

AS3-Tutorium: Flex: Butterfly 01 fluttering

Wechseln zu: [Navigation](#), [Suche](#)

Dieser Artikel erfüllt die [GlossarWiki-Qualitätsanforderungen](#) **nur teilweise**:

| | | | | |
|--|--|--|---|--|
| Korrektheit: 4 (größtenteils überprüft) | Umfang: 3 (einige wichtige Fakten fehlen) | Quellenangaben : 5 (vollständig vorhanden) | Quellenarten: 5 (ausgezeichnet) | Konformität: 5 (ausgezeichnet) |
|--|--|--|---|--|

AS3-Tutorium: Butterfly: [Flash](#) | **Flex**

Flex: [Übersicht](#) | [Teil 1](#) | [Teil 2](#) | [Teil 3](#) | [Teil 4](#) | [Teil 5](#)

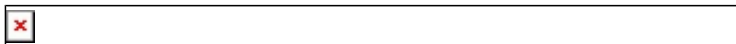
Inhaltsverzeichnis

[1 Flatternder Schmetterling](#)

- [1.1 Ein neues Flex-Projekt](#)
 - [1.2 Schmetterlingssymbol aus Flash CS5 in Flex integrieren](#)
 - [1.3 Schmetterlingssymbol aus Flash CS3, Flash CS4 oder Flash CS5 \(Alternative zu oben\) in Flex integrieren](#)
 - [1.4 Probleme und Lösungen](#)
 - [1.5 Anwendung testen](#)
- ### [2 Erweiterung: Kreis-Animation mit ActionScript](#)
- [2.1 Anmerkungen](#)
- ### [3 Quellen](#)
- [3.1 SVN-Repository-Verweise](#)

1 Flatternder Schmetterling

Im ersten Teil des Flex-Tutoriums wird der Schmetterling aus **Flash**-Tutorium „[Butterfly 01 fluttering](#)“ in eine Flex-Anwendung eingebunden.



Musterlösung (Flex 4/Flash CS5) ([SVN-Repository](#))

Musterlösung (Flex 4/Flash CS4) ([SVN-Repository](#))

1.1 Ein neues Flex-Projekt

Flash Builder 4 (deutsche Version) starten.

Die Flash-Perspektive öffnen: **Fenster** → **Perspektive öffnen** → **Andere...** → **Flash**.

Ein neues Flex-Projekt erzeugen: **Datei** → **Neu** → **Flex-Projekt** → **Projektname:**

Butterfly01Flex → **Fertig stellen**.

Im Editor-Fenster von Eclipse ist die Datei **Butterfly01Flex.xml** geöffnet. Diese Datei kann durch einen Doppelklick auf **Butterfly01Flex** → **src** → **(Standardpaket)** → **Butterfly01Flex.xml** im Fenster **Paket-Explorer** (links oben in der Standardperspektive) jederzeit geöffnet werden.

Im Editor-Fenster der Datei **Butterfly01Flex.xml** in den **Quelle-Modus** wechseln: Klick auf den Button **Quelle** (sofern dieser Button nicht bereits selektiert ist).

Im Editor-Fenster der Datei **Butterfly01Flex.xml**: `minWidth="955" minHeight="600"` durch `width="550" height="400"` ersetzen.

Wenn gewünscht wird, dass sich die Darstellungsgröße der SWF-Datei der Flex-Anwendung an die Größe des Ausgabefensters anpasst: Direkt hinter `width="550" height="400"` folgendes Attribut einfügen: `applicationComplete="stage.scaleMode=StageScaleMode.SHOW_ALL"`.

Im Editor-Fenster der Datei **Butterfly01Flex.xml** in den **Design-Modus** wechseln: Klick auf den Button **Design**. (Der Wechsel in den Design-Modus dauert bei ersten Mal i. Allg. etwas länger.)

Die Komponente **Label** aus dem Fenster **Komponenten** (links oben, zweiter Reiter in der Standardperspektive) in das Editorfenster ziehen.

Den Inhalt des Labels im Editor-Fenster in **Butterfly 01 (Flex)** abändern (Doppelklick mit der linken Maustaste).

