

Weich wechselnde Bildershow

FADESHOW mit Symbolbefehl

für CMBasic

(Version 1.4 / 24.07.2018)

Noch offen:

- **Verlängerung der Anzeige des ersten Bildes.** Vermutlich Opfer der Bildershow-Ladeprozedur oder Programmfehler, da **erstes Bild sofort gedeckt** wird.
- **Fadeshow kann nicht in einer Tabelle platziert werden!** Bild ist vorhanden, aber der aut. Start funktioniert nicht. -> dies ist ein generelles JS-Problem bei CMBasic, da der onload-Befehl vom HTML-Editor herausgeputzt wird.
- Die Fadeshow hat **auf dem iPad eine viel kürzere Fadingphase.** Grund noch unbekannt! Muss mit Safari-Browser etwas zu tun haben.

Vorstudie

Idee der Bildershow:

- Mittels Symbolbefehl soll auf Mitgliederseiten eine endlose, automatisch wechselnde Bildershow eingesetzt werden können.
- Diese Bildershow soll ein grössenmässig einstellbares Anzeigefeld von ca. 300 .. 700 Pixel Breite besitzen und anlaufen sobald die Seite geöffnet wird.
- Das Format der Bilder sollte in etwa immer dasselbe sein (Breite : Höhe = ca. 1,5).
- Die Show soll bis zu 99 Bilder zeigen können und jedes Mal, wenn die Seite geöffnet wird, an der ersten Stelle beginnen.
- Der Übergang soll mit Fading erfolgen
- Es ist keine Bilderbezeichnung vorgesehen
- Die Showlänge (Anzahl Bilder) muss vom Programm erkannt werden
- Die max. 99 Bilder sind in einem vorhandene CMBasic-Verzeichnis (images1 bis 3) abgelegt.
- Die Bilder werden entsprechend der gewünschten Anzeigeposition unnummeriert und mit fs speziell bezeichnet (z.B. fs_1.jpg).
- Die Bilder sollen maximal 800 Pixel breit sein, damit sie normal hochgeladen werden können.
- Im Symbolbefehl sollen das verwendete Bilderverzeichnis, die verwendete Galeriebreite, die Anzeigedauer und die Wechseldauer als Attribut beigefügt sein.
- Das Hochladen der Bilder in das Bilderverzeichnis (z.B. images3) soll über das normale Bild-Upload-Werkzeug des Backends erfolgen.

Name und Definition des Symbolbefehls:

Aufbau ähnlich wie andere Symbolbefehle des CMB. Gewählte Symbolbefehlssyntax:

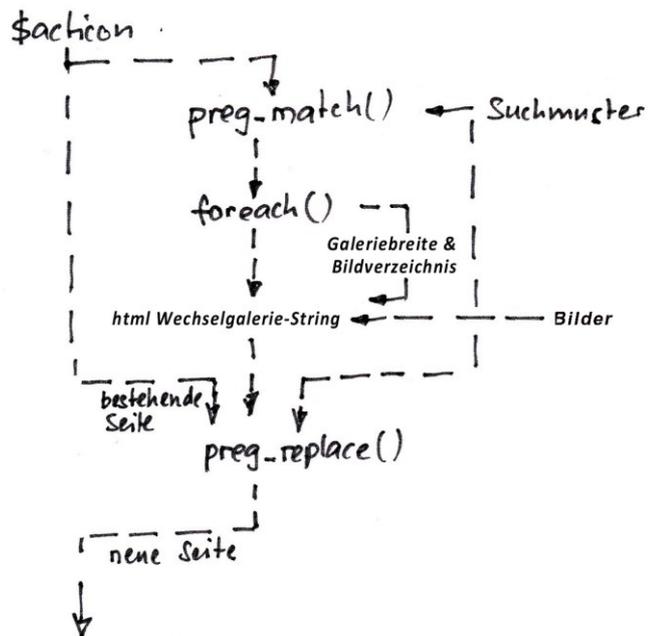
{FADESHOW,Breite,Verzeichnis,Anzeigedauer,Wechsel}

Informationsabläufe im CMBasic

In der Datei **contentshow.inc.php** wird eine zusätzliche Zeile zur Einbindung des Symbolbefehls eingefügt. Diese Zeile ist aber nur funktionsfähig, wenn vorher in der unter dem Verzeichnis „edit“ abgelegten Konfigurationsdatei **config.inc.php** der Symbolbefehl freigeschaltet worden ist.

