

# Class AdminForm

---

Für pragmaMx

Version 1.0

Vom 28.03.2012

## Inhalt

Einleitung:.....	2
Was macht adminForm .....	2
Konfiguration :.....	3
Kernfunktionen:.....	4
Functionen für den Direktzugriff:.....	8
Beispiel für ein Formular:.....	9

## Einleitung:

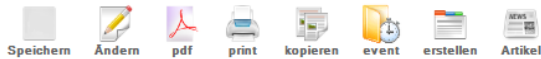
Klasse zur Vereinfachung der Formulargestaltung im Code

Toolbars:

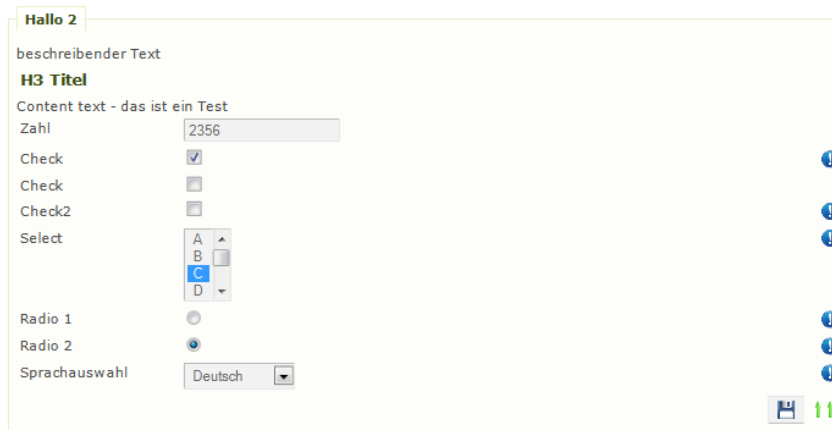
Ohne Beschreibung



Und mit Beschreibung



Komplette Eingabemasken



The screenshot shows a form titled "Hallo 2" with the following elements:

- Label: "beschreibender Text"
- Section: "H3 Titel"
- Text: "Content text - das ist ein Test"
- Field: "Zahl" with value "2356"
- Field: "Check" with checked checkbox
- Field: "Check" with unchecked checkbox
- Field: "Check2" with unchecked checkbox
- Field: "Select" with options A, B, C (selected), D
- Field: "Radio 1" with unselected radio button
- Field: "Radio 2" with selected radio button
- Field: "Sprachauswahl" with dropdown menu showing "Deutsch"
- Buttons: Save, Undo, and Redo at the bottom right.

Inkl. Fileupload, SPAW-Einbindung

Fieldsets klappbar, Infotext als Button mit ToolTips oder als kompletter Text im Formular....

## Was macht adminForm

Adminform übernimmt die komplette Formatierung des Formulars und bereitet deren Funktionsweise auf. Aber der Reihe nach ☺.

Am Anfang des Formulars kann eine Toolbar definiert werden. Hier kommen Schaltflächen zum Einsatz, über welche sich das Formular mit entsprechenden Aktionen auslösen lässt. Nachfolgend kommt nun das eigentliche Formular.

Der Klasse werden (sofern gewünscht) *fieldsets* definiert. In diese Fieldsets können nun die Formularelemente angeordnet werden. Alle Formularelemente werden der Reihenfolge nach in die Fieldsets eingebaut. Der Aufbau erfolgt 3-spaltig (aber tabellenfrei).

Ganz links steht der Feldbezeichner

Als 2. Kommt das Eingabefeld

Rechts kommt der Infotext zur Eingabe (entweder als Text oder als Bild für die Anzeige der Erklärungen als Tooltip). Die Klasse übernimmt die komplette Zuordnung der Feldelemente zu den

Fieldsets, das Layout und die Funktionalität des Formulars. Es bindet alle benötigten JavaScripts ein und blendet auch alle benötigten Button ein.

Ausserdem kann die Klasse die Toolbar-Funktionen auswerten und an das Programm übergeben.