Die Schriftgröße und -art kann, wenn die **Label**-Komponente per Mausklick selektiert wurde, im Fenster **Eigenschaften** (rechts unten, erster Reiter in der Standardperspektive) abgeändert werden.

1.2 Schmetterlingssymbol aus Flash CS5 in Flex integrieren

Im *Flash Builder*, Editor-Fenster der Datei **Butterfly01Flex.xml** in den **Design-Modus** wechseln: Klick auf den Button **Design**.

Im *Flash Builder*, Editorfenster (Design-Modus) der Datei **Butterfly01Flex.xml**: Komponente **Flash Professional**-Komponente (aus Fenster **Komponenten** → **Benutzerdefiniert**) auf die Bühne ziehen.

Im *Flash Builder*: Neue Flash-Komponente erstellen: Entweder Doppelklick auf Komponentensymbol im Editor oder Einfachklick auf Komponentensymbol im Editor + Klick auf **In Flash Professional erstellen...** im Fenster **Eigenschaften** (rechts unten in der Standardperspektive) → **Klassenname: Butterfly**, **SWC-Dateiname: Butterfly01Flash.swc** ⇒ Die Datei **Butterfly01Flash fla** wird von Flex automatisch erstellt und in einen neuen Ordner namens **assets** abgelegt.

In der Bibliothek von *Adobe Flash*: Symbol **Butterfly** löschen.

In *Adobe Flash*: Ihre Lösung **Butterfly01Flash fla** oder die Musterlösung

[Butterfly01Flash11 fla](#) von [AS3-Tutorium: Flash: Butterfly 01 fluttering](#) öffnen (eventuell befinden sich jetzt zwei Dateien mit demselben Namen in Flash; verwechseln Sie sie nicht).

In *Adobe Flash*: In der Bibliothek der Musterlösung **Butterfly01Flash fla** bzw.

Butterfly01Flash11 fla: Alle Ordner und Symbole markieren → **Strg-C** (**Strg-C**).

In *Adobe Flash*: In der Bibliothek der von Flex erstellten `fla`-Datei `Butterfly01Flash.fla`, die kopierten Symbole und Ordner einfügen: **Strg-V** (**Strg-V**).

In *Adobe Flash*: In der Bibliothek der von Flex erstellten `Butterfly01Flash.fla`: Symbol **Butterfly** selektieren.

In *Adobe Flash*: In der Menüleiste der von Flex erstellten `Butterfly01Flash.fla`: **Befehle → Symbol in Flex-Komponente konvertieren** ⇒ Das Icon von Butterfly ändert sich: Fx auf schwarzem Grund.

In *Adobe Flash*, die von Flex erstellte Datei `Butterfly01Flash.fla`: **Datei → Speichern**.

In *Adobe Flash*, links oben auf der Bühne: Klick auf Button **Fertig** ⇒ Im Flex-Ordner `assets` wird die Datei `Butterfly01Flash.swc` erstellt; diese wird automatisch in den Ordner `lib` kopiert.

Im *Flash Builder*, Editorfenster (Design-Modus) der Datei `Butterfly01Flex.xml`: Sofern die Komponente **Flash Professional-Komponente** noch vorhanden ist: Diese entfernen.

Im *Flash Builder*, Editorfenster: Sofern sich die Komponente **Butterfly** noch nicht auf der Bühne befindet: Diese (aus Fenster **Komponenten** → Benutzerdefiniert) auf die Bühne ziehen

Im *Flash Builder*, Editorfenster: Die Komponente **Butterfly** evtl. positionieren und skalieren.

Tipp: Sollte es beim Start einer Flex-Anwendung zu unerklärlichen Fehlern kommen, hilft oft:

In der Menüleiste (ganz oben): **Projekt → Bereinigen...**

1.3 Schmetterlingssymbol aus Flash CS3, Flash CS4 oder Flash CS5 (Alternative zu oben) in Flex integrieren

Voraussetzung: Ein neues Flex-Projekt wurde gemäß der Anleitung von Abschnitt [Ein neues Flex-Projekt](#) erstellt.

Die folgende Beschreibung funktioniert sowohl für Flash CS4, als auch für Flash CS5 und sollte auch für Flash CS3 funktionieren.