Nun wird der erkannte Symbolbefehl zur Datei **bot_fadeshow.inc.php** weitergeleitet, wo er (wie im folgenden Bild gezeigt) in seine Attribute zerlegt und in einen html-String umgewandelt wird, der an Stelle des Symbolbefehls in den Seitencode (\$acticon) eingefügt wird. Dieser html-String enthält neben dem JavaScript die Galeriebreite, einen Bilderlink, die Anzeigedauer und die Wechseldauer.

Das Hochladen und Löschen der Bilddateien erfolgt wie üblich über das bereits bestehende **upload.inc.php**.



Erweiterung der Datei contentshow.inc.php

In Zeile 595 ff folgende Einbindung des Symbolbefehls einfügen:

```

if (defined('FADESHOW') && FADESHOW == TRUE)
    include_once 'minibot/bot_fadeshow.inc.php';
}

```

Erweiterung der Datei config.inc.php:

Neue Zeile mit Freigabemöglichkeit des Symbolbefehls einfügen:

```

define('FADESHOW', TRUE); // Fadeshow-Verwendung freigeben

```

Entwicklung des Moduls „bot_fadeshow.inc.php“:

Verwendete Variablen:

- \$acticon (Seitencode)
- \$breite (eingegebene Galeriebreite in Pixel)
- \$verz (eingegebenes Bilderverzeichnis bezogen auf Root?)
- \$dauer (eingegebene Anzeigedauer der Bilder in ms)
- \$galadd (Attributzusammenfassung für preg_replace am Schluss)
- \$filelist (Liste der Bilder in Array-Form)
- \$fjs (Fader-JavaScript)
- \$wsl (Faktor zur Wechseldauereinstellung)

Programmablauf:

- Suche nach Symbolbefehl im \$acticon-String mittels vereinfachtem Suchmuster
- Herauslesen der Attribute Breite, Verzeichnis, Anzeigedauer und Wechseldauer
- Feststellen der Anzahl Bilder der Bildershow
- Erstellen des Bilderlisten-Strings in Array-Form (ohne Endkomma!)
- Anpassen des JavaScripts, dass es die Länge der Galerie automatisch erkennt und die Bildanzeige entsprechend steuert
- Zusammenstellen des Fadeshow-html-Strings bestehend aus Bilderlisten-String, JavaScript, Galeriebreite, vollständiges Symbolbefehl-Suchmuster und einfügen desselben in den \$acticon-String an Stelle des Symbolbefehls

