

# Forms multipages

## OVERVIEW

INTEGRATED DEVELOPPEMENT FOR FORMS WITH PHP, INCLUDING ADVANCED PHP TOOLS .

It is the development environment php5 for process forms with more pages. It consists of an integrated set of tools. It allows personalized controls on each fields by using callback functions.

### OBJECT:

- Reduce code HTML
- Travel information from screen to screen
- Custom forms fields
- Construct, fit forms
- Allow interface with user control for fields

## Forms multipages

- 1 - PRODUCT
- 2 - SYNTAX FOR SCRIPT PAGE
- 3 - ERROR HANDLING
- 4 - THE CLASS OVERVIEW
- 4 - 1 CLASS 'OBJ\_DAEMONFORM'
- 4 - 2 CLASS 'OBJ\_FORM'
- 4 - 2 - 1 FIELDS
- 4 - 2 - 1 - 1 ATTRIBUTES FIELD
- 4 - 2 - 1 - 2 FUNCTIONS
- 4 - 2 - 2 DESCRIPTION FUNCTIONS

## Forms multipages

### 1 - PRODUCT

You must run the script `Sample_More_Forms.php` for starting your application.

#### CALLING PARAMETERS :

<i>P_Proc</i> = string	String for identify application
<i>P_ScriptAfterValid</i> = string (optional)	String full filename user to be called when form is validated A script to send the results of form is available (" <code>UserSendEmail.php</code> ")
<i>P_FuncUserControl</i> = string (optional)	String full filename user script which contains the callback functions These functions are called after field button submit. They allows a control personalized It contains also the functions to be called where field link error no exists.
<i>P_PagesEnum</i> = string	String contains list of pages for each page: title page = full filename page = option  option => 'on' or 'off' or 'opt' 'on' for page obligatory 'opt' for page optional 'off' for page hidden or removed  character ' ' is the separator between each pages.  example: " title_page_1 = full_filename_page_1 = on   title_page_2 = full_filename_page_2 = opt   title_page_3 = full_filename_page_3 = on

These scripts below are writed for the demonstration:

#### Scripts *P\_ScriptAfterValid*

- "`UserValidForms.php`"
- " `UserSendEmail.php` "

#### Scripts *P\_FuncUserControl*

- "`CallUserFunc_Forms`"
- "`Call UserFunc_AdrEmail.php`"
- "`Call UserFunc_Crypt`"

If *P\_PagesEnum* is not present, the standards pages of demonstration are used.

## Demonstration

### 1 - Forms (more pages)

Three pages: Page\_1.php, Page\_2.php, Page\_3.php

#### Script *P\_ScriptAfterValid UserValidForms.php*

- Display array the results of the form

#### Script *P\_FuncUserControl callFuncUser\_Forms.ph* proceeds controls below:

- CheckUpt field 'Prenom' [first page] is different of 'paul'
- Assign, if field 'Nom' [first page] is 'Laur' and field 'Prenom' is blank, 'Prenom' to 'Pierre'
- CheckUp fields 'Select\_Liste\_1' [second page] with 'Select\_Liste\_2' [Page 2] and alert if if 'Select\_2' and 'choix\_2'
- CheckUp fields 'Select\_CheckBox\_1' [second page] no field link error begin with 'x' and 'Select\_Liste\_2' [Page 2]
- CheckUp and alert if fields 'Prenom' [first page] begin with 'x' and 'Select\_Liste\_2' [second page] is 'Select\_3'

### 2 - Forms (send mail)

Two pages: GetAdrEmail.php, Contact.php

#### Script *P\_ScriptAfterValid UserSendEmail.php*

- Send the results of the form (mail)

#### Script *P\_FuncUserControl callFuncUser\_AdrEmail.php* proceeds controls below:

- CheckUpt field 'Adr\_Email' [first page]
- CheckUpt field 'Key\_for\_Crypt' [first page]
- CheckUpt field 'Email\_Crypted' [first page]

### 3 - Encoding -Decoding)

One page: Crypt.php

#### Script *callFuncUser\_Crypt.php* proceeds controls below:

- CheckUpt field ' Key\_for\_Cryptl' [first page]
- CheckUpt field ' Texte\_Crypted' [first page]
- CheckUpt field ' Texte\_To\_Crypt' [first page]