Im *Flash Builder*, **Paket-Explorer**: Klick mit rechter Maustaste auf den Projektordner `Butterfly01Flex` → **Neu → Ordner** → Ordnername: `assets` (Schreibweise ist bei Flash CS5 wichtig) → **Fertig** stellen.

Im *Flash Builder*, **Paket-Explorer**: Klick mit rechter Maustaste auf den Ordner `assets` → **Importieren** → **Allgemein** (Ordner durch Klick Dreiecksicon öffnen) → **Dateisystem** → **Weiter** → **Aus Verzeichnis**: hier den Ordner mit Ihrer Lösung oder der Lösung von [AS3-Tutorium: Flash: Butterfly 01 fluttering](#) wählen → Häkchen vor `fla`-Datei mit Flash-Musterlösung (`Butterfly01Flash.fla`) → **Fertig** stellen.

Im *Flash Builder*, **Paket-Explorer**: Die `fla`-Datei im Ordner `assets` durch Doppelklick in *Adobe Flash* öffnen.

In der Bibliothek von *Adobe Flash*: Symbol **Butterfly** selektieren.

In *Adobe Flash*: **Befehle → Symbol in Flex-Komponente konvertieren** (in Flash CS4 gibt es diesen Befehl nur, wenn das Plugin **Flex Component Kit** installiert wurde; siehe [Übersicht](#)) ⇒ Das Icon von Butterfly ändert sich: Fx auf schwarzem Grund..

In *Adobe Flash*: **Datei → Speichern**.

In *Adobe Flash*: **Datei → Veröffentlichen**.

Im *Flash Builder*, **Paket-Explorer**: Klick mit rechter Maustaste auf den Projektordner `Butterfly01Flex` → **Aktualisieren**.

Im *Flash Builder*, **Paket-Explorer**: Neu erstellte `swc`-Datei (`Butterfly01Flash.swc`) vom

Ordner `assets` in den Ordner `lib` verschieben.

Im *Flash Builder*, Editorfenster (Design-Modus) der Datei `Butterfly01Flex.mxml`: Komponente `Butterfly` (aus Fenster `Komponenten` → Benutzerdefiniert) auf die Bühne ziehen .

Im *Flash Builder*, Editorfenster: Die Komponente `Butterfly` evtl. positionieren und skalieren.

Tipp: Sollte es beim Start einer Flex-Anwendung zu unerklärlichen Fehlern kommen, hilft oft:

In der Menüleiste (ganz oben): `Projekt` → `Bereinigen...`

1.4 Probleme und Lösungen

Problem: Es wird keine `swc`-Datei erstellt, obwohl im Flash-Menü `Veröffentlichen` ausgeführt oder der Button `Fertig` geklickt wurde.

Lösung: `Datei` → `Einstellungen für Veröffentlichungen...` → Reiter `Flash` → Häkchen vor `SWC exportieren` → `OK`

1.5 Anwendung testen

Auf Icon `Ausführen Als...` (fünftes Icon links oben: weißes Dreieck in grünem Kreis) klicken → `Webanwendung` → `OK`.

Wenn die Datei `Butterfly01Flex.mxml` nicht per `Strg-S` (`Ctrl-S`) oder `Datei` → `Speichern` gesichert wurde, erscheint nun der Dialog `Speichern und starten` in dem die zu speichernden Dateien selektiert sind. Hier ist es sinnvoll die Option `Ressourcen vor den Start immer speichern` zu selektieren (damit dieser Dialog künftig nicht mehr angezeigt wird) und anschließend `OK` zu klicken.

Im Standard-Browser öffnet sich die Flash-Anwendung.

Im Allgemeinen ist es nicht notwendig, die Anwendung immer im Browser zu starten. Der Standalone-Flash-Player ist für Testzwecke i.Allg. vollkommen ausreichend. Ob die Debug-Version im Browser oder im Standalone-Flash-Player gestartet wird, kann man in den Projekteigenschaften festlegen.

Im `Paket-Explorer`: Klick mit rechter Maustaste auf dem Projektordner `Butterfly01Flex` → `Eigenschaften` → `Flex Compiler` → `HTML-Wrapper-Datei generieren` → kein Häkchen (⇒ Anwendung wird im Standalone-Flash-Player gestartet; Häkchen ⇒ Anwendung wird im Browser gestartet) → `OK` → `OK`.

