

Dokumentation der XML-Notation des Bikeschein Frameworks

Autor: Gregor Sklorz (Gregor@Sklorz.net)

Stand: 22.10.2008

Inhaltsverzeichnis

1. Einführung	3
1.1. Grundsätzliches	3
1.2. Struktur	3
1.2.1. config.xml	4
2. Darstellung	6
2.1. Tiefenverwaltung	7
2.2. Masken	7
2.3. Item Typen	7
2.3.1. Content.....	8
2.3.2. Text.....	9
2.3.3. Picture	10
2.3.4. SWF	10
2.4. Item Subtypen	12
2.4.1. Interactive.....	12
2.4.2. Scrollable	15
2.4.3. Transform	16
2.4.4. Border	17
3. Meta Informationen	18
3.1. Hauptseite	18
3.2. Feedback.....	20
3.3. Login.....	20
3.3.1. Anmeldung / Daten ändern.....	22
3.3.2. Passwort Recovery	23

1. Einführung

Das Framework für Bikeschein basiert gänzlich auch ActionScript3 und wird über XML geregelt.

1.1. Grundsätzliches

Es wird XML der Version 1 benutzt. Der Standard XML Tag ist:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

Alle Namen der XML Tags werden klein geschrieben solange nicht anders angegeben.

Ausgehend von der Hauptapplikation „main.as“, können die XMLs in einem beliebigen Unterordner gespeichert werden.

Alle angegebenen URLs sind relativ zur Hauptapplikation.

Alle Änderungen im Verhalten der Seiten müssen mit Flex oder Flash bearbeitet werden.

1.2. Struktur

Die Darstellung der Seiten und Unterseiten werden durch Darstellungsflächen (Content Elemente) ermöglicht. Jeder Content benötigt eine eindeutige ID und wird mit einer content - XML dargestellt. Contents können sich unter einander verschachteln.

Ausgehend von der Hauptseite (mainsite) können weitere Contents als Seitenelemente implementiert werden. Die Koordinaten der einzelnen „item“ Elemente gelten für den Content, nicht für das Hauptfenster.

1.2.1. config.xml

Die config.xml ist der Ausgangspunkt der XML Notation. Diese enthält grundsätzliche Informationen zum Seitenaufbau.

Der Name „config.xml“ ist im Code festgelegt.

Die config.xml muss im Ordner bei der Hauptapplikation liegen.

Das Wurzelement ist:

```
<data typ="config">
```

...

```
</data>
```

Die Elemente sind:

Mainsite:

```
<mainsite><![CDATA[mainsite.xml]]></mainsite>
```

URL der Hauptseiten XML.

Mainmodelid:

```
<mainmodelid><![CDATA[mainsite]]></mainmodelid>
```

Interne ID des Hauptseiten Models

Xmlfolderpath:

```
<xmlfolderpath><![CDATA[xml/]]></xmlfolderpath>
```

URL Prefix für alle weiteren XML Dateien.

Ressourcefolderpath:

```
<ressourcefolderpath><![CDATA[ressource/]]></ressourcefolderpath>
```

URL Prefix für alle Ressourcen welche durch die XML Elemente geladen werden.

Phpurl:

```
<phpurl><![CDATA[/interaktiv/_]]></phpurl>
```

URL Prefix für alle Frameworkseitigen PHP Aufrufe.

BackgroundColor:

```
<backgroundcolor>0x68A5CC</backgroundcolor>
```

Standart Hintergrundfarbe der Hauptseite.

Popupxml:

```
<popupxml><![CDATA[popup.xml]]></popupxml>
```

Content XML für die Darstellung der Servernachrichten.

Loginxml:

```
<loginxml><![CDATA[login.xml]]></loginxml>
```

Content XML für die Darstellung des Loginfensters.

2. Darstellung

Seiten werden durch XML item-Elementen gebildet. Sie sind die Hauptbestandteile der Content-XML. Ein „item“ stellt ein Element der Content-Seite dar.

Das Wurzelement ist:

```
<site typ="content">
```

```
...
```

```
</site>
```

Darin befinden sich die **Item oder Meta Elemente**:

```
<item typ="typename" x="0" y="0" ... ><![CDATA[wert]]></item>
```

Die X- und Y- Koordinaten können auch über die Bildfläche hinausgehen.

Items können mit der „period“ Eigenschaft auch zeitversetzt angezeigt werden.

```
<item typ="typename" period="3"><![CDATA[wert]]></item>
```

Der „period“ Wert gibt die Sekunden an, nachdem das Element dargestellt wird. Bis maximal 5 Sekunden.