## 2 - SYNTAX FOR SCRIPTS FORM

### It is necessary to have a script for each page of the form

#### - a script must always begin with command below

(after <head>):

```
<?php
//Get object Obj_DaemonForm and get Obj_Form for the current page
$ObjForm = _GetDaemon(TRUE/FALSE->)FormsPage(_PAGE_);
// TRUE for mode debugging. Default, /FALSE is positionned
// _PAGE_ contains current page
if (!$ObjForm->AlreadyInitialized()) //test if already initialized
{ //Define fields (cf Objects for forms)

}
?>
<!--affichage du titre de la page --!>
<title><?php echo _GetDaemon()->GetTitlePage(_PAGE_); ?></title>
```

#### - After <body>:

```
<?php echo _GetDaemon()->StartForm(_PAGE_,[enctype]); ?>
example: enctype is optional ["multipart/form-data"] / ["text/plain"]
<?php echo _GetDaemon()->StartForm(_PAGE_,"multipart/form-data"); ?>
```

#### - within body view of fields

```
<?php echo $ObjForm->TextHTML("Field_Name"); ?> for get and view field
<?php echo $ObjForm->TextHTMLMsg("Field_Name"); ?> for view field link error (optional)
etc ...
```

#### - End body Forms

```
<?php echo _GetDaemon()->EndForm(); ?>
</body>
</html>
```

## 3 - ERROR HANDLING

If a error is encountered during the execution, a BackTrace is writed on log file 'Trace.log' [Folder '/Formulaire']. This BackTrace allows know all functions called, before error, and their values parameters. This BackTrace write on log file a dump of all variables used.

A message is registered and displayed with information on error, nature, script name, and number line.

example:

```
15/03/2007 17:49:00
+c:\program files\easyphp1-8\www\formulaire\sample_more_forms.php' => line : 125
+ function: affichpage ( )
+++'c:\program files\easyphp1-8\www\inc\DefDaemonForm_V4.php' => line : 47
++++'c:\program files\easyphp1-8\www\Formulaire\Page_1.php' => line : 51
+++ function: texthtml ( "Nom" )
+++++'c:\program files\easyphp1-8\www\inc\DefForm_V4.php' => line : 455
++++ function: htmlscriptjava ( "Obj_Text"
++++   object 'obj_text' [
++++     "bGetObject" => True
++++     "ArrayChamp" => array (
++++       "name" => "Nom"
++++       "value" => ""
++++       "valueinit" => ""
++++       "status" => True
++++       "nature" => "input"
++++       "type" => "text"
++++       "size" => array (
++++         0 => 40
++++         1 => 2 )
++++       "format" => "s"
++++       "style" => "size =50"
++++       "list" => False
++++       "javascript" => array (
++++         "onKeyDown" => "CheckUpSizeString"
++++         "onKeyUp" => "CheckUpSizeString" )
++++       "namemsg" => "Nom_Msg_" )
++++     "TypObj" => "text" ]
++++   ,
++++   object 'obj_msg' [
++++     "bGetObject" => True
++++     "ArrayChamp" => array (
++++       "name" => "Nom_Msg_"
++++       "value" => "maxi 40c min 2c"
++++       "valueinit" => "maxi 40c min 2c"
++++       "status" => True
++++       "nature" => "input"
++++       "type" => "hidden"
++++       "format" => "s"
++++       "disable" => True
++++       "readonly" => True )
++++     "TypObj" => "msg" ]
++++   )
+++++'c:\program files\easyphp1-8\www\inc\DefForm_V4.php' => line : 571
+++++ function: htmlscriptjava ( )
+++++ Error: 2 'Division by zero' on script
+++++ 'c:\program files\easyphp1-8\www\inc\DefForm_V4.php' line :571
```

# Forms multipages

## 4 - THE CLASS OVERVIEW

### 4 - 1 CLASS 'OBJ\_DAEMONFORM'

OBJ\_DAEMONFORM is the scheduler of system.

#### USAGE

The OBJ\_DAEMONFORM object provides member functions for working with forms

#### SYNOPTIC

FUNCTIONS	USAGE
EndForm ()	Close form cf Syntax for script page (page 4)
FormsPage(\$Page = -1)	return reference OBJ_FORM for page \$Page If \$Page == -1 return reference array(OBJ_FORM) for all pages
GetCurrentPage()	return number current page
_GetDaemon()	return object Obj_DaemonForm
GetTitlePage(\$Page)	return title page for \$Page
GetTypePage(\$Page)	return type for page \$Page ('on','off','opt')
ResultForms	return result forms (body and array files to send as a array ordered map
SetCurrentPage(\$Page)	Set current Page with page \$Page
SetTypePage(\$Page, \$Type)	Set type for page \$Page
StartForm (\$Page, \$EncType)	Start form cf Syntax for script page (page 5)