Man kann die Anwendung auch im Debug-Modus starten (viertes Icon links oben: ein symbolisierter Käfer (= Bug)). In diesem Fall werden Fehlermeldung und `trace`-Ausgaben (`trace("Text, der nur im Debug-Modus angezeigt wird")`) im Fenster `Konsole` (unterhalb des Editor-Fensters in der Standardansicht) ausgegeben.

Für den Debuggingmodus sind allerdings besondere Flash-Player notwendig. Diese können unter <http://www.adobe.com/support/flashplayer/downloads.html> heruntergeladen werden, sofern der Flash Builder meldet, dass kein geeigneter Flash-Player gefunden werden kann.

2 Erweiterung: Kreis-Animation mit

ActionScript



[Musterlösung \(Flex 4/Flash CS5\)](#) (SVN-Repository)

[Musterlösung \(Flex 4/Flash CS4\)](#) (SVN-Repository)

Im Gegensatz zu Flash bietet der Flash Builder keine Möglichkeit Timeline-Animationen grafisch zu erstellen. Man kann allerdings Animationen mit Hilfe von ActionScript programmieren. (Es gibt sogar die Möglichkeit, in Flash eine Animation als ActionScript-Code zu exportieren und diese dann in Flex einzusetzen.)

Wenn der folgende Code in die Datei `Butterfly01Flex.mxml` (im Quellenmodus) eingefügt wird, fliegt der Schmetterling auf einer Kreisbahn und wirft dabei sogar einen Schatten.

Tip: Eventuell ist besser, zunächst den [zweiten Teil des Flex-Tutoriums](#) zu bearbeiten, bevor Sie sich diese erweiterte Musterlösung des ersten Teil näher ansehen. Der MXML-Code des zweiten Teils ist deutlich einfacher.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application
  xmlns:fx="http://ns.adobe.com/mxml/2009"
  xmlns:s="library://ns.adobe.com/flex/spark"
  xmlns:mx="library://ns.adobe.com/flex/mx"
  xmlns:std="*"
  width="550" height="400"
  applicationComplete="o_init()"
>
  <fx:Style source="css/main.css"/>

  <s:Label
    x="110" y="5"
    text="Butterfly 01 Circle (Flex 4/Flash CS4)"
  />
  <std:Butterfly
    id="sp_butterfly_movie"
    x="275" y="50" rotation="90"
    width="125.55" height="93.8"
  />

  <fx:Script>
    <![CDATA[
```

```

import flash.filters.DropShadowFilter;

private const c_x0:    Number = 275;
private const c_y0:    Number = 200;
private const c_r:     Number = 150;
private const c_delta: Number = 1.0/64;

private const c_filters: Array = [new
DropShadowFilter(30,45,0,0.8)];

private var v_phi: Number = 0;

private function o_init(): void
{
    // Scale the stage, when the output Window is scaled.
    this.stage.scaleMode=StageScaleMode.SHOW_ALL;

    // Add a shadow to the butterfly!
    sp_butterfly_movie.filters = c_filters;

    this.addEventListener(Event.ENTER_FRAME, o_move_butterfly);
}

private function o_move_butterfly(p_event: Event): void
{
    sp_butterfly_movie.x = c_x0 + Math.sin(v_phi)*c_r;
    sp_butterfly_movie.y = c_y0 - Math.cos(v_phi)*c_r;

    if (v_phi >= 2*Math.PI)
        v_phi = Math.PI*c_delta;
    else
        v_phi += Math.PI*c_delta;

