

Hart wechselnde Bildershow mit Symbolbefehl für CMBasic V 1.6.3 Beta & V 1.6.7

(Version 04.03.2015)

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. 500 .. 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.
- Es ist vorerst kein Fading und keine Bilderbezeichnung vorgesehen
- Die Showlänge (Bilder) muss vom Programm erkannt werden
- Die max. 99 Bilder sind in einem vorhandene Verzeichnis (z.B. images3) abgelegt.
- Die Bilder werden entsprechend der gewünschten Anzeigeposition umnummeriert (z.B. bs_1.jpg).
- Die Bilder sollen maximal 800 Pixel breit sein, damit sie normal hochgeladen werden können.
- Im Symbolbefehl soll das verwendete Bilderverzeichnis, die verwendete Galeriebreite und die Anzeigedauer 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:

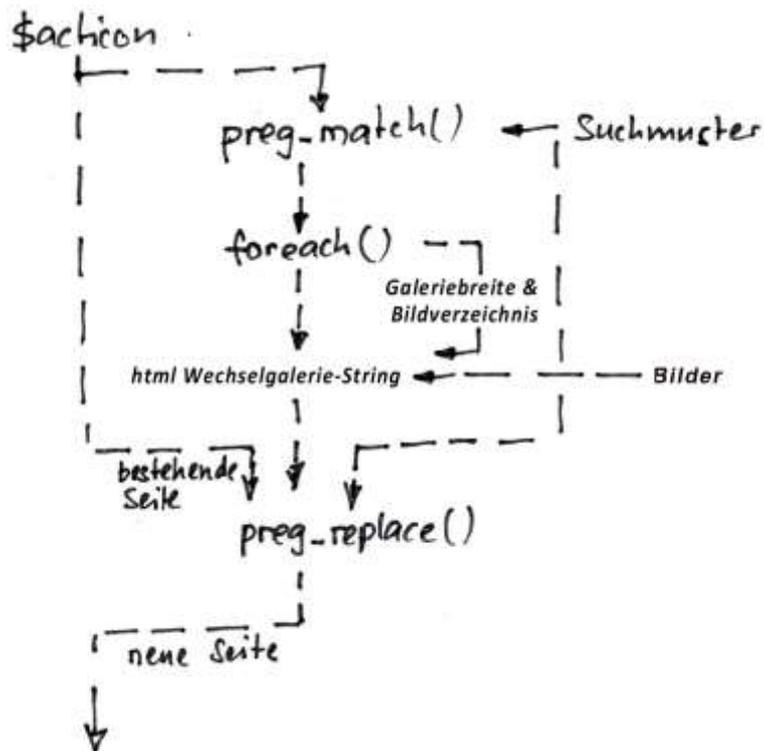
{BISHOW,Breite,Verzeichnis,Dauer}

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 freigegeben worden ist.

Nun wird der erkannte Symbolbefehl zur Datei **bot_bishow.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 den Bilderlinks die Galeriebreite, die Anzeigedauer und das nötige JavaScript.

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



Erweiterung der Datei Config.inc.php:

Neue Zeile mit Freigabemöglichkeit des Symbolbefehls einfügen:

```
define('BISHOW', TRUE); // Bildershowverwendung freigeben
```

Erweiterung der Datei Contentshow.inc.php (Zeile 551 ff)

Neue Zeilen zur Einbindung des Symbolbefehls einfügen:

```
if (defined('BISHOW') && BISHOW == TRUE)
    include_once 'minibot/bot_bishow.inc.php';
}
```

Entwicklung des Moduls „bot_bishow.inc.php“:

Verwendete Variablen:

- \$action (Seitencode)
- \$breite (eingegebene Galeriebreite in Pixel)
- \$verz (eingegebenes Bildverzeichnis bezogen auf Root?)
- \$dauer (eingegebene Anzeigedauer der Bilder in ms)
- \$galadd (Attributzusammenfassung für preg_replace am Schluss)
- \$filelist (Liste der Bilder)

- \$js (JavaScript)

Programmablauf:

