frame !!!!

Figure 8

Code Fields and Dispage Extension Manager Framework

Since Enhanced Studio is managed through **Dispage Extension Manager (DEM)**, the Javascript Framework included in DEM is available from the PHP code of Code fields.

The following libraries are included in DEM ver. 1.0.09 (info on copyright and licenses can be found at <http://www.sugarforge.org/frs/download.php/6328/LICENSE.txt>):

- JQuery-1.4.2 (<http://jquery.com/>)
- JQuery-UI-1.8.4 (<http://jqueryui.com/>)
- Boxy-0.1.4 (<http://onehackoranother.com/projects/jquery/boxy/>)
- JGrid-3.7.2 (<http://www.trirand.com/blog/>)

If used in conjunctions with Logic Hooks, the range of Enhanced Studio customizations widens even further.

The sample “**Multiple File Upload**” is, among the others, a working example of jQuery usage in a Code field: examples of usage of jQuery ID Selector, “\$.attr” and “\$.remove” APIs can be found in it.

Another jQuery sample is “**Generic AJAX Call (using jQuery)**” with an example of the jQuery “\$.ajax” API usage.

An AJAX working example is **Autocomplete AJAX country field (New in 3.1.04)**. Some useful jQuery and AJAX snippets are shown in it:

- How AJAX Callback can be programmed from SugarCRM web application thanks to the **AJAX Callback fields (New in 3.1.04)**
- The usage of the **UI Autocomplete Widget** (included in jQuery-UI)
- The usage of jQuery “.ready” API, to execute a javascript code after the DOM is completely loaded

Code Fields in Module Builder

From Enhanced Studio 3.1, there are no significant differences in the code field management between SugarCRM Studio and Module Builder, except for the following obvious integrations.

How to create a new Code-type field

After followed a procedure similar to the one explained in the paragraph [How to create a “Hello World !” Code Field](#) of the previous Chapter, the following further step must be performed:

5. Deploy the package containing the module.

How to modify an existing Code-type field

Both Static and Dynamic Code fields can be modified according to your preferences using the SugarCRM Module Builder tool, with a procedure similar to the creation:

1. Edit the Code field according to your preferences.
2. Save the field.
3. Deploy the package.

As for the rest, please refer to the corresponding paragraph [How to modify an existing Code-type field](#) of the previous Chapter.

Advanced Code-type field management

The same concepts expressed for SugarCRM Studio (paragraph [Advanced Code-type field management](#)) hold true.

The only relevant difference from Studio is that when referring to a Dynamic field value as “`$bean-><field_name>`”, the “`_c`” postfix is not automatically added.

For example if the field’s name is “try”, its value can be easily dumped through this PHP code :

```
echo $bean->try;
```

AJAX Callback fields **(New in 3.1.04)**

Enhanced Studio makes the AJAX usage easy and fully exportable across different SugarCRM installations.

AJAX calls can be performed from a Code field (for example using jQuery): from Enhanced Studio 3.1.04 even the AJAX Callback can be programmed from SugarCRM application.

The advantages over a traditional PHP script manually added to the SugarCRM directory are easy to see:

- **Authenticated User Access natively checked:** no need to add code to prevent unauthorized access to the PHP Callback and to potentially compromise the security of the SugarCRM application by affecting the access permissions policy in the SugarCRM directory.
- **Ease of programming:** PHP code of AJAX callbacks can be entered from the PHP [EditArea tool](#) SugarCRM Studio or Module Builder and debugged with the [PHP Online Syntax Checking](#) **(New in 3.1.04)**
- **AJAX Callbacks seamlessly exported / imported:** AJAX Callbacks made with Enhanced Studio are automatically included in any published or exported packages containing them, and they are seamlessly rebuilt in any SugarCRM installation in which the packages are installed.

