

liberty of shameless customization of Web Forms and fields

All forms are editable through INI files. By default, You will find en_US.xxxxx.ini (xxxxx representing correspondent module name) file located at application/modules/\$ModuleName/forms/source/en_US.xxxxxForm.ini (\$ModuleName represents your presently working module e.g hotel , property, vacationrentals etc). Each paragraph is responsible for building a single field and it's attributes. You can edit, remove or comment out a paragraph to wipe off a field. Form's source INI file is self explanatory and requires some kind of knowledge in web form management. We would strongly recommend you to comment out (NOT to delete) line, just by adding ; (colon) sign as prefix to each line, so that you can still survive by removing ; (colon) sign for any un-authorized changes. Both delete and comment out results same output but comment out is preferred because you will still have a chance to come back and survive in cause of any wrong doing.

Managing Forms through INI files:

We assumed, you have already [learned and created a dynamic web form using our Application](#). Now we will explain how you to join an existing web form(s) to a listing. Please [Click Here](#) to know how to create a new custom web form or see [Video Tutorials](#) here.

Now, we will show, how can we edit, delete and play with web forms and form's fields through predefined ini files. All forms are editable through .ini files. You will find en_US.xxxxx.ini (xxxxx representing correspondent module name) file located at application/modules/\$YourModuleName/forms/source/en_US.xxxxxForm.ini (\$YourModuleName is your presently working module e.g hotel , property, vacationrentals etc). Each paragraph is responsible for building a single field.

Few Locations of Form's Source:

Property : The forms setting are located
application/modules/Property/forms/source/en_US.ProductForm.ini .

Hotel : The forms setting are located
application/modules/Hotels/forms/source/en_US.ProductForm.ini

Vacation Rentals: The forms setting are located
application/modules/Vacationrentals/forms/source/en_US.VacationrentalsForm.ini

Tours Mgmt : The forms setting are located
application/modules/Tours/forms/source/en_US.ProductForm.ini

Autos Mgmt : The forms setting are located
application/modules/Autos/source/en_US.ProductForm.ini.

Membership Form : The forms setting are located
application/modules/Members/source/en_US.ProductForm.ini .

And so on.....

By default, You will find en_US.xxxxx.ini (xxxxx representing correspondent module name) file located at application/modules/\$ModuleName/forms/source/en_US.xxxxxForm.ini (\$ModuleName represents your presently working module e.g hotel , property, vacationrentals etc).

STEP-11 This tutorial is explain how to manage & customize details listing interface settings.

This Module as a Example

This is Details Listing Interface Folder Path

This is Details Listing Interface .ini File

Edit with Dreamweaver

NOTES ! LOCATION PATH : /<\$ServerPath>/application/modules/<\$Your Module>/forms/source/en_US.ProductForm.ini

Here we choose customizing the forms of property modules just **as example**. This tutorial is applicable to any modules you want because the process is generic to all modules.

Open the .ini file using your preferred html editor and follow the below images. Each paragraph

is responsible for building a single field and it set required attributes. The en_US.xxxxxxForm.ini file is self explanatory and requires some kind of knowledge in web form management.

Samples managing forms through .ini file

STEP-12 This tutorial is explain how to manage & customize details listing interface settings.

```

1 [property]
2 ;Add an Property Name element
3 property.elements.property_name.type = "text"
4 property.elements.property_name.options.size = 50
5 property.elements.property_name.options.required = true
6 property.elements.property_name.options.label = "Property Name:"
7 property.elements.property_name.options.title = "Enter Property Name"
8 property.elements.property_name.options.class = "ui-widget-content ui-corner-all"
9 property.elements.property_name.options.value = ""
10 property.elements.property_name.options.filters.trim.filter = "StringTrim"
11 property.elements.property_name.options.filters.StripTags.filter = "StripTags"
12
13 ; Group element
14 property.elements.group_id.type = "select"
15 property.elements.group_id.options.size = 1
16 property.elements.group_id.options.class = "property_select_group ui-widget-content"
17 property.elements.group_id.options.label = "Select Group :)"
18 property.elements.group_id.options.title = "Select Property Group"
19 property.elements.group_id.options.required = true
                
```

STEP-14 This

```

[property]
;Add an Property Name element
property.elements.property_name.type = "text"
property.elements.property_name.options.size = 50
property.elements.property_name.options.required = true
property.elements.property_name.options.label = "Property Name:"
property.elements.property_name.options.title = "Enter Property Name"
property.elements.property_name.options.class = "ui-widget-content ui-corner-all"
property.elements.property_name.options.value = ""
property.elements.property_name.options.filters.trim.filter = "StringTrim"
property.elements.property_name.options.filters.StripTags.filter = "StripTags"

; Group element
property.elements.group_id.type = "select"
property.elements.group_id.options.size = 1
property.elements.group_id.options.class = "property_select_group ui-widget-content"
property.elements.group_id.options.label = "Select Group :)"
property.elements.group_id.options.title = "Select Property Group"
property.elements.group_id.options.required = true
                
```

NOTES! LOCATION PATH : /<ServerPath>/application/modules/<Your Module>/forms/source/en_US.ProductForm.ini

NOTES!