- Suche nach Symbolbefehl im \$action-String mittels vereinfachtem Suchmuster
- Herauslesen der Attribute Breite, Verzeichnis und Dauer
- Feststellen der Anzahl Bilder der Bildershow
- Erstellen des Bilderlisten-Strings
- Anpassen des JavaScripts, dass es die Länge der Galerie automatisch erkennt und die Bildanzeige entsprechend steuert
- Zusammenstellen des Bildershow-html-Strings bestehend aus Bilderlisten-String, JavaScripts, Galeriebreite, vollständiges Symbolbefehl-Suchmuster und einfügen desselben in den \$action-String an Stelle des Symbolbefehls

Script der Datei bot_bishow.inc.php:

```
<?php
defined('CMBASIC') or die();
$muster_syntax = "|{BISHOW,Breite,Verzeichnis}|"; // Syntax-Muster
$suchmuster = "|{((BISHOW),(. *?))}|si"; // Suchmuster
preg_match_all($suchmuster,$action,$galfund); // durchsucht Seitenstring und gibt alle Arraywerte aus
foreach($galfund[1] as $galpara_komp) { // liest Array $galfund aus
    $galpara = explode(", ", $galpara_komp); // erstellt Array mit Einzelwerten [1] und [2] her
    if (!empty($galpara[1])) { // prüft ersten Einzelwert des Arrays galpara auf Vorhandensein
        $breite = $galpara[1]; // setzt ersten Parameter als Breite
        $galadd = "$galpara[1]"; // erstellt Attributauflistung mit erstem Attribut
    }
    if (!empty($galpara[2])) { // prüft zweiten Einzelwert des Arrays galpara auf Vorhandensein
        $verz = $galpara[2]; // setzt zweiten Parameter als Bilderverzeichnis
        $galadd .= ",$galpara[2]"; // erweitert Attributauflistung mit zweitem Attribut
    }
    if (!empty($galpara[3])) { // prüft dritten Einzelwert des Arrays galpara auf Vorhandensein
        $dauer = $galpara[3]; // setzt dritten Parameter als Anzeigedauer
        $galadd .= ",$galpara[3]"; // erweitert Attributauflistung mit drittem Attribut
    }
}
if (phpversion() >= 5) { // überprüft ob PHP-Anforderungen erfüllt sind
    if (!file_exists("$verz")) { // prüft ob Bilderordner vorhanden ist -> funktioniert richtig!
        $galcode_0 = '<p><strong>Dieser Bilderordner existiert leider nicht!</strong></p>';
    } else {
// Erstellen des Bilder-Strings aus Daten des Bilderordners $verz
$list_0 = ""; // Bilderliste wird auf leer gesetzt
$i = 1; // Schleifenzähler wird auf 1 gesetzt
while (is_file("$verz/bs_$.jpg")) { // prüft ob Bild vorhanden ist -> benötigt ""!
    if ($i == 1){
$list = $list_0."bs_$.jpg"; // setzt erstes Bild ohne Komma ein
    } else {
$list = $list_0.",bs_$.jpg"; // setzt weitere Bilder mit vorangestelltem Komma ein
    }
    $i++; // erweitert Schleifenzähler um 1
}
// JavaScript hier direkt eingefügt, da Verlinkung nicht funktionierte
$js = "<script type='text/javascript'>
var stellg = false;
```

```

function halt() {
    stellg = false;
}
function los() {
    if (stellg == false){
        stellg = true;
        beginn();
    }
}
function makeArray() {
    this.length = makeArray.arguments.length
    for (var i = 0; i < this.length; i++) this[i+1] = makeArray.arguments[i]
}
var pix = new makeArray ($list);
var eff = pix.length + 1;
var nummer = 1;
function beginn() {
    if(stellg == true) {
        document.grossbild.src = '{$verz}/'+pix[nummer];
        nummer += 1;
        if (nummer == eff ) nummer = 1;
        setTimeout('beginn()', {$dauer});
    }
}
</script>";
$galcode_0= <<<GALCODE
{$js}

GALCODE;
}
$galcode = "$galcode_0"; // html-Code mit Stringzeichen versehen
$muster2 = "|{BISHOW,$galadd}|"; // Suchmuster für Symbolbefehl inkl. Attributauflistung
$sacticon = preg_replace($muster2, $galcode, $sacticon); // Ersetzen des Symbolbefehls (muster2) durch
Galeriecode (galcode)
?>

```

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_bishow.inc.php** ins Verzeichnis Minibot ablegen
4. Die Bilder neu bezeichnen und ins gewünschte Bilderverzeichnis hochladen.
5. Auf der gewünschten Seite den Symbolbefehl einfügen.