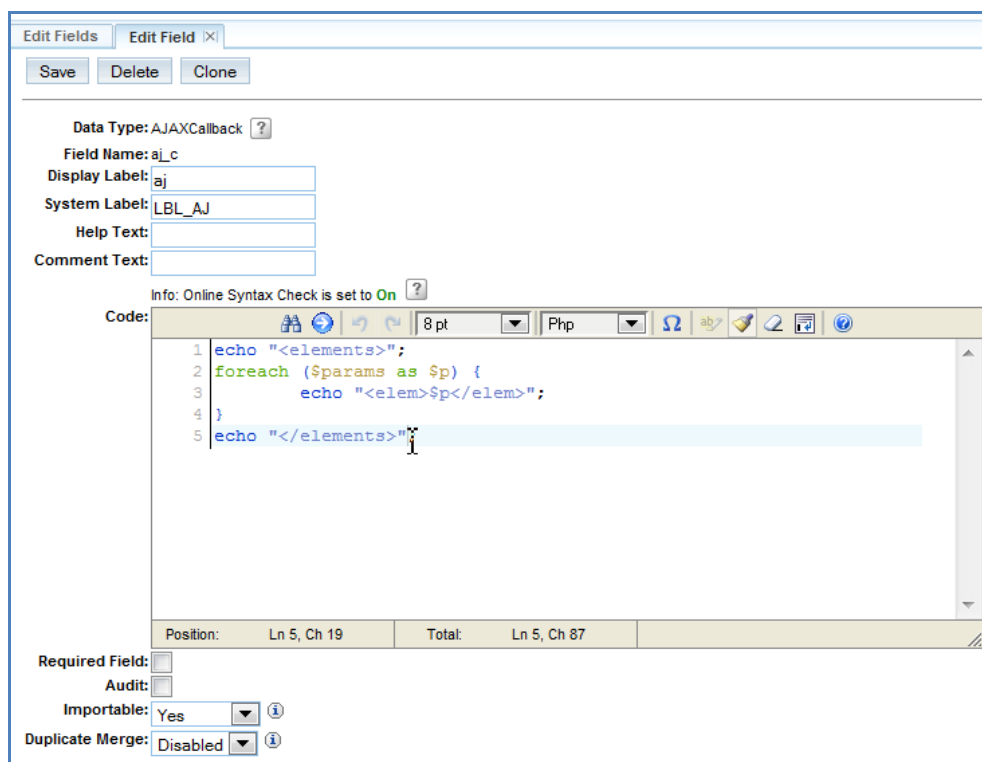


Figure 9

Once added an AJAX Callback field, it can be called with the URI

```
<sugar_base_url>/customAJAXCalls.php?call=<module_name>_<field_name>&<params>
```

where

- *sugar_base_url* is the base URL of SugarCRM.
- *module_name* is the module that the field belongs to.
- *field_name* is the name of the field, **without the “_c” postfix**.
- *params* is an optional query string containing some parameters that may be passed to the AJAX Callback (from which they are available in the **\$params** array). To send an associative array, the *params* query string must be generically set this way

```
params[<key_1>]=<value_1>& ... &params[<key_n>]=<value_n>
```

where all keys and values must be URL-encoded.

An example of AJAX callback usage is [Autocomplete AJAX country field \(New in 3.1.04\)](#) .

Enhanced Studio Configuration **(New in 3.1.04)**

From 3.1.04 release, Enhanced Studio can be configured from Extension Manager options. The Enhanced Studio Option popup (**Figure 10**) is accessible at

Dispage Extensions → “Installed Extension” Tab → Enhanced Studio Extension → Options

A popup with the following Global Options appears

- **Disable Online Syntax Checking:** to disable the [PHP Online Syntax Checking](#) **(New in 3.1.04)**
- **Initial EditArea width (px):** to set the initial width of the [EditArea tool](#) in pixels
- **Initial EditArea height (px):** to set the initial height of the [EditArea tool](#) in pixels

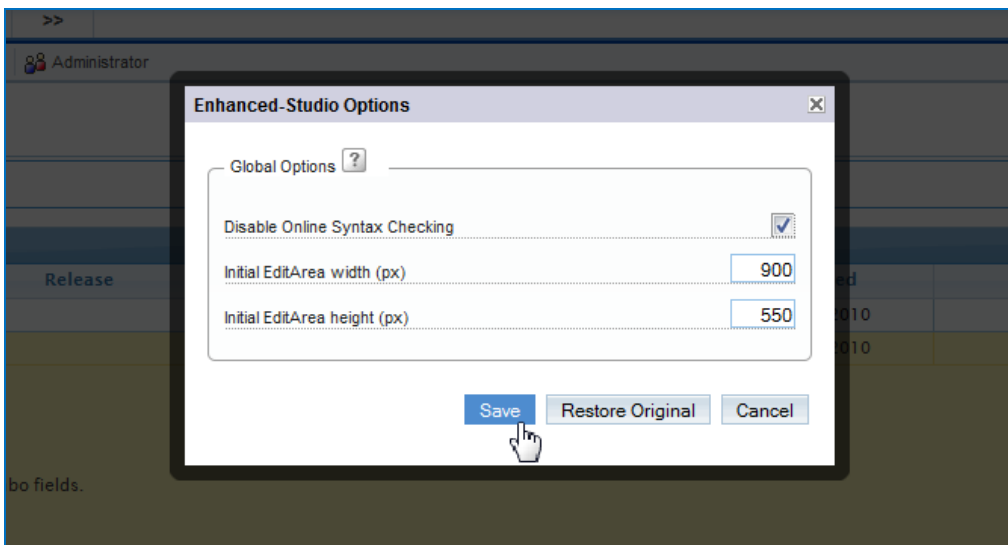


Figure 10

Technical Notes

General

One of the primary targets of the Enhanced Studio project is to minimize the impact on SugarCRM original code. The main reason is to avoid problems with Enhanced Studio functionalities when upgrading the SugarCRM main application.

The use of **Logic Hooks** is along these lines.

A new class called **FilePreserve** had been released in Enhanced Studio 2.2: it permits code-preservation of some SugarCRM files without overwriting them as it would happen in a standard package installation. The usage of the class has been extended to all SugarCRM files and to uninstall process. It is described in the following paragraph.

Code Preservation

When Enhanced Studio is installed, a new extension comes in use to preserve user-defined logic hook settings.

Before the 2.2 release, the “<sugardir>/custom/modules/logic_hooks.php” setting file was simply overwritten during Enhanced Studio installation, so the SugarCRM developers might have lost some customizations.

From the 2.2 release the Code Preservation extension automatically distinguishes between the Enhanced Studio settings and the other settings in the “logic_hooks” file.

It replaces the old Enhanced Studio code with the new release one (surrounded by a comment tag containing the Enhanced Studio Release / Version info) and preserves the original code while copying the contents into the new file version.

The extension usage is also very useful on other files that should have been overwritten by Enhanced Studio and might have been affected by the risk of being changed in future SugarCRM patches (and every time it happens, a new compatible Enhanced Studio patch might be needed).

