BlogCFC XHTML tuning mit ColdFusion Builder

Erfasst am : 30. November 2009 17:32 | Erfasst von : Andreas Schuldhaus
Verknüpfte Kategorie(n): ColdFusion Builder, ColdFusion, Entwicklung

BlogCFC ist eine excellente ColdFusion Blog Software. Programmiert und als Open Source veröffentlich von **Raymond Camden**. Sie wird von zahlreichen Entwicklern weltweit als bevorzugte blogging Software verwendet. BlogCFC kann **hier** heruntergeladen werden.

In einer kurzen Schritt für Schritt Anleitung zeige ich wie man die Suchfunktion von ColdFusion Builder und die Entwickler Plugins von Firefox zum Auffinden und Ersetzen von nicht standardkonformem Code in BlogCFC verwenden kann.

Warum sind standardkonforme, XHTML kompatible Webseiten überhaupt ein Thema und warum sollte man valide Seiten anstreben?

Standardkonformität

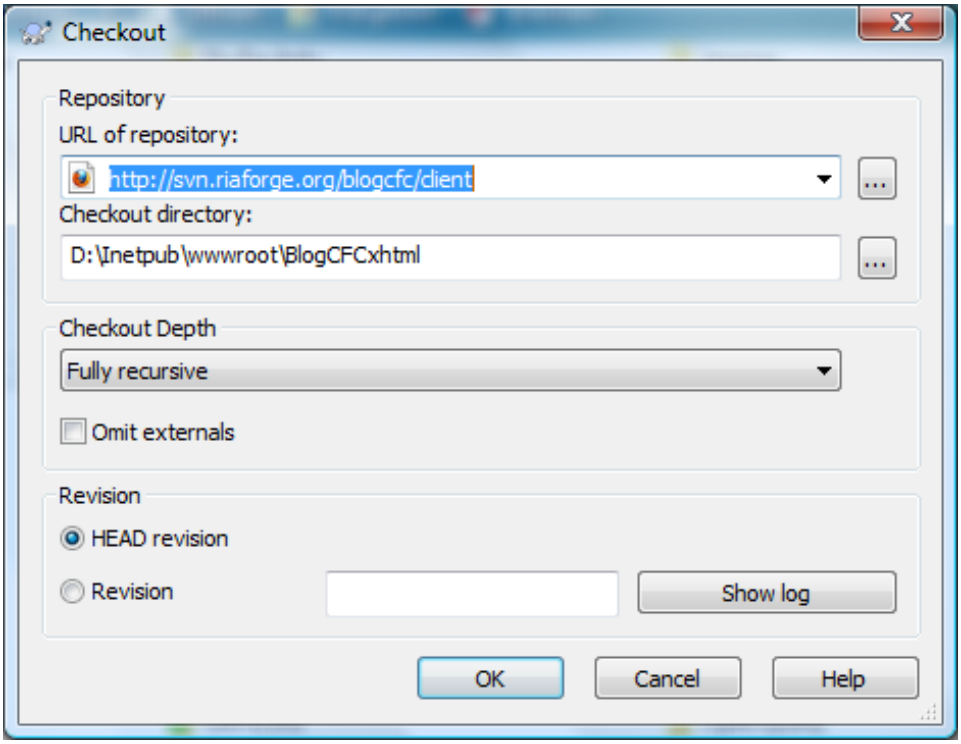
- ist ein sichtbares und messbares Qualitätsmerkmal
- spricht für Sorgfältigkeit des Entwicklers/Designers.
- ist besonders wichtig weil es die erforderliche Basis für Barrierefreiheit (Accessibility) ist.
- ist suchmaschinenfreundlich (SES) - Webseiten werden besser in Suchmaschinen gefunden.
- Man vermeidet unerklärliche und unerwartete Fehler bei der DOM-Manipulation mit Javascript (Ajax etc.), wenn der Code wohlgeformtes XHTML darstellt.