Script der Datei bot_fadeshow.inc.php:

```
<?php
defined('CMBASIC') or die();
/* Definieren der verwendeten Variablen */
$breite = "";
$verz = "";
$dauer = "";
$wsl = "";
$list = "";
/* Erkennen, Analyse des Symbolbefehls und Weitergabe der Attribute */
$muster_syntax = "{FADESHOW,Hoehe,Verzeichnis,Anzeigedauer,Wechsel}";
$suchmuster = "|{((FADESHOW),(.?*))}|si";
preg_match_all($suchmuster,$acticon,$galfund);
foreach($galfund[1] as $galpara_komp) {
    $galpara = explode(", ", $galpara_komp);
    if (!empty($galpara[1])) {
        $breite = $galpara[1]; // Bildbreite in px
        $galadd .= "$galpara[1]";
    }
    if (!empty($galpara[2])) {
        $verz = $galpara[2]; // CMBasic-Verzeichnis -> images1 ... images3
        $galadd .= ",$galpara[2]";
    }
    if (!empty($galpara[3])) {
        $dauer = $galpara[3]; // Anzeigedauer -> 500 bis 10'00 (ms?)
        $galadd .= ",$galpara[3]";
    }
    if (!empty($galpara[4])) {
        $wsl = $galpara[4]; // Wechselgeschwindigkeit -1(0.5Sek) - 50(3Sek)
        $galadd .= ",$galpara[4]";
    }
}

/* Überprüfen der PHP-Version */
if (phpversion() >= 5) { // Filter für PHP-Version / sonst Sprung zum Ende der PHP-Prüfschleife mit Textausgabe
```

```
/* Prüft ob Bilderverzeichnis existiert */
```

```
if (!file_exists("$verz")) { // Führt nur weiter, wenn Bilderordner existiert  
$galcode_0 = '<p><strong>Dieser Bilderordner existiert leider nicht!</strong></p>';  
} else {
```

```
/* Erstellt den Bilder-String $list aus den fs-Dateien des Bilderordners $verz */
```

```
$galcode_0 = ""; // setzt Galeriecode auf leer  
$list_0 = ""; // setzt Bilderliste auf leer  
$i = 1; // setzt Schleifenzähler auf 1  
while (is_file("$verz/fs_{$i}.jpg")) { // prüft ob Bild vorhanden ist -> benötigt ""!  
if ($i == 1){  
$list = $list_0."$verz/fs_{$i}.jpg"; // setzt erstes Bild ohne vorangestelltem Komma ein  
} else {  
$list = $list_0."$verz/fs_{$i}.jpg"; // setzt weitere Bilder mit vorangestelltem Komma ein  
}  
$i++; // erweitert Schlaufenzähler um 1  
}  
}
```

```
/* Das verwendete Fader-Script stammt vom Faderkurs von Felix Riesterer. Es wird direkt unter der Variablen $fjs abgelegt, da eine JS-Auslagerung aus einem html-Script scheinbar Probleme bietet */
```

```
$fjs = "<script type='text/javascript'> // Beginn des $fjs-Scripts  
var FaderFramework = { // spezifiziert die Variablen (Methoden) für das Objekt FaderFramework  
oldWinOnLoad: false, // setzt die Methode „oldWinOnLoad“ auf null  
inits: new Array(), // erstellt mit Methode „inits“ ein neues Array  
faders: new Object(), // erstellt mit Methode „faders“ ein neues Objekt  
init: function (einstellungen) { // erstellt bei Methode „init“ eine Funktion „einstellungen“  
var fader; // eröffnet Variable „fader“  
if (this.inits) { // überprüft ob neuer Array vorliegt  
this.inits[this.inits.length] = einstellungen; // wenn Array vorhanden: Array-Grösse wird zu „einstellungen“  
} else { // sonst, d.h. wenn kein Array vorliegt ..  
fader = new this.Fader(einstellungen); // setzt Variable „fader“ auf Kopie von ..... ?  
if (fader != false && !this.faders[fader.id]) { // überprüft ob Variable „fader“ gesperrt oder nicht vorhanden ist  
this.faders[fader.id] = fader; // übernimmt Inhalt der Variablen „fader“  
window.setTimeout(function () { fader.next(); }, fader.viewTime); // setzt Timer auf ... ?  
} // Abschluss if (fader)-Schleife  
} // Abschluss else-Schleife  
}, // Abschluss init-Schleife  
  
start: function () { // Beginn der „start“-Funktion  
this.oldWinOnLoad = window.onload; // Start bei „on Load“  
window.onload = function () { // Definition der Ladefunktion  
if (typeof(FaderFramework.oldWinOnLoad) == 'function') { // Prüft ob .....?  
FaderFramework.oldWinOnLoad();  
} // Abschluss if-Schleife  
FaderFramework.onload(); // Aufruf der onload-Funktion  
}; // Abschluss windows onload-Funktion  
}, // Abschluss Funktions-Schleife  
  
onload: function () { // Beginn der „onload“-Funktion
```

```

var i, fader, e, scripts = document.getElementsByTagName('script'); //eröffnen der Variablen
for (i = 0; i < scripts.length; i++) { // setzen einer for-Schleife für .....?
    if (scripts[i].src && scripts[i].src.match(/fader-framework\.js/)) {
        this.baseURL = scripts[i].src.replace(/(^\|\\)fader-framework\.js$/, "");
    }
}
if (this.baseURL) {
    e = document.createElement('link');
    e.type = 'text/css';
    e.rel = 'stylesheet';
    e.href = this.baseURL + '/fader-framework.css';
    document.getElementsByTagName('head')[0].appendChild(e);
}
fader = this.inits;
delete this.inits;
for (i = 0; i < fader.length; i++) {
    this.init(fader[i]);
}
}, // Abschluss „onload“-Funktions-Schleife
Fader: function (einstellungen) { // Beginn der „einstellungen“-Funktion
    if (!einstellungen.id || !document.getElementById(einstellungen.id)
        || FaderFramework.faders[einstellungen.id]
        || einstellungen.images.length < 2) {
        return new Boolean(false);
    }
    var i, original = document.getElementById(einstellungen.id);
    this.id = einstellungen.id;
    this.images = new Array();
    this.counter = false;
    this.element = document.createElement('span');
    this.element.className = 'fader';
    original.parentNode.replaceChild(this.element, original);
    for (i = 0; i < einstellungen.images.length; i++) {
        this.images[i] = document.createElement('img');
        this.images[i].src = einstellungen.images[i];
        this.images[i].alt = 'Bild';
        if (i == 0) {
            this.element.appendChild(this.images[i]);
        }
    }
} // Abschluss der „einstellungen“-Funktion

/* Festlegung und Anzeige der Bilder */
this.fade = function (step) { Funktionsname
    var fader = this, imgs = this.element.getElementsByTagName('img'); // ermittelt die 2 Bilder und legt sie ab
    step = (1step) ? 0 : step; // ermittelt den effektiven Durchlässigkeitschritt
    imgs[1].style.opacity = step/100; // stellt auf oberem Bild 2 die entsprechende Durchlässigkeit ein
    imgs[1].style.filter = 'alpha(opacity=' + step + ')'; // dasselbe für ältere IE-Browser

/* Einstellen der Fadedauer */
    step = step + 2; // Erhöht die Durchlässigkeit
    if (step <= 100) { // vergleicht mit Grenzwert, wenn Bild noch im Aufbau gilt ...