File Preserve Class v. 2.0.2 has been released in the 3.1.02 beta Enhanced Studio version. The Class performs the followings:

- **On installation:** patches correctly any SugarCRM php or tpl (smarty template) file and adds a comment containing the name of the Plugin and the correct version to any piece of code, to permit the correct upgrade to further releases.
- **On installation errors:** performs an uninstallation of the patches installed before the errors.
- **On uninstallation:** cleans all the modification made on SugarCRM files.

Samples / Scenarios

1. Progressive Page ID Code

Scenario: add a progressive ID to the page screen of the List View of any module.

Procedure:

- i. Create a new Code field with the values below in the module where to add the counter :

Logic Hook: Default

DB Type: Non-DB

Code:

```
$GLOBALS['count_cc']++;  
echo '<div align="right">'.$GLOBALS['count_cc'].'</div>';
```

- ii. Add it to Layouts → ListView.

2. Opportunities warning on amount

Scenario: add a warning field that shows different images depending on the amount value. It shows “Thumb down” for less than 20,000 \$ amount, “Thumb up” for more than 60,000 \$.

Procedure :

- i. Create a new Code field in the Opportunities module with the values below

Logic Hook: Default

DB Type: Non-DB

Code:

```
if ($bean->amount_usdollar > 60000)  
echo "<image src='http://www.sugarcrm.com/forums/images/icons/icon14.gif'>";  
elseif ($bean->amount_usdollar < 20000)  
echo "<image src='http://www.sugarcrm.com/forums/images/icons/icon13.gif'>";
```

ii. Add it to Layouts → ListView.

The List View should appear like shown in **Figure 11**

Opportunity List					
Select ▾	Delete	Export	Merge Duplicates	Selected: 0	⏪
<input type="checkbox"/>	Name ⇅	Account Name ⇅	Sales Stage	Amount ⇅	warning1
<input type="checkbox"/>	SEA REGION S A 69708 - 1000 units	SEA REGION S A 69708	Prospecting	€75.000,00	👍
<input type="checkbox"/>	EEE Endowments LTD 534404 - 1000 units	EEE Endowments LTD 534404	Negotiation/Review	€75.000,00	👍
<input type="checkbox"/>	NW Bridge Construction 590016 - 1000 units	NW Bridge Construction 590016	Closed Lost	€10.000,00	👎
<input type="checkbox"/>	MTM Investment Bank F S B 961149 - 1000 units	MTM Investment Bank F S B 961149	Qualification	€75.000,00	👍
<input type="checkbox"/>	NW Capital Corp 846231 - 1000 units	NW Capital Corp 846231	Prospecting	€50.000,00	
<input type="checkbox"/>	Jungle Man Inc 276547 - 1000 units	Jungle Man Inc 276547	Closed Won	€10.000,00	👎
<input type="checkbox"/>	MMM Mortuary Corp 51712 - 1000 units	MMM Mortuary Corp 51712	Needs Analysis	€50.000,00	
<input type="checkbox"/>	5D Invest A/S 776587 - 1000 units	5D Invest A/S 776587	Id. Decision Makers	€10.000,00	👎

Figure 11

3. Multiple File Upload

Scenario: create a field that allows to upload multiple files for each Contact.

Procedure :

- i. Create a new Code field in the Contacts module with the values below

Name: file_upload

Logic Hook: Default

DB Type: text

Code:

```

$isEdit = $GLOBALS['app']->controller->action == 'EditView';
if (@$bean->file_upload_c) {
    echo '<table cellpadding="5" style="background-color:#FFFFFF">';
    foreach (explode(';', $bean->file_upload_c) as $n => $f) {
        $ext = strtolower(pathinfo($f, PATHINFO_EXTENSION));
        echo '<tr id="file_upload_div'. $n. '>';
        echo '<td>&nbsp;';
        echo '<a href="cache/public_uploads/'. $f. '" target="_blank">'. $f. '</a><br/>';
        echo '<input type="hidden" name="file_upload_list[]" value="'. $f. '"></td>';
        if ($isEdit) {
            echo '<td style="vertical-align:bottom;"></td>';
        }
        echo '</tr>';
    }
    echo '</table>';
}
if ($isEdit) {
    echo '
<script language="javascript">
    $("#EditView").attr("enctype", "multipart/form-data").attr( "encoding",
"multipart/form-data" );
</script>
<input id="file_upload_c" name="file_upload_c" type="file"/>
';
}

```

- ii. Switch to the “before_save” Logic Hook:

- iii. Enter the following value for “before_save”:

Code:

```

$uploaddir = 'cache/public_uploads/';
if (!file_exists($uploaddir)) {
    mkdir($uploaddir);
}
$file_up = basename($_FILES['file_upload_c']['name']);
$uploadfile = $uploaddir . $file_up;

move_uploaded_file($_FILES['file_upload_c']['tmp_name'], $uploadfile);

$bean->file_upload_c = join(';',
array_filter(array_merge((array)@$_REQUEST['file_upload_list'], array($file_up))));

```

- iv. Save the field
- v. Add it to Layouts → EditView and Layouts → DetailView.

Now multiple files of any type can be uploaded into any Contact or deleted by clicking the *trash* icon, as shown in [Figure 12](#). The DetailView is shown in [Figure 13](#).