Voraussetzung für diese Schritt für Schritt Anleitung:

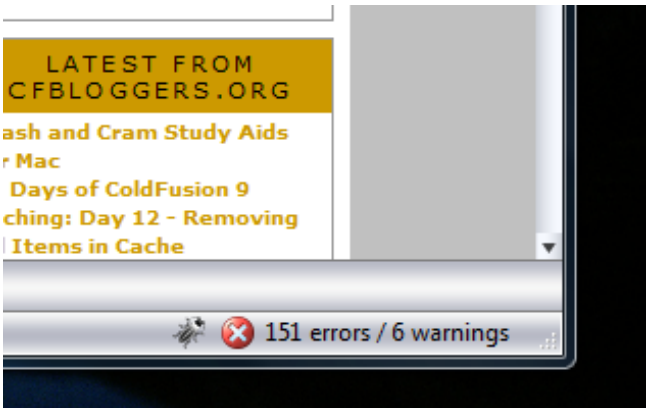
- eine aktuelle und funktionierende Installation von BlogCFC in einer Test/Development Umgebung. Wenn möglich mit ein paar eingegebenen Artikeln und Kommentaren.
- ColdFusion Builder Beta 2 oder aktueller.
- FireFox mit Web Developer Toolbar und HTMLValidator Addon.
- SVN Tool (TortoiseSVN oder ähnliches).

Wir erstellen ein neues Verzeichnis BlogCFCxhtml im lokalen Arbeitsverzeichnis oder im Webroot eines Test/Development Servers.

Mit TortoiseSVN oder über die Commandline holen wir die aktuelle Version von BlogCFC aus dem SVN Repository **http://svn.riaforge.org/blogcfc/client** und checken das Verzeichnis "client" aus.



Mit dem Browser rufen wir nun http://localhost/BlogCFCxhtml/ auf. BlogCFC greift auf die Config Daten der bereits vorinstallierten Version zu und zeigt die bereits eingegebenen Einträge und Kommentare an. Das Validator Plugin im Firefox zeigt (je nach Anzahl von Einträgen, Kategorien) Validierungsfehler an.



Wir erstellen im Verzeichnis BlogCFCxhtml ein neues ColdFusion Builder Projekt "BlogCFCx" und ein Working Set BlogCFCx. Damit haben wir die Voraussetzung geschaffen, die erforderlichen Änderungen mit ColdFusion Builder durchführen zu können.

Da wir grundsätzlich XHTML Konformität herstellen wollen, ändern wir nun den Doctype von