```

```

        window.setTimeout(function () { fader.fade(step); }, $swsl ); // startet Funktion „fade“ nach $swsl ms erneut
    } else { // wenn Grenzwert erreicht ist geht's hier weiter ..
        imgs[1].className = ""; // Klassenname des oberen Bildes wird gelöscht
        this.element.removeChild(imgs[0]); // unteres Bild wird gelöscht

/* Einstellung der Anzeigedauer der Bilder mit $dauer */
        window.setTimeout(function () { fader.next(); }, $dauer); // startet Funktion „next“ nach $dauer ms
    }
}; // Abschluss der „fade“-Funktion
/* Laden des nächsten Bildes */
this.next = function () { // Methode „next“
    this.counter = (this.counter < this.images.length -1) ? this.counter +1 : 0; // erhöht Bildzählerstellung um eins
    this.element.appendChild(this.images[this.counter]); // holt nächstes Bild
    this.images[this.counter].className = 'next'; // gibt dem nächsten Bild die Klasse „next“
    this.fade(); // startet Funktion „fade“ zur Veränderung der Durchlässigkeit
};
} // Abschluss der „einstellen“-Funktion
}; // Abschluss der „FaderFramework“-Funktion

```

```

/* Öffnen eines neuen Fader-Objektes */
FaderFramework.start(); // startet Methode „start“ im übergeordnetes Objekt „Framework“
    FaderFramework.init({ // startet init-Methode zum Einrichten neuer Fader Objekte
        id: 'fader1', // setzt für ID „fader 1“ ein
        images: $list // setzt eine Bilderliste ein
    });
</script>; // Ende des $fjs-Scripts

```

```

/* Erstellen des provisorischen Galeriecodes */
$galcode_0= <<<GALCODE // öffnet Inhalt des Galcode
<style type="text/css"> // öffnet Style-Script
.fader {position: relative; display: inline-block;} // positioniert Fade-Aufgabe
.fader img {vertical-align: top; width: {$breite }px;} // positioniert erstes, unteres Bild
.fader img.next {position: absolute; top: 0; left: 0;} // positioniert zweites, oberes Bild
</style>
{$fjs} // fügt Fader-Script ein
<p></p> // fügt erstes Bild ein
GALCODE; // Abschluss des prov. Galcode