How Forms are rendering to Frontend

Now, we know how to manage web forms and it's fields. Let us know show you, how the changes reflects to frontend (public viewable) interface. Here is a short explanation illustrating how program parse form's ini files:

property.elements.property_name.type = "text"

Splitting above command as :

property = Modules name

elements = elements / field of a form

property_name = field name

type = attributes of a field / element

= **"text"** is a Changeable Value

See the following images for better illustration

Samples managing forms through .ini file

STEP-1 The indicated box shown below are details listing interface label and values.

STEP-2 The indicated box shown below are details listing interface labels and values.

STEP-3 The indicated box shown below are details listing interface labels and values.

STEP-1 The indicated box shown below are details listing interface labels and values.

The screenshots show a property listing interface with annotations for labels and values. The first screenshot shows a property listing for '11501 Shadestone Terrace' with a price of £494. The second screenshot shows a 'DETAILS' section with various fields. The third screenshot shows a 'REALTOR REPRESENTATION' form with fields for 'Presented by', 'Package Name', 'Company Name', 'Contact Info', and 'Web Site'. The fourth screenshot shows a property listing for '11501 Shadestone Terrace' with various details.

How Forms are rendering to Backend

Let us know show you, how the changes reflects to back-end(Admin or after login) interface. You can manage form more efficiently through add.phtml or edit.phtml block file to manage forms. A Block is basically a data resource for Views. It uses the sites Models, grabs the data, makes any necessary adjustments, and then makes that data available to a View. For instance, when you override / create a Block that calls a product collection, that product collection is pulling some products' data from a Model. The Block is where you create such a product collection and implement the product collection's methods, such as adding filters or ordering the products in a

certain way. That product collection is then made available to your View so you can display the products on a page.

Here is where the backend form blocks are located to play with.

Managing Admin Add / Edit Blocks

STEP-16 This tutorial is explain how to delete details listing interface field.

This Module as a Example

This is backendpro Folder Path

Edit with Dreamweaver

This is backendpro add.phtml File

NOTES ! LOCATION PATH : $/$$.$</math>/application/modules/$.$</math>/views/scripts/backendpro/add.phtml

STEP-18 This

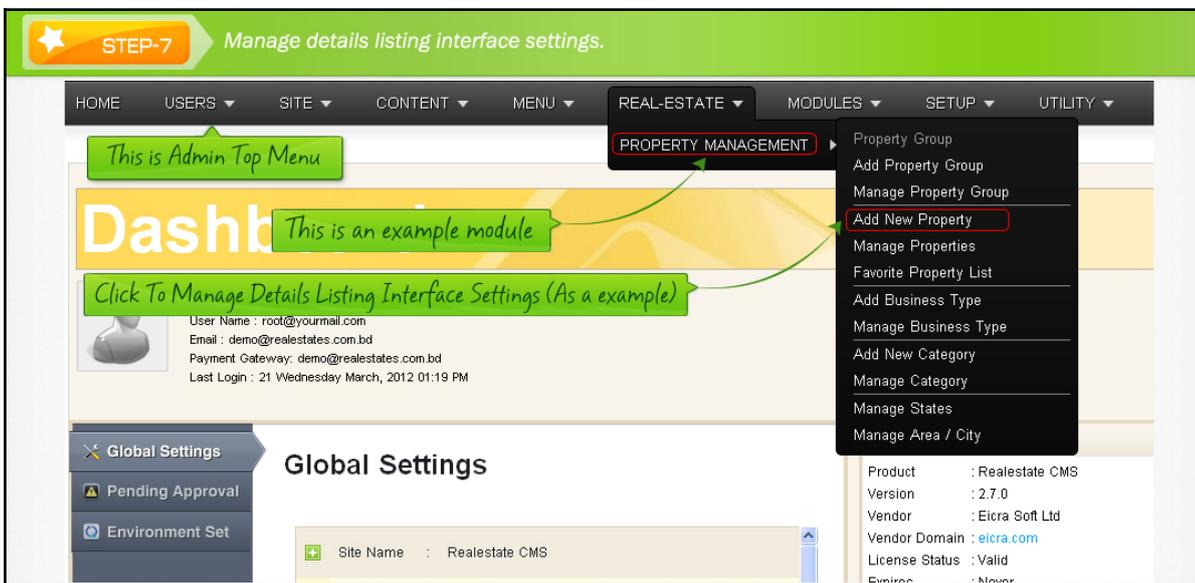
This Module as a Example

This is backendpro Folder Path

NOTES ! LOCATION PATH :