## Konfiguration :

Folgende Konfigurationsmöglichkeiten sind vorhanden:

Variable mit Standardwert	Beschreibung
<code>,tb_pic_heighth' =&gt; 30,</code>	höhe Bilder der Toolbar, standard 30px
<code>,tb_pic_text' =&gt; true,</code>	Anzeige Text unter den Toolbarimages true false
<code>'tb_pic_path' =&gt; "images /adminform",</code>	standardpfad zu den Bildern -> pfad zur pmxRoot
<code>'tb_direction' =&gt; "right",</code>	standartdausrichtung right/left
<code>,tb_text' =&gt; "",</code>	text seitlich der Toolbar
<code>,tb_show'=&gt;'top',</code>	anzeige der Toolaber top=oben, bottom=unten, both = oben und unten
<code>,formname'=&gt;'adminForm',</code>	Name des Formulars
<code>,checklistflag' =&gt; false,</code>	gesetzt wenn eine Checkliste erwartet wird
<code>,collapsibleflag' =&gt; false,</code>	gesetzt, wenn die Fieldsets klappbar sein sollen
<code>,buttontext'=&gt;false,</code>	Anzeige von Save und UP-Button unterhalb der Fieldlist
<code>,formopenflag'=&gt;FALSE,</code>	gesetzt, wenn form geöffnet wurde
<code>'target_url'=&gt;\$_SERVER['REQUEST_URI'],</code>	Target-Url des Formulars -> standard = selfPHP
<code>,enctype'=&gt;' text/plain',</code>	encrypt-Type für das Formular
<code>,title'=&gt; "",</code>	<h3>-Titel über dem Formular
<code>,description'=&gt; "",</code>	Text unter dem Titel
<code>,cssclass'=&gt;'toolbar',</code>	css-Klasse für die Toolbar ... standard ,toolbar'
<code>'infobutton'=&gt;false,</code>	Infobutton (true) oder Infotext (false)

<code>'infobuttonflag'=&gt;false,</code>	
<code>'tooltiptheme'=&gt;'light',</code>	Style der ToolTips plain, dark, light, blue, red, green
<code>'tooltipstyle'=&gt;'youtube',</code>	shadow, rounded, youtube, jtools, cluetip, tipped, tippy
<code>,checkselector'=&gt;'cid',</code>	Name der Checkbox, die für "alle selektieren" verantwortlich ist

## Kernfunktionen:

### ➤ FormOpen(\$action="", \$etype="")

Zugriff: public

Parameter:

Parameter	Format	Standardwert	Beschreibung
Action	String	Php_self	Url, an welche das Formular die Daten sendet
Etype	String	text/plain	Ebcrypt-String für das Formular, sobald Fileupload-Elemente verwendet werden, passt die klasse diesen Wert automatisch an

Rückgabewert: string

Beschreibung:

Diese Funktion gibt den kompletten HTML-Formularkopf zurück und wird dynamisch an die Verhältnisse des Formulars und der Konfiguration angepasst. Es bindet benötigte JavaScripts automatisch mit ein. Diese Funktion kann entweder direkt aufgerufen werden oder sie wird automatisch beim Aufruf der Funktion Show abgearbeitet. Beim direkten Aufruf über echo oder in eine Variable sollte daran gedacht werden, dass, wenn möglich vorher die einzelnen Formularfelder definiert wurden, oder die Konfiguration entsprechend angepasst wurde.

### ➤ \_\_set(\$name, \$val)

**Zugriff:** public

**Parameter:**

Parameter	Format	Standardwert	Beschreibung
Name	String		Variable für Konfiguration
Val	Mixed		Wert für Variable

**Beschreibung:** hiermit können die Konfigurationsdaten verändert werden

### ➤ \_\_get(\$value\_name)

**Zugriff:** public

**Parameter:**

Parameter	Format	Standardwert	Beschreibung
Name	String		Variable für Konfiguration
Val	Mixed		Wert für Variable

**Rückgabe:** Wert der variablen, oder false

**Beschreibung:** Auslesen der Konfigurationsdaten

### ➤ Show()

**Zugriff:** public

**Parameter:** keine

**Rückgabe:** string

**Beschreibung:** ist eigentlich die Kernfunktion, welche das komplette Formular zusammenstellt:

- Den Formulkopf
- Die Toolbar
- Die Fieldsets
- Alle Formularelement
- Alle benötigten Button und JavaScripts
- Den Formulfuss

Die Function kombiniert die einzelnen beschriebenen Funktionen und gibt den kompletten HTML-Code zurück.