### 4 - 1 - 1 'OBJ\_DAEMONFORM' FUNCTIONS

#### 4-1-1-1 ENDFORMS

CLOSE FORM (CF SYNTAX FOR SCRIPT PAGE (PAGE 5)

EXAMPLE :

```
_GETDAEMON () ->ENDFORM ()
```

#### 4-1-1-2 FORMSPAGE

RETURN REFERENCE OF OBJ\_FORM FOR PAGE \$PAGE

IF \$PAGE == -1, RETURN ARRAY OF OBJ\_FORM FOR ALL PAGES

EXAMPLE :

```
GET OBJ_FORM FOR THE SECOND PAGE
```

```
$OBJFORMS = _GETDAEMON () ->FORMSPAGE (2)
```

#### 4-1-1-3 GETCURRENTPAGE

RETURN NUMBER CURRENT

EXAMPLE :

```
$CURRENTPAGE = _GETDAEMON () ->GETCURRENTPAGE () : ;
```

## Forms multipages

### 4-1-1-4 `_GETDAEMON`

RETURN OBJ\_DAEMONFORM. IF NOT EXISTS, IT IS CREATED

### 4-1-1-5 `GETTITLEPAGE`

RETURN TITLE FOR PAGE \$PAGE (CF SYNTAX FOR SCRIPT PAGE (PAGE 5))

EXAMPLE:

```
$TITLE = _GETDAEMON() ->GETTITLEPAGE($PAGE) ;
```

### 4-1-1-6 `GETTYPEPAGE`

RETURN TYPE FOR PAGE \$PAGE.

```
$TYPE IS EITHER 'OPT' FOR OPTIONAL  
                'ON' FOR ACTIVE  
                'OFF' NOT ACTIVE
```

EXAMPLE:

```
$TYPE = _GETDAEMON() ->GETTYPEPAGE($PAGE) ;
```

### 4-1-1-7 `RESULTFORMS`

RETURN ARRAY RESULT FORM

```
ARRAY['BODY'] => STRING WHICH CONTAINS ALL FIELDS WITH THEIR NAMES AND VALUES  
['FILE'][FIELD-NAME] => FILENAME ULOADED  
['FILE'][TYPE]      => TYPE FILE
```

EXAMPLE:

```
$RESULTFORM = _GETDAEMON() ->RESULTFORMS() ;
```

### 4-1-1-8 `SETCURRENTPAGE`

SET CURRENT PAGE WITH \$PAGES

EXAMPLE:

```
_GETDAEMON() ->SETCURRENTPAGE($PAGE) ;
```

### 4-1-1-9 `SETTYPEPAGE`

SET TYPE PAGE \$PAGES WITH \$TYPE (CF `GETTYPEPAGE`)

EXAMPLE:

```
_GETDAEMON() ->SETTYPEPAGE($PAGE, $TYPE) ;
```

### 4-1-1-10 `STARTFORM`

START FORM FOR PAGE \$PAGES. THE SECOND PARAMETER IS OPTIONAL  
(CF SYNTAX FOR SCRIPT PAGE (PAGE 5))

EXAMPLE:

```
_GETDAEMON() ->STARTFORM($PAGE, "MULTIPART/FORM-DATA") ;
```

## Forms multipages

### 4 - 2 CLASS 'OBJ\_FORM'

#### USAGE

The OBJ\_FORM object provides member functions for working with fields form

**If error is encountered, all functions return NULL**

<b>SYNOPTIC</b>	
<b>FUNCTIONS</b>	<b>USAGE</b>
AlreadyInitialized()	return TRUE if form is already initialised
GetStatus()	return status form (cf SetStatus)
SetDebug(TRUE/FALSE)	Set mode debug for form page
SetStatus()	Set status form _FORM_VIRGIN_ => form virgin _FORM_VALID_ => form valid on get _FORM_ERROR_ => form error on get _FORM_RAZ_ => erase form

### 4 - 2 - 1 FORM

#### 4 - 2 - 1 - 1 FORM FUNCTIONS

##### 4-2-1-1-1 ALREADYINITIALIZED

###### DESCRIPTION

This function check if form already initialised.

```
$ObjForm->AlreadyInitialized();
```

###### RETURN VALUE

This function return TRUE if form is already initialised. Otherwise return FALSE.