```

```

/* Umformen des provisorischen Galeriecodes zum Einfügen */
$galcode = "$galcode_0"; // html-Code zusätzlich mit Stringzeichen versehen

```

```

/* Abschluss der PHP-Prüfschleife */
} else {
$galcode_0 = '<p><strong>Die Fader-Show geht leider wegen veralteter PHP-Version nicht!</strong></p>';
}

```

```

/* Einfügen des fertigen Galeriecodes an Stelle des Symbolbefehls */
$muster2 = "|{FADESHOW, $galadd}|"; // Suchmuster für Symbolbefehl inkl. Attributauflistung
$acticon = preg_replace($muster2, $galcode, $acticon); // Ersetzen des Symbolbefehls (muster2) durch
// den Galeriecode (galcode)
?>

```

Beschreibung fürs Handbuch

Einbinden einer weich wechselnden Bildershow mit bis zu 99 Bildern

Wenn erwünscht, lässt sich pro Seite eine Fader-Bildershow mit bis zu 99 automatisch wechselnden Bildern gleichen Formats einfügen. Es ist keine Bilderbeschriftung vorhanden. Die Show wird linksbündig angezeigt, kann aber eingemittet werden.

Der dazu verwendete Symbolbefehl lautet:

{FADESHOW,Breite,Verzeichnis,Anzeigedauer,Wechsel}

Das erste fixe Attribut (FADESHOW) dieses Befehls zeigt dem System an, dass an diesem Platz eine Fadeshow erstellt werden soll. Mit dem zweiten Attribut (Breite) wird die Breite des Galeriefensters bestimmt, mit dem dritten Attribut (Verzeichnis) wird das Bilderverzeichnis angegeben, in welchem die einzelnen Bilder abgelegt sind, mit dem vierten Attribut (Anzeigedauer) wird die Anzeigedauer und mit dem fünften Attribut (Wechsel) wird die Übergangsgeschwindigkeit zwischen den Bildern eingestellt.

Als Breite ist die Pixelausdehnung des Anzeigefensters (maximal 500) einzusetzen. Eingabe erfolgt ohne Einheit px!

Unter Verzeichnis wird das Bilderverzeichnis (images1 bis images3) eingetragen, in welchem die Bilder zu finden sind.

Mit dem Attribut Anzeigedauer kann die Anzeigedauer der Bilder in Millisekunden (z.B. 2'000) festgelegt werden. Eingabe ohne Einheit ms!

Mit dem Attribut Wechsel kann die Übergangsgeschwindigkeit zwischen den Bildern beeinflusst werden. Eingabe einer Zahl von 1 bis 100, wobei die Ueberblendzeit proportional zur Grösse der Zahl steigt. Bei 1 dauert es 0.5 Sekunden und Bei 100 dauert es 50 Sekunden. Eingabe erfolgt ohne Einheit.

Die in der Fadeshow verwendeten Bilder müssen im Bilderverzeichnis unter einer normierten, laufenden Dateinamennummer abgelegt sein, damit sie von den restlichen Bildern unterschieden werden können. Die dabei verwendete Syntax sieht wie folgt aus:

fs_1	erstes Bild
fs_2	zweites Bild
fs_3	drittes Bild
.....
fs_98	achtundneunzigstes Bild
fs_99	letztes Bild

Bemerkung: Die Nummerierung der Bilder muss lückenlos sein. Bei fehlenden Bildern wechselt die Show zurück zum ersten Bild.

Die Bilder der Fadeshow werden nach der Umbenennung über das Register "Bilder" im Backend in das entsprechende Verzeichnis hochgeladen. Da dazu die Bilderverzeichnisse images1 bis images3 verwendbar sind, können parallel bis zu 3 unabhängige Fadeshows eingerichtet werden.

Das Auswechseln, Erweitern bzw. Löschen von Bildern erfolgt im Bilderverzeichnis, welches im Backend unter dem Register "Bilder" zu finden ist. Hier existiert hinter jedem Bild ein Knopf "löschen". Die Bilder müssen entsprechend der oben genannten Norm bezeichnet sein.

Befinden sich die Bilder mit der vorgegebenen Bezeichnung im festgelegten Bilderverzeichnis, erscheint am Platz des Symbolbefehls die erwünschte Wechselgalerie.