    sp_butterfly_movie.rotation = v_phi*180/Math.PI+90;
}

```

```

]]>
</fx:Script>
</s:Application>

```

2.1 Anmerkungen

Die Hauptbühne wird in einem Flex-Web-Projekt mit Hilfe der Spark-Komponente `Application` definiert. In einem Flex-Air-Projekt müsste die Komponente `WindowedApplication` verwendet werden.

Im öffnenden Tag dieser Komponente werden vier Namespaces definiert:

s: Der Namespace für die moderneren Spark-Komponenten (existieren seit Flex 4; setzen mindestens eine **Flash Player** der Version 10 voraus)

mx: Der Namespace für die alten MX-Komponenten (existieren seit Flex 2; sollten nur noch verwendet werden, wenn es keine Spark-Komponenten als Ersatz gibt)

fx: Der Namespace für alle nicht-grafischen Flex-Elemente, wie z.B. AS3-Scripts, CSS-Style-Anweisungen etc.

std: Der Namespace für das Standard-Package. (Im Flash-Builder wird dafür automatisch das Kürzel **ns1** vergeben, wenn das Kürzel **std** nicht zuvor definiert wurde.)

Die Komponente **Butterfly** liegt im Standard-Package, da in Flash unter **Eigenschaften** → **Klasse** kein anderes Package angegeben wurde. Bei der **Label**-Komponente handelt es sich dagegen um eine Spark-Komponente; ihr wird daher der Namespace **s** zugewiesen.

Um von ActionScript aus auf das Schmetterlingsobjekt zugreifen zu können, wird ihm der Identifikator **sp_butterfly_movie** zugewiesen. Das Attribut **id** im öffnenden Tag einer MXML-Komponente entspricht dem Attribut **name** („Instanzname“), das einem Objekt auf der Bühne in Flash zugewiesen werden kann.

Sobald die Flex-Anwendung vollständig gestartet wurde und insbesondere auch die Hauptbühne **stage** erstellt wurde, signalisiert die Komponente **Application** bzw. **WindowedApplication** dies mit dem Ereignis **ApplicationComplete**. (Im Gegensatz dazu wird **CreationComplete** bereits signalisiert, wenn die Hauptbühne **stage** noch nicht initialisiert, d.h. gleich **null** ist.)

Im obigen Beispiel wird veranlasst, dass die **Observer**-Methode **o_init** aufgerufen wird, sobald das Ereignis **ApplicationComplete** eintritt. Die Methode **o_init** sorgt zunächst dafür, dass sich die Größe der Bühne an die Größe des Ausgabefensters anpasst. (Dafür ist es notwendig, dass **stage** bereits initialisiert wurde.)

Da es sich bei **ButterflyMovie** um ein Objekt der Klasse **MovieClip** handelt, kann man dem Schmetterling ein Array mit Filtern zuweisen, wie zum Beispiel einen Filter, der einen Schattenwurf simuliert (`new DropShadowFilter(30,45,0,0.8)`).

Zu guter Letzt wird in der Methode **o_init** festgesetzt, dass bei jedem **ENTER_FRAME**-Event (also jede vierundzwanzigstel Sekunde, wenn Framerate 24 Bilder pro Sekunde beträgt) die **Observer**-Methode **o_move_butterfly** aufgerufen wird. Diese Methode berechnet jeweils die neue Position des Schmetterlings (auf einer Kreisbahn) und dreht den Schmetterling (`sp_butterfly_movie.rotation`) in Flugrichtung.

3 Quellen

[Kowarschick, W.: Multimedia-Programmierung](#)

[Musterlösung \(Flex 4/Flash CS5\)](#)

[Musterlösung \(Flex 4/Flash CS4\)](#)

[Erweiterte Musterlösung \(Kreisflug, Flex 4/Flash CS5\)](#)

[Erweiterte Musterlösung \(Kreisflug, Flex 4/Flash CS4\)](#)

3.1 SVN-Repository-Verweise

[Musterlösung \(Flex 4/Flash CS5\)](#)

[Musterlösung \(Flex 4/Flash CS4\)](#)

Erweiterte Musterlösung (Kreisflug, Flex 4/Flash CS5)

Erweiterte Musterlösung (Kreisflug, Flex 4/Flash CS4)

Dieser Artikel ist [GlossarWiki-konform](#).

Kategorien:

[AS3-Tutorium: Flex: Butterfly](#)

[Flex-Beispiel](#)

[Kapitel:Multimedia-Programmierung:Beispiele](#)

Diese Seite wurde zuletzt am 29. November 2010 um 18:53 Uhr bearbeitet.

Inhalt verfügbar unter [CC BY-SA 4.0](#).