###### EXAMPLES

```
if ($ObjForm->AlreadyInitialized() === TRUE)  
{ //This forms is initialised;
```

##### 4-2-1-1-2 GETSTATUS

###### DESCRIPTION

This function return the form status.

```
$ObjForm->GetStatus();
```

###### REMARKS

These values may be:

- \_FORM\_VIRGIN\_ => form virgin
- \_FORM\_VALID\_ => form valid on get
- \_FORM\_ERROR\_ => form error on get
- \_FORM\_RAZ\_ => erase form

### 4-2-1-1-3 SETDEBUG

#### DESCRIPTION

This function set or erase the mode debugging.

```
$ObjForm->SetDebug($bMode);
```

#### EXAMPLES

Set mode debugging: **\$ObjForm->SetDebug(TRUE);**

Erase mode debugging: **\$ObjForm->SetDebug(FALSE);**

### 4-2-1-1-4 SETSTATUS

#### DESCRIPTION

This function set the form status.

```
$ObjForm->SetDebug($Status);
```

#### REMARKS

These values may be:

- **\_FORM\_VIRGIN\_** => form virgin
- **\_FORM\_VALID\_** => form valid on get
- **\_FORM\_ERROR\_** => form error on get
- **\_FORM\_RAZ\_** => erase form

#### EXAMPLES

Set Status to **\_FORM\_ERROR\_**

```
$ObjForm->SetStatus(_FORM_ERROR_);
```

## Forms multipages

### 4 - 2 - 2 FIELDS

### 4 - 2 - 2 - 1 ATTRIBUTES FIELD

#### SYNOPTIC ATTRIBUTES

*on => available, off => not available | value between [...] are the default value | system => used by system*

ATTRIBUTES	BUTTON	CHECKBOX	RADIO	SELECT	TEXT
nature	[input] button	off	off	off	[input] textarea
type	['button'] 'submit' 'reset' 'upload_image'*1 'upload_file'*1	off	off	off	['Text'] 'password' 'hidden' 'file'
value	label_button image	list pre_checked	list pre_checked	list pre_checked	text_init
func_user	on*2	on*2	on*2	on*2	on*2
func_user_err	on*3	on*3	on*3	on*3	on*3
size	off	off	off	count select_display	maxi/mini
format	off	off	off	off	['s'] 'e' 'n####.##' 'aaaa-mm-jj' 'jpg jpeg gif'*4
style	on	on	on	on	on
accesskey	on	off	off	off	off
list	off	on	on	on	system*5
multiple	off	[TRUE]/FALSE	off	[TRUE]/FALSE	off
optional	[FALSE]/TRUE	[FALSE]/TRUE	[FALSE]/TRUE	[FALSE]/TRUE	[FALSE]/TRUE
disable	[FALSE]/TRUE	[FALSE]/TRUE	[FALSE]/TRUE	[FALSE]/TRUE	[FALSE]/TRUE
readonly	[FALSE]/TRUE	[FALSE]/TRUE	[FALSE]/TRUE	[FALSE]/TRUE	[FALSE]/TRUE
id	on	on	on	on	on
status	on	on	on	on	on
javascript	on	on	on	on	on
link	on*6	system	system	system	system *7
namemsg	system	system	system	system	system
valueinit	system	system	system	system	system

\*1 => if type is 'upload\_image' image uploaded is displayed inside button. Attribute 'value' must content  
<img src ='MyImage', ...> for default image with height and width and border for image to upload.  
=> if type is 'upload\_file' or 'upload-image':

Ctrl + Click display inside new window the file or the image uploaded

\*2 => Name callback user function called after button submit for personalized control  
if field is other that button: parameters are (attribute 'value' of field, array all fields of page)  
if field is type 'button': parameters are (name field button, array all fields of page)

\*3 => Name user function called for display error, if no field link error specify

\*4 => if type text is 'file', contents the formats for files allowed

\*5 => if type text is 'file', contents the array for file to upload

otherwise and format is 'n####', contents the rules for numeric value

\*6 => if type is 'upload\_image' or 'upload\_file', contents name of field text type 'file'

\*7 => if type text is 'file', contents field name of hidden field which contents the source full file name to upload.

=> if this a hidden field for a field text type 'file' to upload, contents the field name button type 'upload\_image'