Now, we know how to manage web forms and it's fields. Let us know show you, how the changes reflects to backend (admin view) interface. Here are few images illustrating how program parse form's ini files:

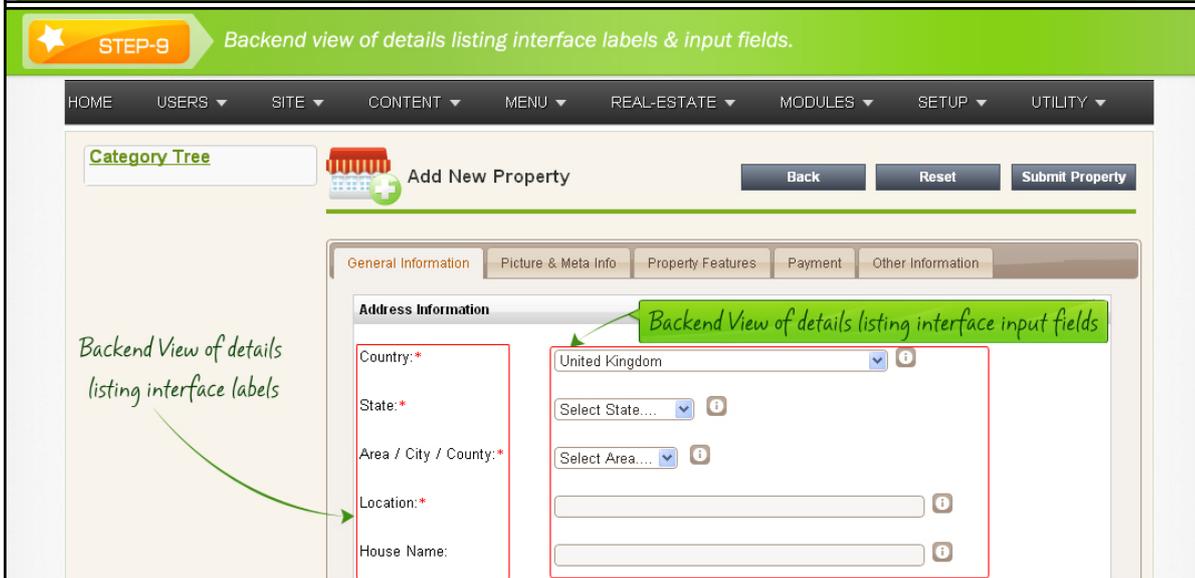
Managing Admin Add / Edit Blocks



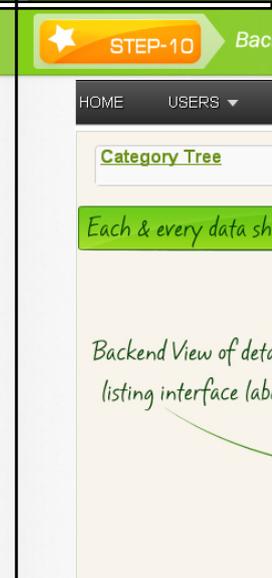
NOTES ! Dashboard => Top Menu => Select - <\$YOUR MODULE> Than - Property Management (As a example) => And Now Click- Add New Property (As a example) => to start manage details listing interface settings.



NOTES ! Dashboard => Top Add New Property



NOTES ! Dashboard => Top Menu => Select - <\$YOUR MODULE> Than - Property Management (As a example) => And Now Click- Add New Property (As a example) => to start manage details listing interface settings.



NOTES ! Dashboard => Top Add New Property

How to Remove / Delete new field(s):

We would not recommend deleting field(s) or tabs (containing list of fields) because improper deletion this may crash your search and program if any of key fields is missed. This will be one of **HIGH RISK** changes which is strongly discouraged. If you still want to remove field(s), please apply **EXTREME** cautions before deleting because every fields are pointed to RDBMS fields storing it's value to the database, furthermore fields maintains many relational links with external files followed by it's referential instance.

By default the fields are mostly divided into two parts. There are:

01) Core / Key fields

02) Descriptive / informative fields

Core / Key Fields :

Most of the cases the key Fields are mandatory with few exceptions and used as search parameter. These key fields maintain many relational links with external files followed by it's referential instance and values MUST be set in database. For example price, room number, properly owner / agent, size , availability calendar etc and you will understand its importance just using common sense. Most of these fields are used for searching / filtering at frontend, therefore system cannot afford removing key fields. If you delete / remove any key field, it may end up resulting destruction and system may crush down.