Contacts: Javier Aiken

Save Cancel View Change Log

Contact Information

First Name:

Last Name: *

file upload:

- [statistics.PNG](#)
- [Specifications.doc](#)
- [invoice.pdf](#)

Account Name:

Lead Source:

Campaign:

Title:

Department:

Reports To:

Sync to Outlook®:

Do Not Call:

Assigned to:

Figure 12

Contacts: Javier Aiken

Edit Duplicate Delete Find Duplicates Manage Subscri

Name:	Javier Aiken	<input type="button" value="vCard"/>
Account Name:	Tortoise Corp 244806	
file upload:	<ul style="list-style-type: none"> statistics.PNG Specifications.doc invoice.pdf 	
Lead Source:	Public Relations	
Campaign:		
Title:	Mgr Operations	
Department:		
Reports To:		
Sync to Outlook®:	<input type="checkbox"/>	
Do Not Call:	<input type="checkbox"/>	

Figure 13

4. SugarCRM standard text field

Scenario: reproduce the standard SugarCRM text field.

The effect is the same of adding ad SugarCRM text field, but it can be used as a base to develop new features (such as AJAX spellcheck, Autocomplete or Javascript validation).

Procedure:

- i. Create a new Code field in any module with the values below

Logic Hook: Default

DB Type: text

Code:

```
if ($GLOBALS['app']->controller->action == 'EditView')
    echo '<input type="text" value="'. $bean-><field_name>.'" size="30"
    id="<field_name>" name="<field_name>"/>';
else
    echo $bean-><field_name>;
```

where the string “<field_name>” in the Code must be replaced with the Field Name (with the “_c” postfix if used in SugarCRM Studio).

- ii. Add it to any Layout

The result is the same as a standard SugarCRM text field were used.

5. Google Map automatic link

Scenario: add a link to the Google Map of the customer address on Contacts / Accounts module.

Procedure for Accounts Module:

- i. Create a new Code field in the Accounts module with the values below

Logic Hook: Default

DB Type: Non-DB

Code:

```
$search_string =
urlencode("$bean->billing_address_street $bean->billing_address_city -
$bean->billing_address_country");

echo '<a href="http://maps.google.it/maps?f=q&hl=en&geocode=&q=' .
$search_string . '&ie=UTF8&z=12&iwloc=addr" style="color:#0000FF;text-
align:left" target="_blank">Click here to view the map</a>';
```

- ii. Add it to Layouts → DetailView or EditView.

Procedure for Contacts Module:

- i. Create a new Code field in the Contacts module with the values below

Logic Hook: Default

DB Type: Non-DB

Code:

```

$search_string =
urlencode("$bean->primary_address_street $bean->primary_address_city -
$bean->primary_address_country");

echo '<a href="http://maps.google.it/maps?f=q&hl=en&geocode=&q=' .
$search_string . '&ie=UTF8&z=12&iwloc=addr" style="color:#0000FF;text-
align:left" target="_blank">Click here to view the map</a>';
    
```

- ii. Add it to Layouts → DetailView or EditView.

The Detail View for a contact should appear like the one shown in **Figure 14**

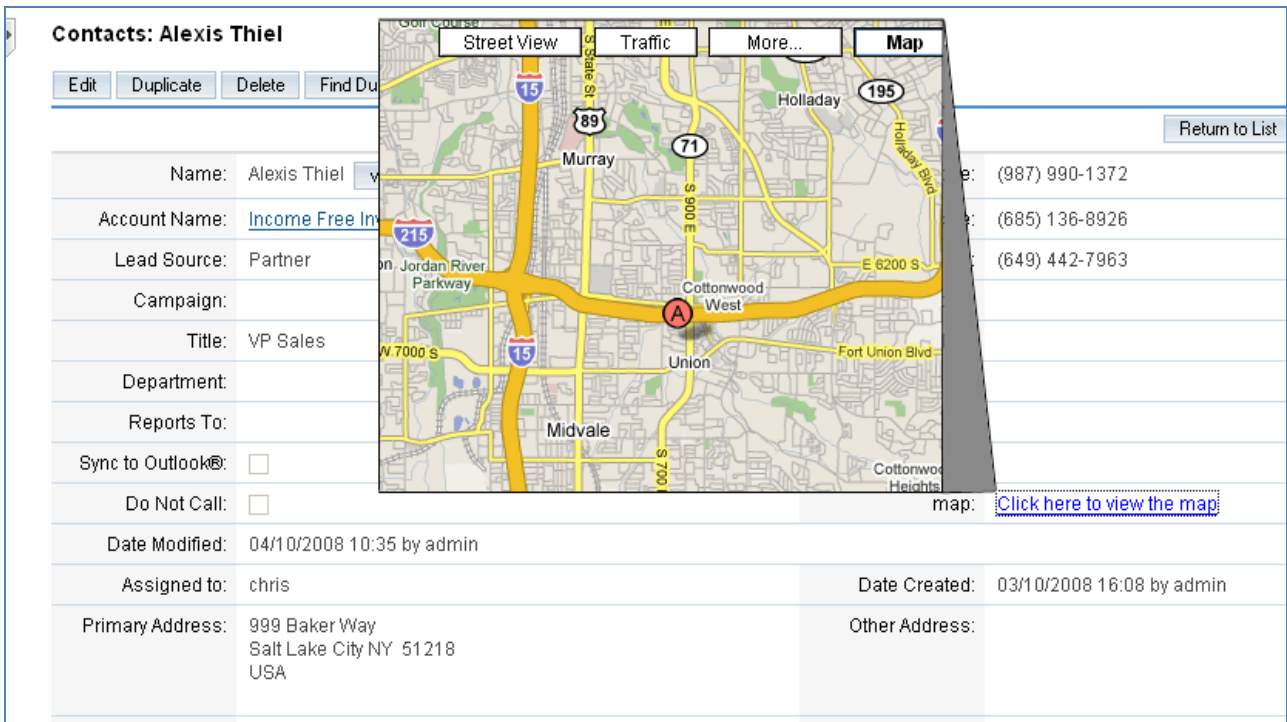


Figure 14

6. Generic AJAX Call (using YUI libraries)

Scenario: perform any AJAX call and dump results on the screen without reloading the page using the YUI libraries integrated in SugarCRM.