## Forms multipages

### 4 - 2 - 2 - 2 FUNCTIONS

#### USAGE

#### SYNOPTIC

FUNCTION	BUTTON	CHECKBOX	RADIO	SELECT	TEXT	COMMENT
Add_Button(...)	on	off	off	off	off	Create object Obj_Button
Add_CheckBox(...)	off	on	off	off	off	Create object Obj_CheckBox
Add_Radio(...)	off	off	on	off	off	Create object Obj_Radio
Add_Select(...)	off	off	off	on	off	Create object Obj_Select
Add_Text(...)	off	off	off	off	on	Create object Obj_Text
GetIfClicked(...)	on	off	off	off	off	Return TRUE if button clicked
GetFieldStatus	on	on	on	on	on	Return status field
GetFieldValue(...)	off	on	on	on	on	Return field value
GetFieldValueMsg(...)	on	on	on	on	on	Return field link error value
GetList(...)	off	on	on	on	off	Return list choice
GetMagic(...)	on	on	on	on	on	Return attribute field
SetFieldStatus						Set status field
SetFieldValue(...)	off	on	on	on	on	Set new field value
SetFieldValueMsg(...)	on	on	on	on	on	Set new field link error value
SetList(...)	off	on	on	on	off	Set new list choice
SetMagic(...)	on	on	on	on	on	Set attribute field
TextHTML(...)	on	on	on	on	on	Build field code HTML
TextHTMLMsg(...)	on	on	on	on	on	Build field link error code HTML

## Forms multipages

For each field, it is possible to use function JavaScript.

example:

```
$ArrayJavaScript =  
    array ( "onKeyDown" => "MyFunction_1",  
           "onBlur"    => "MyFunction_2",  
           "onKeyUp"   => "MyFunction_3");
```

For each field, it is possible to put a attribute 'style'.

example:

```
$MyFieldObject("style" => "class=\"Class Name\" size=\"##\" col=\"##\" " etc..",  
"javascript"=>$ArrayJavaScript);
```

### 4-2-2-2-1 CONSTRUCT OBJECT *OBJ\_BUTTON*

#### DESCRIPTION

Build object for field button

**GetDaemon()->\$ObjForm(PAGE\_):**

**\$ObjForm->Add\_Button(\$NameFieldButton, \$Attributes, \$FieldLinkError = FALSE);**

#### RETURN VALUE

This function return **TRUE** if no error, else return **NULL**

The error messages

- name already exist
- name not present
- value attribute 'nature' incorrect
- value attribute 'type' incorrect
- attribute not available

#### EXAMPLES

Build button field '**Continue**' without field link error and with callback user function "**myControl**".

```
$ObjForm->Add_Button("Continue",array("type"=>"submit", "func_user" =>"myControl"));
```

**Note:** function **myControl** must be in script php and the name of this script must defined with parameter *P\_FuncUserControl*

Build button field '**erase**' without field link error.

```
$ObjForm->Add_Button("erase",array("type"=>"reset"));
```

Build button field '**Return**' with function javascript and with label 'Previous Page', without field link error.

```
$ObjForm->Add_Button("Return",array("value"=> "Previous Page",  
"javascript"=> array("onClick" => "GoPrevious")));
```

Build button field '**Graphic**' without field link error, for display a image in file '*FileImage*'

```
$Imag = "<img src='FileImage' width='50' height='50' border='0' alt='Return'>";
```

```
$ObjForm->Add_Button("Graphic",array("value" => $Imag));
```

Build button field '**Graphic**' for display a field text ('type' => 'file') named '*FileImageUpload*' (**file image upload**)

```
$Imag = "<img src='DefaultImage.jpg' width='50' height='50' border='0'>";
```

```
$ObjForm->Add_Button("Graphic",array("type" => "upload_image", "value" => $Imag,  
"link"=>"FileImageUpload"));
```

**Note:** The field named '*FileImageUpload*' must be created before the field button, otherwise, generate error!

## Forms multipages

### 4-2-2-2-2 CONSTRUCT OBJECT *OBJ\_CHECKBOX*

#### DESCRIPTION

Build object for field checkbox

**GetDaemon()->\$ObjForm(PAGE\_):**

Build object for field checkbox

**\$ObjForm->Add\_Radio(\$NameFieldCheckbox, \$Attributes, \$FieldLinkError = FALSE);**

#### RETURN VALUE

This function return TRUE if no error, else return NULL

The error messages

- name already exist
- name not present
- value attribute 'multiple' incorrect for multiple choices
- attribute 'list' not present
- attribute 'list' is not array
- attribute 'list' contains element which is not Obj\_Choice
- attribute not available

#### EXAMPLES

Build Checkbox Field named 'MyCheckBox', not optional, with field link error.