2.1. Tiefenverwaltung

Die Tiefe eines Elements wird automatisch zugewiesen. Das erste item ist das unterste, das letzte item ist das oberste.

Die Tiefe kann mit der „depth“ Eigenschaft direkt zugewiesen werden:

```
<item typ="typename" depth="3"><![CDATA[wert]]></item>
```

2.2. Masken

Auf jeden Content kann eine Maskierung der Elemente darüber gelegt werden. Dazu bekommt das XML item Tag des Maskenelements die ID „mask“ zugewiesen.

```
<item typ="typename" id="mask"><![CDATA[wert]]></item>
```

2.3. Item Typen

Die Itemtypen bilden die grundlegenden Darstellungselemente.

Die Art der Typen wird mit der Eigenschaft „typ“ festgelegt. Weitere Eigenschaften bestimmen die Eigenschaften des Erstellten Elements.

```
<item typ="typename" ><![CDATA[wert]]></item>
```

Fette **Eigenschaftsname** sind Pflicht Eigenschaften und haben keinen Standardwert.

Datentyp ist die logische art des Datentyps: String, Integer usw.

„int“ = Integer

„Number“ = Gleitkomma Nummer

„RGB“ = entsprechen einer Hexadezimal Notation: **0xRRGGBB**

„bool“ = **true** oder **false**

„0 - 1“ = Gleitkommawert zwischen 0 und 1

2.3.1. Content

Darstellungsfläche für weitere Items mit Hilfe einer weiteren XML.

XML Wert = URL zu der XML Datei des Contents.

Eigenschaft	Beschreibung	Datentyp	Standart
typ	muss "content" sein	String	
x	Abstand vom oberen Rand	int	0
y	Abstand vom linken Rand	int	0
width	Breite des Contentfläche	Number	1
height	Höhe der Contentfläche	Number	1
id	ID des Erstellten Contents	String	
alpha	Transparenz des Hintergrunds	0 - 1	0
backgroundcolor	Frabe des Hintergrunds	RGB	0x000000
rotation	Winkel für die Drehung des Contents in Grad	int	0

2.3.2. Text

Text Element. Wird dynamisch erzeugt.

XML Wert = Text des Textelements.

Anmerkung:

Benutzte Schriftarten müssen in der Hauptapplikation als ActionScript Meta Tag eingebunden werden, sonst werden Sie nicht dargestellt.

- Aktuell sind „Arial“ und „Crillee“ eingebunden.

Eigenschaft	Beschreibung	Datentyp	Standart
typ	muss "text" sein	String	
x	Abstand vom oberen Rand	int	0
y	Abstand vom linken Rand	int	0
width	Breite des Contentfläche	Number	100
height	Höhe der Contentfläche	Number	dynamisch
font	Schriftart des Texts	String	Arial
color	Farbe des Texts	RGB	0x000000
italic	Kursiv	bool	false
bold	Fett	bool	false
size	Grösse der Schrift	int	12
underline	Unterstrich	bool	false
selectable	Text ist markierbar	bool	false
password	Text ist ein einzeliges Passwort Eingabefeld	bool	false
input	Text ist ein einzeliges Eingabefeld	bool	false

multiline	Text kann über mehrere Zeilen gehen	bool	true
rotation	Winkel für die Drehung des Texts in Grad	int	0

2.3.3. Picture

Statisches Bildelement.

Kann folgende Dateitypen laden:

BMP, JPG, PNG, GIF, SWF (mit AS1 – AS3)

XML Wert = URL der Bildressource.

Eigenschaft	Beschreibung	Datentyp	Standart
typ	muss "picture" sein	String	
x	Abstand vom oberen Rand	int	0
y	Abstand vom linken Rand	int	0
rotation	Winkel für die Drehung des Bildes in Grad	int	0

2.3.4. SWF

Animiertes Element. Optional mit enthaltener Logik.

Mögliche Dateitypen: SWF (nur AS3)

SWF Elementen können Variablen zu gewiesen werden welche in der SWF Applikation verarbeitet werden können.

Die Zuweisung funktioniert über die „varX“ Eigenschaft. X ist eine laufende Nummernfolge von 1 bis n, welche nicht unterbrochen werden darf.

```
<item typ="swf" var1="wert1" var2="wert2" var4="wert4">
```

```
<![CDATA[wert]]></item>
```

Die Werte haben zwei Sonderfälle.

Der Wert „md5hash“ übergibt den „key“ Parameter aus der URL.

Der Wert „user“ übergibt ein Objekt mit den Daten des aktuell angemeldeten Users. Bei nicht angemeldetem User wird das Login geöffnet.