Code:

```

echo '
<script>
function ajax_action () {

    var callback = {
        success: function(o) {
            ... write here the js action ...
        }
    }

    var connectionObject = YAHOO.util.Connect.asyncRequest ("GET", ... write
here your URI ..., callback);
}
</script>
<input type="button" value="ajax1" onclick="ajax_action();">';

```

The AJAX URI might call for instance a .php file written by the Administrator that performs any action, with GET parameters passed as input and the result echoed and returned by the “success” function. For example the result can be Javascript alerted through the following callback:

```

var callback = {
    success: function(o) {
        alert(o.responseText)
    }
}

```

7. Generic AJAX Call (using jQuery)

Scenario: perform any AJAX call and dump results on the screen without reloading the page using the jQuery functions integrated in **Dispage Extension Manager**.

Code:

```
echo '  
<script>  
function ajax_action2 () {  
  
    $.ajax({url: "<your_uri>",  
           type: "<POST/GET>",  
           data: <your_data>,  
           success: function(data) {  
               ... write here the js action ...  
           }  
    });  
}  
  
</script>  
<input type="button" value="ajax2" onclick="ajax_action2();">;
```

To find out how the jQuery *\$.ajax* call works, follow the link <http://api.jquery.com/jquery.ajax/>

8. Autocomplete AJAX country field (New in 3.1.04)

Scenario: add the **Autocomplete feature** (natively included in dispage Extension Manager) to the “Country” field of the Billing Address in the Accounts Edit View. The autocomplete values are AJAX-loaded while typing.

Procedure:

- i. Create a Code field in the Accounts module with the values below

Logic Hook: after_ui_frame

DB Type: Non-DB

Code:

```
if ($GLOBALS['app']->controller->action == 'EditView')
    echo '
<script language="javascript">
    $(document).ready(function () {
        $("#billing_address_country").autocomplete({
            "source" : "customAJAXCalls.php?call=Accounts_countries_cb"
        });
    });
</script>
';
```

- ii. Create an AJAXCallback field in with

Field Name: countries_cb

Code:

```
if (!@$_REQUEST["term"]) die('[]');

require_once('include/utils.php');
require_once('include/SugarTheme/SugarTheme.php');
require_once('include/language/en_us.lang.php');

$found = array();

foreach ($app_list_strings['countries_dom'] as $c) {
    if (stripos($c, $_REQUEST["term"]) === 0) $found[] = "'".$c.'"';
}
echo '[' . join(', ', $found) . '];
```

No need to add any of the fields above to the EditView Layout.

Figure 15 is a screenshot taken while typing in Account’s Country field of Billing Address.

Figure 15

9. Generic SugarCRM DB / External DB Query

Scenario: connect to SugarCRM DB or any external DB, execute a query (possibly containing any data of the current field through the “\$bean” variable) and echo the results. MySQL DB Engine is assumed to be used in this and the next sample.

Code:

```
global $connection_code_field;

if (!isset($connection_code_field)) {
    $connection_code_field = mysql_connect("<db_host>", "<db_user>",
"<db_password>");
    mysql_select_db("<db_name>", $connection_code_field);
}
$sql = "<db_query>";

$res = mysql_query($sql, $connection_code_field);
if (mysql_num_rows($res))
    $value = mysql_result($res, <id_field>);

echo $value;
```

where <db_host>, <db_user>, <db_password>, <db_name>, <db_query> and <id_field> must be properly set. When connecting to the SugarCRM DB, connection parameters can be retrieved automatically like in the following example.

10. Contact List in Account Module

Scenario: add a list of clickable related contacts for each account.

Procedure:

iii. Create a new Code field in the Accounts module with the values below

Logic Hook: Default

DB Type: Non-DB

Code:

```
global $db;

$sql = "SELECT ac.contact_id as id, CONCAT(c.first_name, ' ', c.last_name) as name
FROM contacts c JOIN accounts_contacts ac ON c.id = ac.contact_id WHERE
ac.account_id = '$bean->id'";

$res = $db->query($sql);
$values = array();
while($value = $db->fetchByAssoc($res))
    $values[] = "<a
href='index.php?module=Contacts&return_module=Accounts&action=DetailView&record='.
$value['id']. '>'>".$value['name']. "</a>";
echo join($values, '<br/>');
```

iv. Add it to Layouts → DetailView or ListView.