**\$ObjForm->Add\_CheckBox("MyCheckBox",array("list"=>\$Choice), array());**

#### LIST CHOICE FOR OBJECT RADIO, CHECKBOX

**For each field type Obj\_Radio or Obj\_CheckBox, a object Obj\_Choice must be created before Add\_ChecBox() or Add\_Radio().**

Parameters:

- 1=> Value button radio or checkbox
- 2=> TRUE if pre-checked
- 3=> Label button radio or checkbox  
If not present, this label must be by user on the script
- 4=> if Label present must be 'left' or 'right' for align label

example:

```
$Choice = array( new Obj_Choice("Mr",TRUE,"Mr","left"),  
                new Obj_Choice("Mme",FALSE,"Mme","left"),  
                new Obj_Choice("Mlle",FALSE,"Mlle","left") );
```

```
$Choice = array( new Obj_Choice("M",TRUE),  
                new Obj_Choice("F") );
```

## Forms multipages

### 4-2-2-2-3 CONSTRUCT OBJECT *OBJ\_RADIO*

#### DESCRIPTION

Build object for field radio

**GetDaemon()->\$ObjForm(PAGE\_):**

**\$ObjForm->Add\_Radio(\$NameFieldRadioName, \$Attributes, \$FieldLinkError = FALSE);**

#### RETURN VALUE

This function return **TRUE** if no error, else return **NULL**

The error messages

- name already exist
- name not present
- more options choice pre-checked
- attribute 'list' not present
- attribute 'list' is not array
- attribute 'list' contains element which is not *Obj\_Choice*
- attribute not available

#### EXAMPLES

Build Radio Field named '**MyRadio**', not optional, without field link error.

**\$ObjForm->Add\_Radio("MyRadio",array("list"=>**\$Choice**, "func\_user\_err"=>"MyFuncError"));**

**Note:** function **MyFuncError** must be in script php and the name of this script must defined with parameter *P\_FuncUserControl*

**cf CONSTRUCT OBJECT *OBJ\_CHECKBOX* for LIST CHOICE FOR OBJECT RADIO, CHECKBOX**

## Forms multipages

### 4-2-2-2-4 CONSTRUCT OBJECT *OBJ\_SELECT*

#### DESCRIPTION

Build object for field select

**GetDaemon()->\$ObjForm(PAGE\_):**

**\$ObjForm->Add\_Select(\$NameFieldSelect, \$Attributes, \$FieldLinkError = FALSE);**

#### RETURN VALUE

This function return **TRUE** if no error, else return **NULL**

The error messages

- name already exist
- name not present
- value attribute 'multiple' incorrect for multiple select
- attribute 'list' not present
- value attribute 'size' incorrect for view select
- attribute 'list' is not array
- attribute 'list' contains element which is not *Obj\_List*
- attribute not available

#### EXAMPLES

Build Select Field named '**MySelect**', not optional, with 2 options select, with field link error.

**\$ObjForm->Add\_Select("MySelect",array("list"=>\$List, "size"=> 2), array());**

#### LIST SELECT FOR OBJECT SELECT

**For each field type *Obj\_Select*, a object *Obj\_List* must be created before *Add\_Select()*.**

Parameters:

1=> \$OptGroupLabel    label OptGroup for option  
                          must be FALSE if end option

2=> \$OptGroupStyle    design view OptGroup

3=> \$Label;            label option

4=> \$PreSelected;    TRUE if pre-checked

5=> \$Value;            value option, if "", content \$Label

2=> \$Style;            design view:

example:

```
$List = array( new Obj_List("", "", "Select_1"),  
              new Obj_List("", "", "Select_2"),  
              new Obj_List("", "", "Select_3", TRUE) );
```

```
$List = array( new Obj_List("First List", "", "Select_1"),  
              new Obj_List("", "", "Select_2"),  
              new Obj_List("Second List", "", "Select_3", TRUE),  
              new Obj_List("", "", "Select_4") );
```

The tag 'size' means number of option to view.

If this number is < all options, a window unrolling is used.

if not present: All option are viewed.

## Forms multipages

### 4-2-2-2-5 CONSTRUCT OBJECT *OBJ\_TEXT*

#### DESCRIPTION

Build object for field textt

**GetDaemon()->\$ObjForm(PAGE\_):**

**\$ObjForm->Add\_Text(\$NameFieldText, \$Attributes, \$FieldLinkError = FALSE);**

#### REMARKS

Value available for attribute '*format*'

's' for string value

'e' for Email address

'n' for numeric value

"n" => number

"n####" => integer with 4 digits

"n####-" => number signed with 4 digits

"n####.##-" => number signed with 4 integers and 2 decimals

"n0.##-" => number signed decimal

"n.-" => number signed with no limit

'aaaa-mm-jj' for date ('-' or '/' or ':' as separator)

'mm-aaaa-jj' for date ""

'jj-mm-aaaa' for date ""

'jpeg|gif|jpg' , 'pdf|doc|etc...' => list extension allowed for file uploaded

#### RETURN VALUE

This function return **TRUE** if no error, else return **NULL**.