XML Wert = URL der SWF Applikation.

Eigenschaft	Beschreibung	Datentyp	Standart
typ	muss "swf" sein	String	
x	Abstand vom oberen Rand	int	0
y	Abstand vom linken Rand	int	0
rotation	Winkel für die Drehung des Contents in Grad	int	0
varX ...	Variablenzuweisung an die SWF Applikation. X steht für eine laufende Ziffer von 1 bis n.	variabel	-

2.4. Item Subtypen

Subtypen werden als „subtypes“ Eigenschaft, kommasepariert im XML item-Tag eingetragen.

```
<item subtypes="subtypname, subtypname, ..." >  
<![CDATA[wert]]></item>
```

2.4.1. Interactive

Klickbares Element mit verschiedenen Möglichkeiten. Drei verschiedene Auswirkungen werden unterschieden, welche jeweils bestimmte Eigenschaften benötigen.

Interner Link

Öffnet eine XML im betreffenden Content Element.

Eigenschaft	Beschreibung	Datentyp	Standart
subtypes	muss "interactive" enthalten	String	
url	URL der Ziel-XML	String	
destination	ID des bestehenden Ziel-Content Elements	String	

Anmerkung:

Um einen internen Link innerhalb eines Texts (als HTML <a> Tag) zu setzen muss in der die „href“ Eigenschaft die XML URL und ID des Contents eingetragen werden:

```
<a href="event:folder/content.xml, contentID">
```

Externer Link

Öffnet einen externen Link im Browser.

Eigenschaft	Beschreibung	Datentyp	Standart
subtypes	muss "interactive" enthalten	String	
url	URL des externen Links (es wird kein Präfix benutzt)	String	
target	Link Target - "_self", "_blank", "_parent" oder "_top"	String	_self

Aktion

Setzt ein internes Ereignis ab, welches an verschiedenen Stellen abgefangen werden kann. Diese müssen vorher im an betreffender Stelle im Programm definiert werden.

Eigenschaft	Beschreibung	Datentyp	Standart
subtypes	muss "interactive" enthalten	String	
action	Vordefinierter Aktionsbefehl - "action", "return", "close" oder "info"	String	
info	Falls action="info" dann Übergebene Information	String	-
parameter	Falls action='unbekannt' dann Parameter der action zur Weiterverarbeitung	String	-

Der Controller der Bikeschein Applikation verarbeitet weitere action/parameter Kombinationen welche nicht durch das Element definiert sind.

action	parameter	Reaktion
initlogin	-	startet das Loginfenster
initlernsystem	-	startet das Lernsystem von Bikeschein oder das Login
initchallenge	-	startet die Bikeschein Challenge oder das Login
initforum	-	startet das Forum von Bikeschein oder das Login

2.4.2. Scrollable

Bildet das Item Element scrollfähig ab und stellt die Bedienelemente zu Verfügung.

Eigenschaft	Beschreibung	Datentyp	Standart
subtypes	muss "scrollable" enthalten	String	
height	Neue Höhe des scrollfähigen Elements	int	100
scrollheight	Neue Höhe des scrollfähigen Elements, falls "height" bereits vom Item verwendet wird	int	height
scrollcolor	Farbe der Bedienelemente	RGB	0x000000
scrollalpha	Transparenz der Bedienelemente	0 - 1	1

2.4.3. Transform

Bildet das Item Element Maus-reaktiv ab. Die vordefinierten Reaktionen werden kommasepariert in der „transformations“ Eigenschaft eingetragen. Diese können jeweils verschiedene Eigenschaften benötigen. Für die Skalierungswerte können auch höhere als 100% benutzt werden. Beispielsweise „2,3“ , würde das Element auf 230% skalieren.

Eigenschaft	Beschreibung	Datentyp	Standart
subtypes	muss "transform" enthalten	String	
transformations	Kommaseperierte Transformationen des Elements - "scale" und, oder "springScale"	String	
targetscalex	Prozentuale Breite nach Animation	0 - 1 - *	1
targetscaley	Prozentuale Höhe nach Animation	0 - 1 - *	1
startscalex	Prozentuale Breite vor der Animation	0 - 1 - *	1
startscaley	Prozentuale Höhe vor der Animation	0 - 1 - *	1

2.4.4. Border

Bildet einen Rahmen um das Item Element.

Eigenschaft	Beschreibung	Datentyp	Standart
subtypes	muss "border" enthalten	String	
bordercolor	Farbe des Rahmens	RGB	0x000000
thickness	Breite des Rahmens	int	12

Das Framework kann mit weiteren Typen, Subtypen oder Transformationen ausgestattet werden.