Descriptive / informative fields :

These fields carry less impotence in comparison with key fields. For example address, features, brochure, maps, policies etc, where these fields are never used in search parameter. The values of these fields are option and database allows you to go with empty value. You can remove these fields with extreme caution.

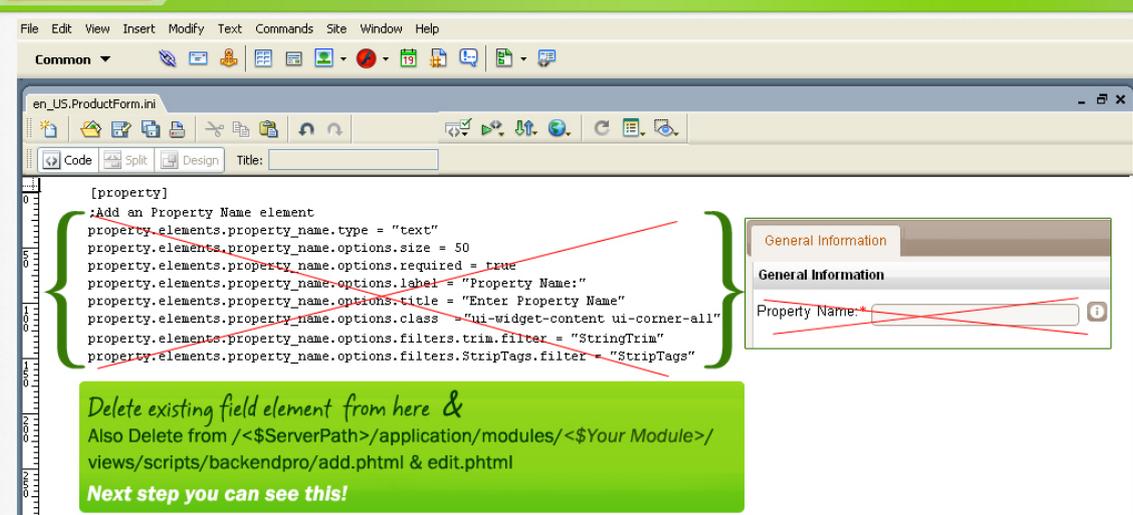
Removing Field(s) :

All forms are editable through .ini files. You will find en_US.xxxxx.ini (xxxxx representing correspondent module name) file located at application/modules/\$YourModuleName/forms/source/en_US.xxxxxForm.ini (\$YourModuleName is your presently working module e.g hotel , property, vacationrentals etc). Each paragraph is responsible for building a single field. You can remove or comment out a paragraph to wipe off a field. en_US.xxxxxForm.ini file is self explanatory and requires some kind of knowledge in web form management. We would strongly recommend you to comment out (NOT to delete) line, just by adding ; (colon) sign as prefix to each line, so that you can still survive by removing ; (colon) sign for any un-authorized changes. Both delete and comment out results same output but comment out is preferred because you will still have a chance to come back and survive in cause of any wrong doing.

See the image shown below

Managing Admin Add / Edit Blocks

STEP-15 This tutorial is explain how to delete details listing interface field.

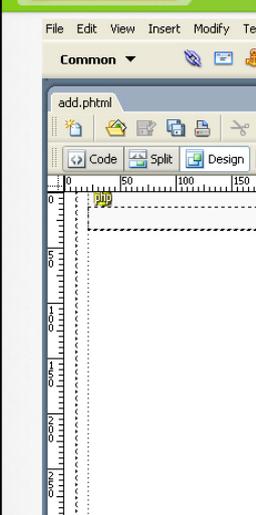


```
[property]
: Add an Property Name element
property.elements.property_name.type = "text"
property.elements.property_name.options.size = 50
property.elements.property_name.options.required = true
property.elements.property_name.options.label = "Property Name:"
property.elements.property_name.options.title = "Enter Property Name"
property.elements.property_name.options.class = "ui-widget-content ui-corner-all"
property.elements.property_name.options.filters.trim.filter = "StringTrim"
property.elements.property_name.options.filters.StripTags.filter = "StripTags"
```

Delete existing field element from here &
Also Delete from </\$ServerPath>/application/modules/<\$Your Module>/
views/scripts/backendpro/add.phtml & edit.phtml
Next step you can see this!

NOTES ! LOCATION PATH : </\$ServerPath>/application/modules/<\$Your Module>/forms/source/en_US.ProductForm.ini

STEP-17 This



NOTES ! LOCATION PATH :

Revision: 11

FAQ -> liberty of shameless customization of Web Forms and fields

<http://www.httpsdoc.com/entry/108/>