The error messages

- name already exist
- name not present
- value attribute 'type' incorrect
- value attribute 'nature' incorrect
- value attribute 'size' incorrect
- value attribute 'format' incorrect
- attribute not available

#### EXAMPLES

Build Text Field named '**Name**', not optional, number chars allowed between 40 and 2, without field link error.

```
$ObjForm->Add_Text("Name",array("size"=> array(40,2), "func_user_err"=>"MyFuncError"));
```

**Note:** function **MyFuncError** must be in script php and the name of this script must defined with parameter *P\_FuncUserControl*

Build Text Field named '**FirstName**', optional, number chars allowed between 20 and 2, with field link error initialised with "**max = 20c, min = 2c**".

```
$ObjForm->Add_Text("FirstName",array("size"=> array(20,2), "optional"=>true),  
array("value"=>"max = 20c, min = 2c"));
```

Build Text Field named '**Text**', not optional, number chars allowed between 1000 and 5, on 50 cols and 20 rows, with message and with field link error initialised with "**Please be concise**".

```
$ObjForm->Add_Text("Text",array( "style" => "cols=\`50\` rows=\`10\` wrap=\`virtual\`", "nature"  
=> "textarea", "size"=> array(1000,5)), array("value" => "Please be concise"));
```

## Forms multipages

Build numeric Field named '**Number**', not optional, 4 digits and 2 decimals, between 9999 and -2000 with field link error.

```
$ObjForm->Add_Text("Number", array("format"=>"n####,##-", "size"=>array(9999,-2000), array());
```

Build Field named '**FileImageUpload**', for upload file image with field link error, with max size file to 500ko. If size attribute is not present, default size (php.ini) is the max size for upload

```
$ObjForm->Add_Text("FileImageUpload", array("type" => "file", "format"=>"jpg|jpeg|gif",  
"size"=> 500000), array());
```

**cf (Obj\_Button for display File Image)**

Build Email Field named '**Email**', not optional, with field link error and. function user for control

```
$ObjForm->Add_Text("Email",array("format"=>"e", "func_user" =>"myControl"), array());
```

**Note:** function **myControl** must be in script php and the name of this script must define with parameter *P\_FuncUserControl*

## Forms multipages

### 4-2-2-2-6 GETIFCLICKED

#### DESCRIPTION

return **TRUE** if field button named **\$Name** is clicked  
**\$ObjForm->GetIfClicked(\$Name);**

#### RETURN VALUE

This function return **TRUE** or return **FALSE**,

#### EXAMPLES

Test if button named **\$Name** is clicked  
**\$bClick = \$ObjForm->GetIfClicked(\$Name);**

### 4-2-2-2-7 GETFIELDSTATUS

#### DESCRIPTION

**\$ObjForm->GetFieldStatus(\$Name);**

#### RETURN VALUE

This function return the value of attribute 'status' for a field object named **\$Name**. If error encountered, return **NULL**. This value can be either **TRUE** either a string that contains an error message

#### EXAMPLES

Get Status of field named **\$Name**  
**\$Status = \$ObjForm->GetFieldStatus(\$Name);**

### 4-2-2-2-8 GETFIELDVALUE

#### DESCRIPTION

**\$ObjForm->GetFieldValue(\$Name);**

#### RETURN VALUE

This function return the value of attribute 'value' for a field object named **\$Name**. If error encountered, return **NULL**

#### EXAMPLES

Get Value of field named **\$Name**  
**\$Value = \$ObjForm->GetFieldValue(\$Name);**

## Forms multipages

### 4-2-2-2-9 GETFIELDVALUEMSG

#### DESCRIPTION

`$ObjForm->GetFieldValueMsg($Name);`

#### RETURN VALUE

This function return the value of attribute 'value' for a field link error object named **\$Name**. If error encountered, return NULL

#### EXAMPLES

Get Value of field named **\$Name**:

`$ValueMsg = $ObjForm->GetFieldValueMsg($Name);`

### 4-2-2-2-10 GETLIST

#### DESCRIPTION

This function return the list choices (attribute 'list') for objects Obj\_Radio, Obj\_CheckBox and Obj\_Select named **\$Name**.

`$ObjForm->GetList($Name);`

#### RETURN VALUE

This function return the list choices (attribute 'list') for objects Obj\_Radio, Obj\_CheckBox and Obj\_Select. named **\$Name**. If error encountered, return NULL

#### EXAMPLES

Get list choices of field named "MySelect":

`$ListChoices = $ObjForm->GetList("MySelect");`

### 4-2-2-2-11 GETMAGIC

#### DESCRIPTION

This function return the value of attribute \$Attribute for a field named **\$Name**.