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.
```

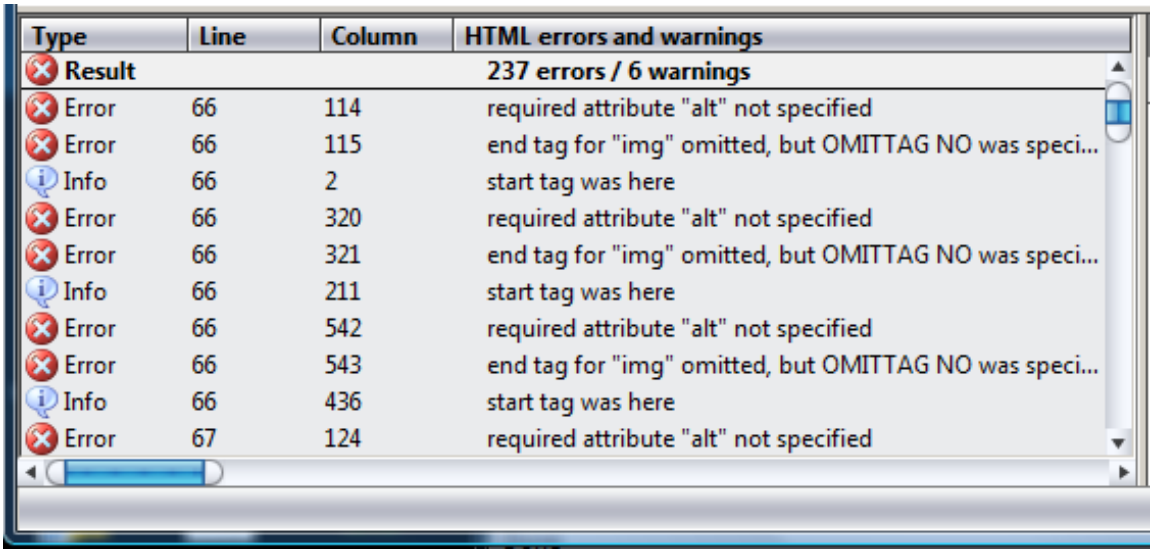
auf

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Die Suche nach <!doc ergibt im CFBuilder 5 Treffer. In der Administration wollen wir nichts ändern und der Eintrag ist auch bereits XHTML 1.0 strict. Es bleiben also 4 Templates in denen wir die Zeilen ersetzen müssen. Das können wir leicht per Copy&Paste durchführen.

Mit http://localhost/BlogCFCxhtml/index.cfm?reinit=1 rufen wir die Änderungen auf. Peng! Vorübergehend erhöht sich die Anzahl der Fehler!



Type	Line	Column	HTML errors and warnings
Result			237 errors / 6 warnings
Error	66	114	required attribute "alt" not specified
Error	66	115	end tag for "img" omitted, but OMITTAG NO was speci...
Info	66	2	start tag was here
Error	66	320	required attribute "alt" not specified
Error	66	321	end tag for "img" omitted, but OMITTAG NO was speci...
Info	66	211	start tag was here
Error	66	542	required attribute "alt" not specified
Error	66	543	end tag for "img" omitted, but OMITTAG NO was speci...
Info	66	436	start tag was here
Error	67	124	required attribute "alt" not specified

Aber keine Angst, jetzt geht es erst richtig los. Das HTML Validator Plugin, zeigt uns auch sofort was die Probleme sind. Nicht korrekt geschlossene Tags wie img, input, meta etc., fehlende alt Attribute in img Tags und eine ganze Reihe von anderen Problemen werden aufgelistet.

Für Tags, die nicht korrekt geschlossen sind, bietet sich ein Global Search mittels Regular Expressions an. Wir schränken die Suche auf Templates mit den Endungen .cfm, .cfc ein.

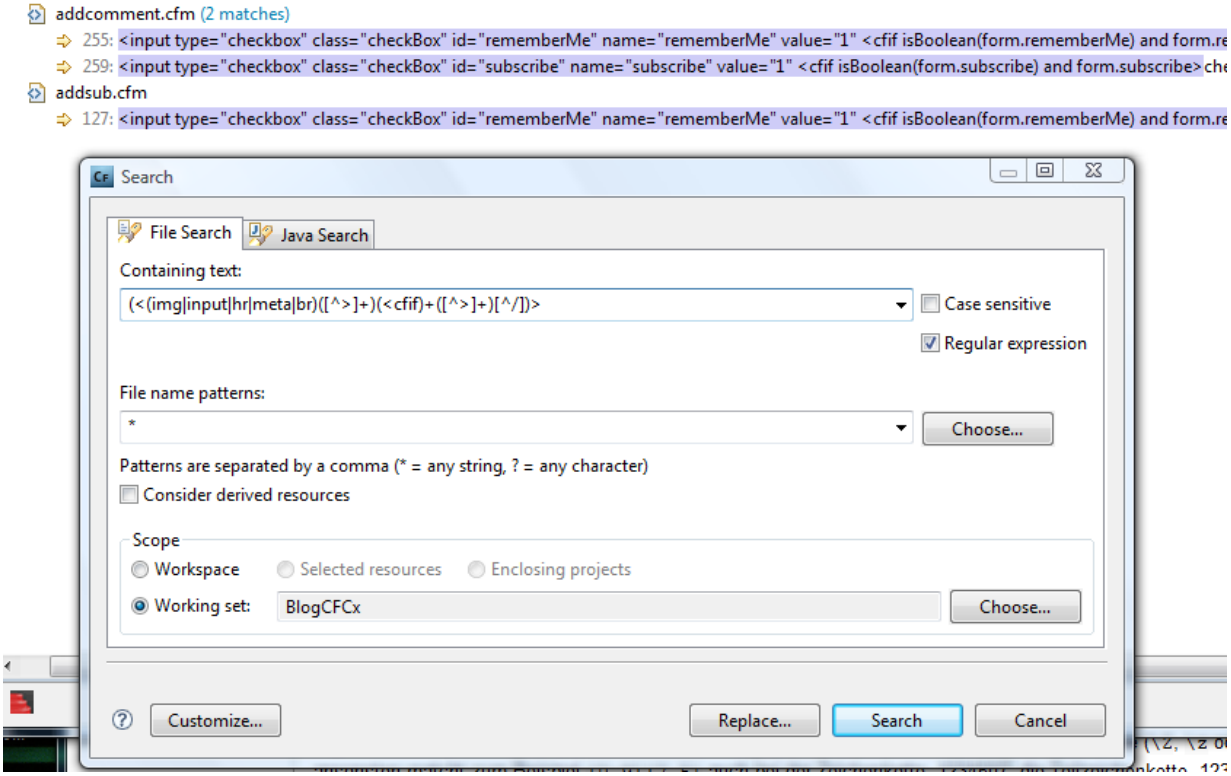
Als erstes ersetzen wir alle