➤ **addFieldSet(\$fname,\$ftitle,\$flegend="", \$collapsible=false)**

Zugriff: public

Parameter:

Parameter	Format	Standardwert	Beschreibung
<b>Fname</b>	String	-	Name des FieldSets
<b>Ftitle</b>	String	-	Bezeichnung (erscheint oben im Kopf des FieldSets)
<b>Flegend</b>	String	Leer	Beschreibung, erscheint am Anfang des Fieldsets.
<b>Collapsible</b>	Boolean	False	True = Fieldset ist einklappbar False = Fieldset ist fest

Rückgabe: keiner

**Beschreibung:** Wenn Fieldset verwendet werden sollen, dann **müssen** diese erst mit dieser Funktion definiert werden. Entscheidend ist der Parameter *fname*. Dieser wird auch bei der function *add* benötigt, um einem Formularelement das richtige Fieldset zuzuweisen.

**ACHTUNG : WERDEN KEINE FIELDSETS DEFINIERT, WERDEN ALLE HINZUGEFÜGTEN FORMULARELEMENTE UNTEREINANDER ANGEZEIGT.**

➤ **add (\$ffieldset,\$ftype,\$flegend,\$ffieldname="", \$fvalue="", \$fdesc="", \$ffieldlen=0, \$fextern="")**

Zugriff: public

Parameter:

Parameter	Format	Standardwert	Beschreibung
<b>Ffieldset</b>	String	-	Name des definierten FieldSets worin dieses Element erscheinen soll

## Class adminForm für pragmaMX

<b>Ftype</b>	String	Input	Eingabetype: input = Texteingabe output = allgemeine Textausgabe textarea= Textarea/SPAW yesno = ja/nein select = Auswahlliste checkbox= CheckBox password= Passworteingabe hidden= verstecktes feld file= Uploadfeld textarea= Textbox oder SPAW radio= Radiobuttons html = html-code allgemein
<b>Flegend</b>	String		Beschreibung des Feldes (erscheint in der linken Spalte)
<b>Ffieldname</b>	String		Feldname (eindeutig und ohne Sonderzeichen)
<b>value</b>	mixed		Wert des Eingabefeldes Bei SELECT = array( Anzige => wert )
<b>Fdesc</b>	String		Beschreibungstext (Info)
<b>Ffieldlen</b>	Intval	0	Ggf. Eingabelänge definieren
<b>Fextern</b>	String		Interne Verwendung

Rückgabe: keine

**Beschreibung:** ist die Funktion, welche die eigentlichen Formularelemente in das Formular „einbaut“.

Hier mal anhand einer Ausgabe kurz angezeigt:

```
<label class="forminputtitle" title="fdesc" for="Form1Test221">flegend</label>
<div class="forminputfield">
<input id="Form1Test221" type="ftype" title="fdesc" value="value" name="ffieldname">
</div>
<div class="forminputdesc">fdesc</div>
```

Der Parameter *ffieldset* ist hier nicht zu sehen, da er die Zuweisung für das evtl. FieldSet ist.

Die ausserdem hier gezeigten CSS-Klassen werden über die Adminform-Klasse beschrieben.

## Functionen für den Direktzugriff:

### ➤ FormClose()

**Zugriff:** public

**Parameter:** keine

**Rückgabe:** string

**Beschreibung:** gibt den kompletten HTML-Formularfuss zurück. Wie FormOpen wird auch hier der String dynamisch anhand der Formulargestaltung zusammengestellt.

### ➤ FieldSetOpen(\$fieldname,\$legend,\$extendedtext="", \$collapsible = false)

**Zugriff:** public

**Parameter:**

Parameter	Format	Standardwert	Beschreibung
<b>Fieldname</b>	String	NULL	Name des Fieldsets – wird intern verwendet, Umlaute und Sonderzeichen nicht verwenden !!
<b>Legend</b>	String	NULL	Beschreibung des Fieldsets. Die Beschreibung wird im Kopf des Fieldsets ausgegeben
<b>Collapsible</b>	Boolean	False	Bei true wird das Fieldset aufklappbar gemacht, bei false ist es immer komplett sichtbar.

**Rückgabe:** string

**Beschreibung:** gibt den HTML-Code für den Anfang des Fieldsets zurück. Muss paarweise immer mit der Funktion *FieldSetClose* verwendet werden

### ➤ FieldSetClose(\$button = true)

**Zugriff:** public

**Parameter:**

Parameter	Format	Standardwert	Beschreibung
-----------	--------	--------------	--------------



<b>Button</b>	Boolean	True	Gibt im „Fuss“ des Fieldsets einen „Save“-Button aus und einen Link zum FormAnfang
---------------	---------	------	--

**Rückgabe:** string

**Beschreibung:** gibt den HTML-Code für das Fieldset-Ende zurück. Muss immer Paarweise mit der function *FieldsetOpen* verwendet werden.