A sample of List View for the field is shown in **Figure 16**

Account List					
Select	Delete	Export	Merge Duplicates	Compose Email	Selected: 0
					(1 - 20 of 50)
<input type="checkbox"/>	Account Name	City	Type	contactslst	User
<input type="checkbox"/>	Dirt Mining Ltd 401795	Los Angeles	Customer	Priscilla Meldrum Jaime Risley Demarcus McCreary Constance Ludwig Moises Cash Reyes Scalf Wilton Waites	chris <input type="checkbox"/>
<input type="checkbox"/>	Jungle Man Inc 338625	Salt Lake City	Customer	Kristi Alcaraz Ambrose Ladner Damion Knights Bud Goudeau Lionel Alford Imelda Karpinski Bennett Arline	chris <input type="checkbox"/>
<input type="checkbox"/>	C Nelson Inc 762191	Kansas City	Customer	Marvin Berta Adelaide Kalman Jed Winbush	chris <input type="checkbox"/>
<input type="checkbox"/>	Jungle Man Inc 276547	San Francisco	Customer	Eddie Swims Rosemarie Trew Jocelyn Watford	chris <input type="checkbox"/>
<input type="checkbox"/>	Income Free Investing LP 400013	Alabama	Customer	Charles Work Iving Rome Winnie Boggs Virginia Winkel Rita Unger Sid Somerville Alexis Thiel	chris <input type="checkbox"/>

Figure 16

Webography

Enhanced Studio on Dispage website

Enhanced Studio 3.1 has its own place among the Dispage Extensions at Dispage Website:

<http://www.dispage.com/index.php/products/enhanced-studio>

(New) A useful wiki section can be found at:

[http://www.dispage.com/wiki/Category:Enhanced Studio](http://www.dispage.com/wiki/Category:Enhanced_Studio)

Enhanced Studio on SugarForge

The official Enhanced Studio page on SugarForge can be found at

http://www.sugarforge.org/frs/?group_id=580

The history of all the releases is available at the download section, as well as a dedicated **Forum Area, Documentation, Reviews** and more.

Enhanced Studio on SugarExchange (2.0 and 3.0 rel.)

Visit the Enhanced Studio page on the official Marketplace of Sugar Products at

http://www.sugarexchange.com/product_details.php?product=580

Before purchasing the product, **please read carefully the EULA license:**

http://www.sugarforge.org/frs/download.php/5544/FULL_VERSION_LICENSE.txt

Enhanced Studio on Sugar Forums

(<http://www.sugarcrm.com/forums/>)

Here's a collection of the most suggestive posts related to Enhanced Studio

- (ENG) Enhanced Studio : A new tool to customize Sugar
<http://www.sugarcrm.com/forums/showthread.php?t=33140>
- (ENG) Enhanced Studio Support
<http://www.sugarcrm.com/forums/showthread.php?t=39370>
- (ENG) Enhanced Studio 3.0
<http://www.sugarcrm.com/forums/showthread.php?t=48669>
- (ENG) Enhanced Studio and Logic Hook Programming Interface (support)
<http://www.sugarcrm.com/forums/showthread.php?t=58251>
- (ENG) Need Help
<http://www.sugarcrm.com/forums/showthread.php?t=58341>
- (ENG) include custom CSS or JS files in theme
<http://www.sugarcrm.com/forums/showthread.php?t=57261>
- (ESP) Insertar foto en formulario
<http://www.sugarcrm.com/forums/showthread.php?t=57782>
- (DEU) Modulfremde Felder
<http://www.sugarcrm.com/forums/showthread.php?t=57943>
- (ENG) Auto Populate?
<http://www.sugarcrm.com/forums/showthread.php?t=57519>
- (ENG) Custom code for modules
<http://www.sugarcrm.com/forums/showthread.php?t=55763>
- (ENG) Setting the right date
<http://www.sugarcrm.com/forums/showthread.php?t=55301>
- (ENG) Studio plugin to version 5.5
<http://www.sugarcrm.com/forums/showthread.php?t=55468>
- (ENG) What's the best way to add ajax/javascript functionality to a listview field?
<http://www.sugarcrm.com/forums/showthread.php?t=54994>

- (ENG) Geocoding project - populate returned data
<http://www.sugarcrm.com/forums/showthread.php?t=55317>
- (ENG) Adding multiple Fields similiar to the Email +
<http://www.sugarcrm.com/forums/showthread.php?t=52293>
- (DEU) Telefon Länder Vorwahl
<http://www.sugarcrm.com/forums/showthread.php?t=54612>
- (ENG) How to add HTML code?!
<http://www.sugarcrm.com/forums/showthread.php?t=54539>
- (DEU) Felder Kontakt aus Firma übernehmen
<http://www.sugarcrm.com/forums/showthread.php?t=53565>
- (ENG) Calculated Fields
<http://www.sugarcrm.com/forums/showthread.php?t=51636>
- (ENG) view total relationships in ListView
<http://www.sugarcrm.com/forums/showthread.php?t=53550>
- (ENG) data from 1 field updates another field
<http://www.sugarcrm.com/forums/showthread.php?t=53075>
- (ENG) Apply Javascript to a field
<http://www.sugarcrm.com/forums/showthread.php?t=52886>
- (ENG) No permission to access php script
<http://www.sugarcrm.com/forums/showthread.php?t=55038>
- (ENG) Need to customize OpenSales
<http://www.sugarcrm.com/forums/showthread.php?t=53932>
- (ENG) Sugar Bean Fields
<http://www.sugarcrm.com/forums/showthread.php?t=49353>
- (ENG) add javascript in head tags
<http://www.sugarcrm.com/forums/showthread.php?t=43880>
- (ENG) Arguments passing to extended controller
<http://www.sugarcrm.com/forums/showthread.php?t=52388>
- (ENG) call external web with parameters
<http://www.sugarcrm.com/forums/showthread.php?t=52189>