Die Fadeshow funktioniert in Tabellen nicht!

Die Fadeshow-Option muss vor der Verwendung eingerichtet und freigegeben werden. Dazu muss die Datei `bot_fadeshow.inc.php` neu eingefügt und die beiden Dateien `contentshow.inc.php` und `edit/config.inc.php` angepasst werden.

Einrichtvorgehen:

1. Datei `edit/config.inc.php` bezüglich Freigabemöglichkeit des Symbolbefehls erweitern
2. In der Datei `contentshow.inc.php` eine neue Zeile zur Einbindung des Symbolbefehls einfügen
3. Die Datei `bot_fadeshow.inc.php` ins Verzeichnis Minibot ablegen
4. Die Bilder neu mit `fs` kennzeichnen und ins gewünschte Bilderverzeichnis hochladen.
5. Auf der gewünschten Seite den Symbolbefehl einfügen.

Test der entwickelten Application FADESHOW

Funktionstest in Arbeitsumgebung

Ein Funktionstest zeigte, dass der Symbolbefehl {FADESHOW} funktioniert. Beim Test wurden folgende Eigenheiten festgestellt:

- Pro Seite ist nur eine einzige FADESHOW funktionsfähig (Grund: Beeinflussung der Variablen des JavaScripts)
- Bei Positionierung des Systembefehls in einer Tabelle startet die Fadeshow nicht.
- Die Positionierung des Systembefehls ist irgendwo auf der Seite möglich
- Parallel sind so viele unterschiedliche Bildershows möglich wie es Bilderverzeichnisse hat
- Die Fadershows hat auf dem iPad eine viel kürzere Fadingphase
- Ein und dieselbe Bildershow kann auf mehreren verschiedenen Seiten eingerichtet werden

Überprüfung des Einrichtverhaltens

Eingeben der Breite:

- Bei der Eingabe von Breite = 0 bzw. keiner Breite startet die Show nicht
- Bei zu grosser Breite (d.h. über 600 Pixel) ragt die Show über den Rand hinaus.
- Höhenangaben sind mit der Einheit px oder pixel möglich

Eingeben des Verzeichnisses:

- Leeres Verzeichnis führt zur Anzeige des unvollständigen Symbolbefehls.

- Falsches Verzeichnis kann zur Anzeige einer falschen Bildershow führen.
- Nicht vorhandenes Verzeichnis führt zur Anzeige des falschen Symbolbefehls.

Bilder im Verzeichnis:

- Fehlendes erstes Bild führen zu weissem Bild und Stillstand
- Fehlende Bilder führen zu Abbruch und Wiederbeginn der Bildershow
- Falscher Bildeintrag mit grossen Buchstaben wird ignoriert
- Bei minimaler Anzahl Bilder unter 2 startet die Show nicht

Anzeigedauer:

- Leeres Attribut Dauer führt zur Anzeige des falschen Symbolbefehls
- Bei Attribut Dauer mit Einheit (ms) läuft die Bildershow nicht ab
- Minimale Länge von 100ms führt zu Bilderfilm
- Maximale Länge von 100'000 funktioniert entsprechend langsam

Eingeben der Wechselbedingungen:

- Bei leerem Attribut Wechsel läuft die Bildershow nicht ab
- Bei Eingabe von Wechseldauer mit Einheit (ms) läuft die Bildershow nicht ab
- Optimale Einstellungsstufen liegen zwischen 1 bis 100.

Löschen von Bildern:

- Gelöschte Bilder erzeugen eine Lücke und führen zu Wiederbeginn der Bildershow

Position der Bilder Bildern:

- Bilder sind normalerweise linksbündig angezeigt; können aber auch eingemittet werden.

Start der Show:

- Das erste Bild wird viel zu kurz gezeigt (-> **evtl. noch verbessern!**). Am Besten wird als erstes Bild das letzte der Show verwendet.

Gleichzeitige Verwendung von 2 Fadeshows pro Seite:

- Geht nicht, da die Applikation immer dieselbe ID verwendet

Verwendung bei Versionen 1.6.3 & 1.6.8:

- Geht bei Version 1.6.3 und 1.6.8!