3. Meta Informationen

Für verschiedene Funktionalitäten von Bikeschein werden Meta Tags benötigt.

Diese werden in die Content XMLs eingetragen.

```
<meta eigenschaft="wert" ... ></meta>
```

Meta Tags haben keine bestimmten Eigenschaften und auch keinen Wert. Sie übergeben nur Werte wo sie benötigt werden.

Durch die Meta Tags können auch einzeln stehende Funktionen dargestellt werden.

3.1. Hauptseite

Die Hauptseite wurde für Bikeschein mit einigen Funktionen ausgestattet.

Die Grafiken am unteren Rand und die Hintergründe werden mit Meta tags angesprochen. Jede Content XML kann einen Meta Tag enthalten mit den Eigenschaften „background“ und „bottom“. Diesen Eigenschaften werden die IDs der betreffenden Grafiken als wert übergeben. Oder „none“ um nichts anzuzeigen. Die betreffende Grafik wird beim Aufruf der jeweiligen XML angezeigt. Die Grafiken selbst werden in der Hauptseite (mainsite.xml) eingebunden.

```
<meta bottom="home" background="none"></meta>
```

Das Menu der Hauptseite wird ebenfalls mit Meta Tags eingerichtet.
Dazu werden dem Meta Tag auf der Hauptseite folgende Eigenschaften mitgegeben:

bigmenux	= X Koordinate des Hauptmenus
bigmenuy	= Y Koordinate des Hauptmenus
bigmenudistance	= Vertikaler Abstand zwischen den Elementen
smallmenux	= X Koordinate des unteren Menus
smallmenuy	= Y Koordinate des unteren Menus
smallmenudistance	= Vertikaler Abstand zwischen den Elementen

```
<meta bigmenux="20" bigmenuy="140" bigmenudistance="-3"  
smallmenux="20" smallmenuy="440" smallmenudistance="1"></meta>
```

3.2. Feedback

Das Feedback Formular wird mit folgendem Meta Tag gestartet:

```
<meta init="feedback" x="30" y="100" url="delegatefeedback.xml">
</meta>
```

Die „delegatefeedback.xml“ enthält Definitionen der Inhalte für das Formular. In dieser XML müssen je ein Element mit der ID „inputfield“ und „send“ sein. Diese kennzeichnen das Eingabefeld („inputfield“) und den Senden-Button („send“)

```
<item typ="text" id="inputfield" input="true" multiline="true">
</item>
<item typ="picture" id="send" subtypes="interactive" action="accept">
<![CDATA[swf/absendenbutton.swf]]> </item>
```

3.3. Login

Das Login Fenster wird bereits beim Start mit Hilfe der config.xml erstellt.

Es wird jeweils automatisch bei Bedarf dargestellt, mit einem Link (siehe oben Interactive) oder mit einem Meta Tag beim Aufruf einer XML Seite.

```
<meta init="login"> </meta>
```

In der login.xml, welche durch die config.xml eingetragen wird, steht ein Meta Tag mit URL für XML Dateien welche die jeweiligen Untermenus des Logins erstellen.

`<meta x="220" y="0"` – X und Y Koordinaten des Loginfensters

`registerurl="register.xml"` – XML für die Anmeldeformular Seite

`changedataurl="changeuserdata.xml"` – XML für die Daten ändern Seite

`passwordrecoverurl="passwordrecover.xml">` - XML für die „Passwort wiederherstellen“ Seite

`</meta>`

Die „login.xml“ muss dazu noch einige Benötigte Elemente implementieren. Diese werden mit der „id“ Eigenschaft gekennzeichnet.

Login Elemente

ID	Element
closebutton	„Login schließen“ Knopf
registerbutton	„Anmeldung öffnen“ Knopf
changedatabutton	„Daten ändern öffnen“ Knopf
logoutbutton	„Ausloggen“ Knopf
loginbutton	„Einloggen“ Knopf
passwordrecoverbutton	„Passwort wiederherstellen öffnen“ Knopf
name	Nickname Eingabefeld
password	Passwort Eingabefeld

3.3.1. Anmeldung / Daten ändern

Die Anmelde und „Daten ändern“ Seiten haben einen Meta Tag für ihrer Koordinaten (relativ zum Login Fenster) und bestimmte benötigte Elemente für die Darstellung.

```
<meta x="0" y="0"></meta>
```

Anmeldung und Daten ändern Elemente

ID	Elemente
closebutton	„Fenster Schliesen“ Knopf
inputframe	Ein SWF Element bei dem alle Benutzerdaten übergeben und ausgelesen werden. Die Anwendung erwartet ein „COMPLETE“ Event von der SWF um die Daten auszulesen