`$ObjForm->GetMagic($Name, $Attribut);`

#### RETURN VALUE

This function return the value of attribute \$Attribute for a field named **\$Name**.  
If error encountered, return NULL.

#### EXAMPLES

Get value of attribute 'size' for object named "**MyName**"

`$Size = $ObjForm->GetMagic("MyName", "size");`

## Forms multipages

### 4-2-2-2-12 SETFIELDSTATUS

#### DESCRIPTION

Set the status of attribute 'status' for a field object named **\$Name** with **\$NewStatus**.  
**\$NewValueStatus** can be either **TRUE** either a string which contains a error message  
**\$ObjForm->SetFieldStatus(\$Name, \$NewStatus = TRUE);**

#### RETURN VALUE

This function return **TRUE** or return **NULL** if field object named **\$Name** not exists

#### EXAMPLES

Set Status of field named "MyName" with "error on your answer"  
**\$bRes = \$ObjForm->SetFieldStatus("MyName", "error on your answer");**

Set Status of field named "MyName" with **TRUE**  
**\$bRes = \$ObjForm->SetFieldStatus("MyName");**

### 4-2-2-2-13 SETFIELDVALUE

#### DESCRIPTION

Set the status of attribute 'value' for a field object named **\$Name** with **\$NewValue**.  
**\$ObjForm->SetFieldValue(\$Name, \$NewValue);**

#### RETURN VALUE

This function return **TRUE** or return **NULL** if field object **\$Name** not exists

#### EXAMPLES

Set Value of field named "MyName" with **10-30-2007**  
**\$bRes = \$ObjForm->SetFieldValue("MyName", 10-30-2007);**

### 4-2-2-2-14 SETFIELDVALUEMSG

#### DESCRIPTION

Set the status of attribute 'value' for a field object link error object named **\$NameMsg** with  
**\$NewValueMsg**.  
**\$ObjForm->SetFieldValueMsg(\$NameMsg, \$NewValueMsg);**

#### RETURN VALUE

This function return **TRUE** or return **NULL** if field object **\$NameMsg** not exists

#### EXAMPLES

Set Value of field object link error named "MyNameMsg" with "**please your firstname**"  
**\$bRes = \$ObjForm->SetFieldValue("MyNameMsg", "please your firstname");**

## Forms multipages

### 4-2-2-2-15 SETLIST

#### DESCRIPTION

Set the list choices (attribute 'list') for object Obj\_Radio, Obj\_CheckBox and Obj\_Select. named \$Name.  
**\$ObjForm->SetList(\$Name, \$NewListChoices);**

#### RETURN VALUE

This function return **TRUE** or return **NULL** if error,

#### EXAMPLES

Set the list choices with **\$NewListChoices** for object named "MySelect"  
**\$bOk = \$ObjForm->SetList("MySelect", \$NewListChoice);**

### 4-2-2-2-16 SETMAGIC

#### DESCRIPTION

Set the value of attribute **\$Attribute** for a field named \$Name.  
**If attribute \$Attribute not exists and \$bForce is to TRUE, the attribute \$Attribute is created.**  
**\$ObjForm->SetMagic(\$Name, \$Attributes, \$NewValue, \$bForce = FALSE);**

#### RETURN VALUE

This function return **TRUE** or return **NULL** if error.

#### EXAMPLES

Set value of attribute 'size' for object named "MyName", with max length to 20 and min length to 2:  
if attribute 'size' no exists, it is created  
**\$bOk = \$ObjForm->SetMagic("MyName", "size", array(20,2), TRUE);**

### 4-2-2-2-17 TEXTHTML

#### DESCRIPTION

This function build code HTML for a field named \$Name.  
**\$ObjForm->TextHTML(\$Name);**

#### RETURN VALUE

This function return a string code HTML.  
If error encountered, return **NULL**

#### EXAMPLES

Get code HTML for a field named "MyName";  
**\$String = \$ObjForm->TextHTML("MyName");**

### 4-2-2-2-18 TEXTHTMLMSG

#### DESCRIPTION

This function build code HTML for a field link error named \$NameMsg.  
**\$ObjForm->TextHTMLMsg(\$NameMsg);**

#### RETURN VALUE

This function return a string code HTML.  
If error encountered, return **NULL**

#### EXAMPLES

Get code HTML for a field link error named "MyNameMsg":  
**\$String = \$ObjForm->TextHTMLMsg("MyNameMsg");**