```
<br> Tags mit <br />
```

Danach wenden wir uns allen anderen nicht korrekt geschlossenen Tags zu. Wir suchen zuerst einmal die Sonderfälle, in denen innerhalb eines Tags ein <cfif auftritt. Das ist deshalb wichtig, weil sonst ein Replace den cfif Tag schliessen und einen Error erzeugen würde.

```
(<(img|input|hr|meta)([^>]+)(<cfif)+([^>]+)[^/])>
```

zeigt uns alle gewünschten Ergebnisse.

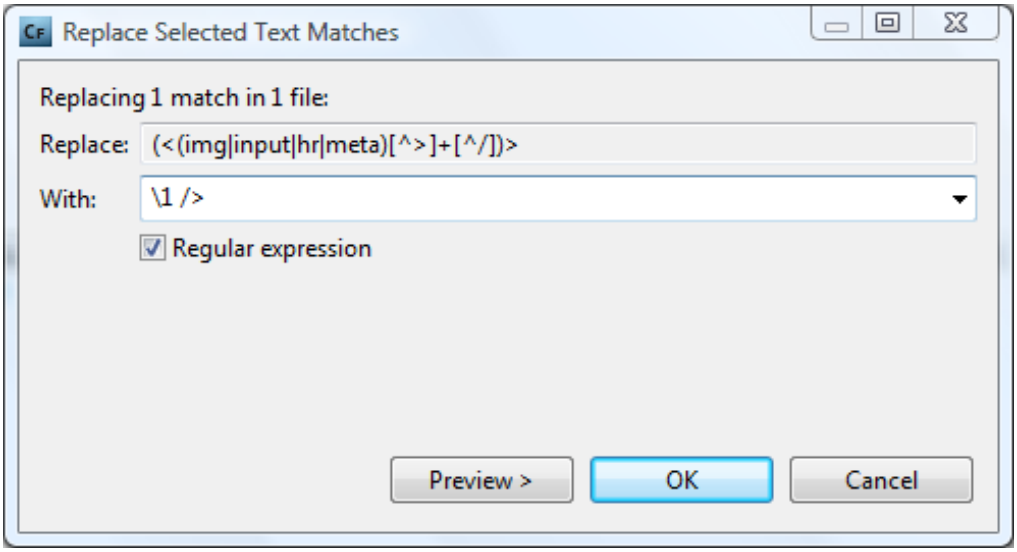


Das viermalige Auftreten des Falles erledigen wir durch manuelle Änderung. Direkt in den Templates. Diese werden durch Klicken auf das Suchergebnis geöffnet.

Folgender Regexp Suchbegriff