Beschreibung fürs Handbuch

Einbinden einer Bildershow mit bis zu 99 Bildern

Wenn erwünscht, lässt sich pro Seite eine Bildershow mit bis zu 99 automatisch wechselnden Bildern gleichen Formats einfügen. Es ist kein Fading und keine Bilderbeschriftung vorhanden.

Der dazu verwendete Symbolbefehl lautet:

{BISHOW,Breite,Verzeichnis,Dauer}

Das erste fixe Attribut (BISHOW) dieses Befehls zeigt dem System an, dass an diesem Platz eine Bildershow 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 und mit dem vierten Attribut (Dauer) wird die Anzeigedauer der einzelnen Bilder eingestellt.

Als Breite ist die Pixelausdehnung des Anzeigefensters (maximal 800) 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 Dauer kann die Anzeigedauer der Bilder in Millisekunden (z.B. 2'000) festgelegt werden. Eingabe erfolgt ohne Einheit ms!

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

| | |
|-------|-------------------------|
| bs_1 | erstes Bild |
| bs_2 | zweites Bild |
| bs_3 | drittes Bild |
| bs_4 | viertes Bild |
| | |
| bs_98 | achtundneunzigstes Bild |
| bs_99 | letztes Bild |

Bemerkung: Die Nummerierung der Bilder darf nicht lückenbehaftet sein. Bei fehlenden Bildern wechselt die Show zurück zum ersten Bild.

Die Bilder der Bildershow werden nach der Umbenennung über das Register "Bilder" im Backend in das entsprechende Verzeichnis hochgeladen. Da dazu sind die Bilderverzeichnisse images1 bis images3 verfügbar sind, können parallel bis zu 3 unabhängige Diashows 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 Norm bezeichnet sein.

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

Test der entwickelten Funktion

Funktionstest in Arbeitsumgebung

Ein Funktionstest zeigte, dass der Symbolbefehl ‚BIGAL‘ funktioniert. Beim Test wurden folgende Eigenheiten festgestellt:

- Pro Seite ist nur eine einzige Bildershow funktionsfähig (Grund: Variablen-Beeinflussung der JavaScripts)
- Die Positionierung des Systembefehls ist irgendwo auf der Seite möglich
- Parallel sind bis zu 3 unterschiedliche Bildershow auf verschiedenen Seiten möglich
- Ein und dieselbe Bildershow kann auf mehreren verschiedenen Seiten eingerichtet werden

Überprüfung des Einrichtverhaltens

Eingeben der Breite:

- Leere Breite führt zu Galeriebreite wie Bilder
- Bei zu grosser Breite (d.h. über 750 Pixel) ragt die Show über den Rand hinaus.
- Breitenangaben mit Einheit px oder Pixel gehen auch

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
- Falscher Bildeintrag mit zwei Underline2 führt zu Abbruch und Wiederbeginn der Bildershow
- Minimale Anzahl Bilderzahl von 1 führt zu Daueranzeige des ersten Bildes

Löschen von Bildern:

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

Eingeben der Anzeigedauer:

- Leeres Attribut Dauer führt zur Anzeige des falschen Symbolbefehls
- Bei Attribut Dauer mit Einheit (ms) läuft die Bildershow nicht an
- Minimale Länge von 1ms führt zu Bilderfilm
- Maximale Länge von 100'000 funktioniert entsprechend langsam

Feststellungen im Programm:

- echo " <javascript>xyz</script>"; ergibt leere Anzeige. Scheinbar werden generell JavaScripts nicht angezeigt.
- \$js = " <javascript>xyz</script>"; funktioniert nur, wenn innerhalb des JavaScriptes keine doppelten Stringzeichen (") mehr enthalten sind.
- \$js = "<script type='text/javascript' src='css/wegal/wegal.js'></script>"; funktioniert nicht. Auch nicht bei readfile() wird nur Filegrösse ausgegeben!
- Direktes Einbinden von <script type="text/javascript" src=" bildershow.js"></script> in GALCODE geht auch nicht. Evtl. geht das nur unter html5?
- BISHOW lässt sich auch mittels **onMouseover="los()"** und **onClick="los()"** starten. Der Eventhandler onmouseout="halt()" kann parallel mit onMouseover verwendet werden!