- (ENG) State Abbreviation
<http://www.sugarcrm.com/forums/showthread.php?t=52564>
- (ENG) Enhanced Studio: customize Sugar
<http://www.sugarcrm.com/forums/showthread.php?t=52166>
- (ENG) Related Fields / How do I change whats Displayed?
<http://www.sugarcrm.com/forums/showthread.php?t=52028>
- (ITA) Gestione parco macchine
<http://www.sugarcrm.com/forums/showthread.php?t=50181>
- (ENG) Calculated Fields
<http://www.sugarcrm.com/forums/showthread.php?t=51636>
- (ENG) Format Field
<http://www.sugarcrm.com/forums/showthread.php?t=51698>
- (ENG) Assigning Co-owner for Account - Multiple Assigned User
<http://www.sugarcrm.com/forums/showthread.php?t=49860>
- (ENG) Disable field
<http://www.sugarcrm.com/forums/showthread.php?t=51455>
- (CHI) SugarCRM 插件介绍
<http://www.sugarcrm.com/forums/showthread.php?t=50380>
- (ENG) custom field with condition
<http://www.sugarcrm.com/forums/showthread.php?t=49876>
- (DEU) Werte werden angezeigt -->DB
<http://www.sugarcrm.com/forums/showthread.php?t=50249>
- (ENG) How to add field from another module?
<http://www.sugarcrm.com/forums/showthread.php?t=38092>
- (DEU) Wert Currency in Enhanced Studio
<http://www.sugarcrm.com/forums/showthread.php?t=50077>
- (DEU) Einfache Berechnung
<http://www.sugarcrm.com/forums/showthread.php?t=50074>
- (DEU) Verkaufschancen
<http://www.sugarcrm.com/forums/showthread.php?t=49478>

- (DEU) Kontaktformular mit Foto
<http://www.sugarcrm.com/forums/showthread.php?t=49807>
- (ENG) I want to delete record not the relationship
<http://www.sugarcrm.com/forums/showthread.php?t=49729>
- (ENG) why can't I invoke external webservice in sugarcrm?
<http://www.sugarcrm.com/forums/showthread.php?t=49271>
- (DEU) In Contacts ein Feld aus Accounts anzeigen
<http://www.sugarcrm.com/forums/showthread.php?t=49461>
- (ENG) Enhanced Studio - Logic_hooks not working
<http://www.sugarcrm.com/forums/showthread.php?t=48801>
- (ENG) HTTP Error: no data present after HTTP headers
<http://www.sugarcrm.com/forums/showthread.php?t=49189>
- (DEU) Zwei kleine Fragen (Link und Optik)
<http://www.sugarcrm.com/forums/showthread.php?t=47943>
- (ENG) Enhanced Studio: MySQL or MS SQL ?
<http://www.sugarcrm.com/forums/showthread.php?t=48936>
- (DEU) Feld aus Accounts in Kontakt anzeigen lassen
<http://www.sugarcrm.com/forums/showthread.php?t=48465>
- (ENG) upgrade from 5.0 to 5.2.0e
<http://www.sugarcrm.com/forums/showthread.php?t=48703>
- (ENG) Enhanced Studio Logic_hooks modification
<http://www.sugarcrm.com/forums/showthread.php?t=48561>
- (DEU) Feld aus Accounts in Kontakt anzeigen lassen
<http://www.sugarcrm.com/forums/showthread.php?t=48465>
- (ENG) Enhance Studio External DB Query
<http://www.sugarcrm.com/forums/showthread.php?t=48264>
- (ENG) I need help with the COde Field in the Enhanced Studio
<http://www.sugarcrm.com/forums/showthread.php?t=46540>
- (ENG) Enhanced Studio code filed value in database
<http://www.sugarcrm.com/forums/showthread.php?t=48212>

- (ENG) Create add email address field with custom code
<http://www.sugarcrm.com/forums/showthread.php?t=47008>
- (ENG) Enhanced Studio - Unable to Edit after Deployment
<http://www.sugarcrm.com/forums/showthread.php?t=47448>
- (ENG) how i can save all field that have created by custom code
<http://www.sugarcrm.com/forums/showthread.php?t=47377>
- (ENG) add two values and disply result in third field before save
<http://www.sugarcrm.com/forums/showthread.php?t=47469>
- (ENG) calculate lots of fields
<http://www.sugarcrm.com/forums/showthread.php?t=47315>
- (ENG) Immediately calculated fields
<http://www.sugarcrm.com/forums/showthread.php?t=46719>
- (ENG) I need help with the COde Field in the Enhanced Studio
<http://www.sugarcrm.com/forums/showthread.php?t=46537>
- (ENG) Additional fields in accounts
<http://www.sugarcrm.com/forums/showthread.php?t=46698>
- (DEU) Listenansicht Beziehung Subpanel
<http://www.sugarcrm.com/forums/showthread.php?t=46127>
- (ENG) I need help with the COde Field in the Enhanced Studio
<http://www.sugarcrm.com/forums/showthread.php?t=46534>
- (ENG) need help in enhanced studio(code field)
<http://www.sugarcrm.com/forums/showthread.php?t=46461>
- (ENG) how to implement google map in sugar
<http://www.sugarcrm.com/forums/showthread.php?t=46445>
- (DEU) Wer kann mir wg. Code für mit enhanced studio ne Idee geben?!
<http://www.sugarcrm.com/forums/showthread.php?t=45110>
- (ENG) How to integrate Strikelron US Address Verification Web service into SugarCRM
<http://www.sugarcrm.com/forums/showthread.php?t=45587>
- (ENG) How to consume Strikelron US Address Verification Web service within SugarCRM (cont.)
<http://www.sugarcrm.com/forums/showthread.php?t=44537>