➤ **addFieldSet(\$fname,\$ftitle,\$flegend="", \$collapsible=false)**

**Zugriff:** public

**Parameter:**

Parameter	Format	Standardwert	Beschreibung

**Rückgabe:**

**Beschreibung:**

## Beispiel für ein Formular:

```
$tb=load_class("AdminForm");  
*/// optionale Angaben  
$tb->__set('infobutton', false);  
$tb->__set('description', "Beschreibungstext");  
$tb->__set('title', "Titel");  
$tb->__set('tb_pic_text', true); // 0/1 schaltet Bildunterschrift aus/ein
```

## Class adminForm für pragmaMX

```
(standard=ein)
$tb->__set('formname','Form1');











// ToolbarButton hinzu
$tb->addToolbar("save",_SAVE,"home",false);
$tb->addToolbar("edit");
$tb->addToolbarLink("pdf","/pmlx112rc1/modules.php?name=SiriusGallery");
$tb->addToolbar("print");
$tb->addToolbar("copy");
$tb->addToolbar("event");
$tb->addToolbar("categorys","erstellen");
$tb->addToolbarLink("news","modules.php?name=News");

$tb->addFieldSet("Test1","Hallo 2","beschreibender Text",false);
$tb->addFieldSet("Test2","Titel","beschreibender Text 22 ",true);
$tb->addFieldSet("Test3","Hallo 22222","beschreibender Text ",false);

$tb->add("Test1","output","H3 Titel","Content text - das ist ein Test");
$tb->add("Test2","output","Tooltip http://craigsworks.com/projects/simpletip/#");
$tb->add("Test3","output","H3 Titel 22","Content text - das ist ein Test");
$tb->add("Test1","", "Zahl", "zahl2", "2356");
$tb->add("Test2","input","Nummer","zahl3","2356","Hier nur Zahlen eingeben");
$tb->add("Test2","input","Zahl mit ganz langem
einleitungstext","zahl4","2356","beschreibung mit langem einleitungstext");
$tb->add("Test2","input","Zahl","zahl5","123456");
$tb->add("Test2","html","<hr>");
$tb->add("Test2","file","File","file1","pole","hallo das ist die File beschreibung
für den Upload",40);
$tb->add("Test2","yesno","JaNein","yesno",1,"beschreibender Text ");
$tb->add("Test3","checkbox","Check","wcheck[1]","1","beschreibung");
$tb->add("Test3","checkbox","Check","wcheck[2]","0");
$tb->add("Test1","checkbox","Check2","wcheck[3]","0","beschreibung");
$tb->add("Test2","usergroup","usergroup","usergroup","", "beschreibung");
$tb->add("Test1","select","Select","select2",3,"beschreibung",3,
    array('A'=>'1','B'=>'2','C'=>3,'D'=>4,'e'=>4,'f'=>4,'g'=>4,'h'=>44,));
$tb->add("Test2","textarea","SPAW","textbox","bla bla","beschreibung",80,true);
$tb->add("Test","radio","Radio 1","rcheck52","1","beschreibung");
$tb->add("Test","radio","Radio 2","rcheck52","1","beschreibung");
$tb->add("Test","selectlanguage","Sprachauswahl",
    "language","german","beschreibung");



echo $tb->getAdminForm();
```

Dieses erzeugt folgendes Ausgabeformular:





**Titel**  
Beschreibungstext  
         
 




**Hallo 2**  
beschreibender Text  
**H3 Titel**  
Content text - das ist ein Test  
Zahl   
Check2  beschreibung  
Select 

A	↑
B	□
C	□
D	↓

beschreibung  
Radio 1  beschreibung  
Radio 2  beschreibung  
Sprachauswahl  beschreibung  
 

**Titel**

**Hallo 22222**  
beschreibender Text  
**H3 Titel 22**  
Content text - das ist ein Test  
Check  beschreibung  
Check   
   
 

Wie ersichtlich ist das 2. FieldSet geschlossen dargestellt. Es kann aber über die Button oben und unten rechts   oder das  neben dem Titel eingeblendet werden. Das aufgeklappte Fieldset sieht dann so aus :