3.3.2. Passwort Recovery

Die XML zum zurücksetzen des Passwort benötigt einen Meta Tag mit den X, Y Koordinaten (Relativ zum Loginfenster).

```
<meta x="50" y="200"></meta>
```

Und Elemente folgenden IDs:

Tabelle 1

ID	Element
headtext	Überschrift des Formulars
infotext	Infotext im Formular
username	Nickname Eingabefeld
email	Email Eingabefeld
okbutton	„Daten abschicken“ Knopf
abortbutton	„Formular schließen“ Knopf

4. Fallbeispiel

Für einen schnellen Einstieg in die XML Struktur kann dieses Beispiel benutzt werden.

4.1. News Hinzufügen

Alle News befinden sich im Ordner „**xml\news**“

Der Dateiname einer News-XML ist frei wählbar. In diesem Fall sind es das News-Datum ohne Punkte. Z.B: **29102007.xml**

Anmerkung: Erstellte Elemente können im Nachhinein nicht mehr geändert werden.

Wegen dem Highlight der News-Links (Rot statt Schwarz) muss die komplette Seite für jede News neu erstellt werden. Deswegen gibt es zu jeder News eine betreffende Seite. Z.B: **29102007news.xml**

Diese Seite enthält die Links für die News und wird auch beim Klick auf News im Hauptmenu gestartet.

Um neue News einzufügen kopiert man am besten einfach bestehende News im „**news**“ Ordner und benennt sie um. Z.B: **30102008.xml** und **30102008news.xml**

In der **30102008.xml** wird die News in das „Text Item“ eingefügt. Siehe bestehende **29102007.xml** Datei als Beispiel.

In der **30102008news.xml** sollte man nun die neue XML eintragen:

```
ersion="1.0" encoding="ISO-8859-1" ?>
.type="content">
```

```
<item typ="swf" x="30" y="20"><![CDATA[swf/news/head.swf]]></item>
<item typ="content" x="70" y="210" width="" height="" id="newscont" backgroundColor="0x00000000" alpha="1"><![CDATA[news/29102007.xml]]></item>
<item typ="text" x="500" y="170" width="120" font="arial" size="11" subtypes="interactive" url="news/01092008news.xml" destination="contentsite"><![CDATA[
<item typ="text" x="500" y="240" width="120" font="arial" size="11" subtypes="interactive" url="news/08022008news.xml" destination="contentsite"><![CDATA[
<item typ="text" x="500" y="310" width="120" font="arial" size="11" color="0xd60021" subtypes="interactive" url="news/29102007news.xml" destination="cont
<item typ="text" x="500" y="380" width="120" font="arial" size="11" subtypes="interactive" url="news/26032007news.xml" destination="contentsite"><![CDATA[
```

Content Item

Link Item

Dazu muss die XML der News im Content Item eingetragen werden:

```
<item typ="content" x="70" y="210" width="" height="" id="newscont"
backgroundcolor="0x00000000"
alpha="1"><![CDATA[news/30102008.xml]]></item>
```

Und der Betreffende Link sollte erstellt oder ersetzt werden:

```
<item typ="text" x="500" y="310" width="120" font="arial" size="11"
color="0xd60021" subtypes="interactive"
url="news/30102008news.xml" destination="contentsite">
<![CDATA[><b> 30-10-2008 </b><br>Ein Monat geht erfolgreich
zu Ende]]></item>
```

„y“ gibt die vertikale Position des Links an.

„color“ gibt dessen Farbe an. Da nur der aktive Linke Rot sein soll, sollte auch nur der aktive Link die „color“ Eigenschaft erhalten.

„url“ gibt an welche News-XML gestartet werden soll. Im aktiven Link ist es unnötig da diese bereits angezeigt wird.

Im CDATA wird der News Titel eingetragen.

Dieses Item sollte nun in alle anderen News XML kopiert werden, oder die alten ersetzen. In allen anderen News-XML ist der neue News-Link nicht der aktive. Daher sollte die „color“ Eigenschaft des Items entfernt werden.

Danach kann die neue News direkt gelesen werden. Dazu sollte man testen oder Y-Position des Links stimmt, und auch bei jeder News die richtigen Links rot oder schwarz sind.

Soll diese News direkt zuerst gestartet werden. Beim Klick auf News im Hauptmenu z.B. dann muss die „url“ Eigenschaft

url="news/30102008news.xml" in den Links der **mainsite.xml** eingetragen werden.