- (ENG) Add account information to cases detail screen
<http://www.sugarcrm.com/forums/showthread.php?t=44857>
- (DEU) Änderung der Ansichten
<http://www.sugarcrm.com/forums/showthread.php?t=45395>
- (ENG) Need coding help for enhanced studio...
<http://www.sugarcrm.com/forums/showthread.php?t=45136>
- (DEU) berechnung von 2 felder in ein neues schreiben
<http://www.sugarcrm.com/forums/showthread.php?t=45006>
- (DEU) Aufbau der View
<http://www.sugarcrm.com/forums/showthread.php?t=44303>
- (DEU) Eigenen PHP Code einfügen
<http://www.sugarcrm.com/forums/showthread.php?t=42295>
- (ENG) How to consume ZIP Code Information Web service from Strikelron within SugarCRM
<http://www.sugarcrm.com/forums/showthread.php?t=44150>
- (ENG) empty list view after adding custom fields
<http://www.sugarcrm.com/forums/showthread.php?t=42519>
- (ENG) add javascript in head tags
<http://www.sugarcrm.com/forums/showthread.php?t=43880>
- (DEU) Suche Checkbox
<http://www.sugarcrm.com/forums/showthread.php?t=43718>
- (ESP) Campo Calculado
<http://www.sugarcrm.com/forums/showthread.php?t=43621>
- (ENG) After upgrading to 5.2 , Studio isnt functioning
<http://www.sugarcrm.com/forums/showthread.php?t=41902>
- (ENG) Enhanced Studio problem with onclick
<http://www.sugarcrm.com/forums/showthread.php?t=41902>
- (ENG) A free version of Enhanced Studio? That do the same job?
<http://www.sugarcrm.com/forums/showthread.php?t=42010>
- (ENG) After upgrading to 5.2 , Studio isnt functioning
<http://www.sugarcrm.com/forums/showthread.php?t=41902>

- (ENG) Enhanced Studio problem with onclick
<http://www.sugarcrm.com/forums/showthread.php?t=42010>
- (ENG) Simple Calculations
<http://www.sugarcrm.com/forums/showthread.php?t=42612>
- (ENG) Enhanced Studio 2.1 available
<http://www.sugarcrm.com/forums/showthread.php?t=37595>
- (ENG) How to get the user logged??
<http://www.sugarcrm.com/forums/showthread.php?t=38258>
- (ENG) [Tutorial] Howto use ajax with sugar
<http://www.sugarcrm.com/forums/showthread.php?t=36226>
- (ESP) Usando ajax con sugar
<http://www.sugarcrm.com/forums/showthread.php?t=36182>
- (ENG) Evaluate Fields from other
<http://www.sugarcrm.com/forums/showthread.php?t=38053>
- (POR) Adicionar campo com imagem
<http://www.sugarcrm.com/forums/showthread.php?t=37994>
- (ENG) Zend_Framework Thoughts?
<http://www.sugarcrm.com/forums/showthread.php?t=37262>
- (ENG) Show iframe with URL parameters on Account DetailView page?
<http://www.sugarcrm.com/forums/showthread.php?t=35318>
- (ENG) Relate and Auto populate fields
<http://www.sugarcrm.com/forums/showthread.php?t=35905>
- (ENG) Maps for Contacts / Accounts addresses with Enhanced Studio
<http://www.sugarcrm.com/forums/showthread.php?t=33792>
- (ENG) Create DB storable custom fields with Enhanced Studio 2.0
<http://www.sugarcrm.com/forums/showthread.php?t=34736>
- (ENG) manually remove enhanced search/studio
<http://www.sugarcrm.com/forums/showthread.php?t=34991>
- (ENG) Add a contact field with tasks count
<http://www.sugarcrm.com/forums/showthread.php?t=34833>

- (DEU) add contact field to activities schedule call
<http://www.sugarcrm.com/forums/showthread.php?t=34500>
- (ENG) Enhanced Studio - losing subpanels and fields in details view
<http://www.sugarcrm.com/forums/showthread.php?t=34573>
- (ENG) custom field of html type with dynamic querystring???
<http://www.sugarcrm.com/forums/showthread.php?t=33745>
- (ENG) enhanced studio and external Databases
<http://www.sugarcrm.com/forums/showthread.php?t=34550>
- (ENG) Recommend any free modules/dashlets?
<http://www.sugarcrm.com/forums/showthread.php?t=34395>
- (DEU) Datensatz ID im Detail View
<http://www.sugarcrm.com/forums/showthread.php?t=33954>
- (ENG) Not sure what this is called...
<http://www.sugarcrm.com/forums/showthread.php?t=33779>
- (ENG) customize the way a field is displayed
<http://www.sugarcrm.com/forums/showthread.php?t=33791>
- (ENG) Custom fields and cache problems
<http://www.sugarcrm.com/forums/showthread.php?t=33734>