```
(<(img|input|hr|meta)[^>]+[^/])>
```

findet alle aufgeführten, nicht XHTML kompatibel geschlossenen Tags. Mit rechtem Mausklick auf die jeweilige Zeile im Suchergebnis können wir ein kontrolliertes Replace durchführen. Wichtig ist es, darauf zu achten, dass die zuvor geänderten Zeilen mit cfif ausgelassen werden!



Als nächstes suchen wir alle href, action und src Attribute, die & (ampersand) enthalten, aber kein &. Das bedeutet jetzt etwas mehr manuelle Arbeit.

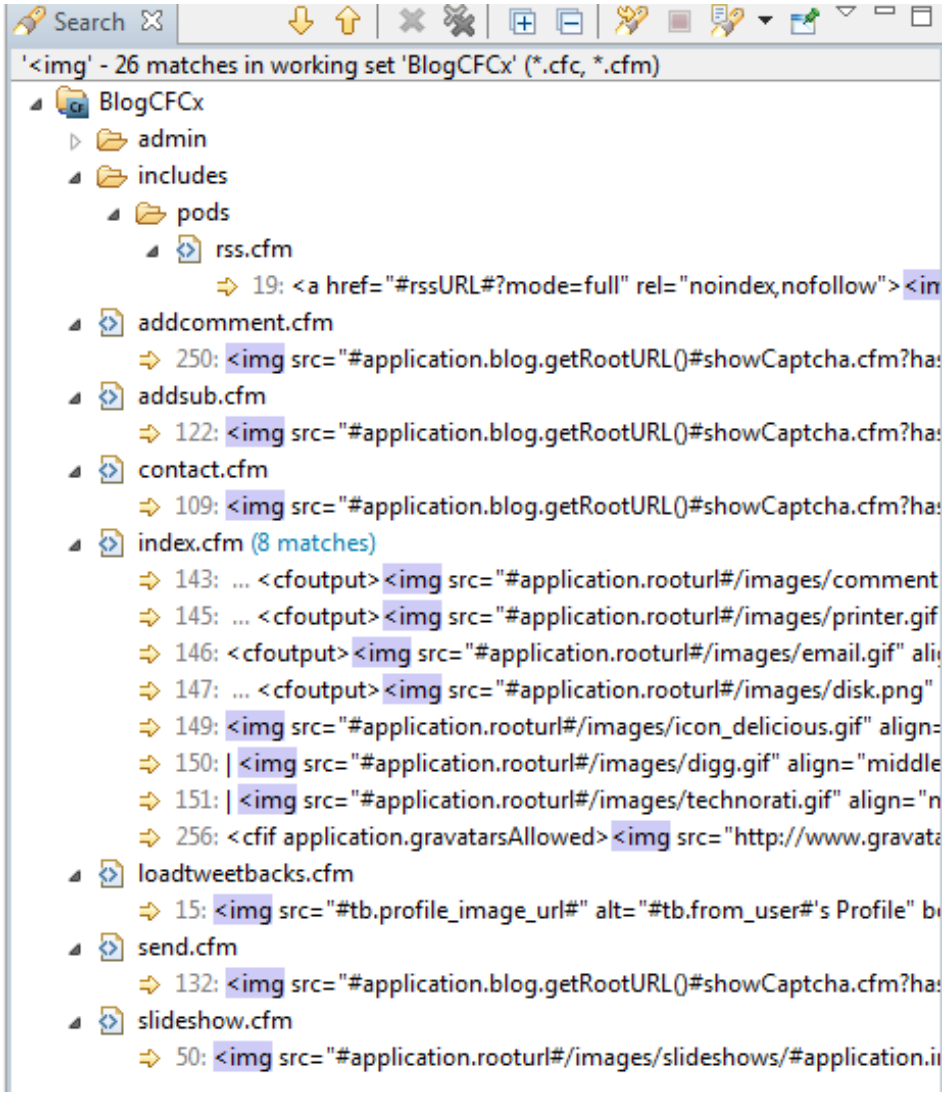
```
(href|action|src)="[^"]+&(?!amp;)+[^"]+"
```

Die Suchergebnisse sind rasch aufgelistet und wir springen direkt über das Suchergebnis zur Stelle im Code die wir ausbessern wollen und hängen an die fraglichen & jeweils noch ein amp; daran. Danach erhalten wir folgendes Ergebnis:

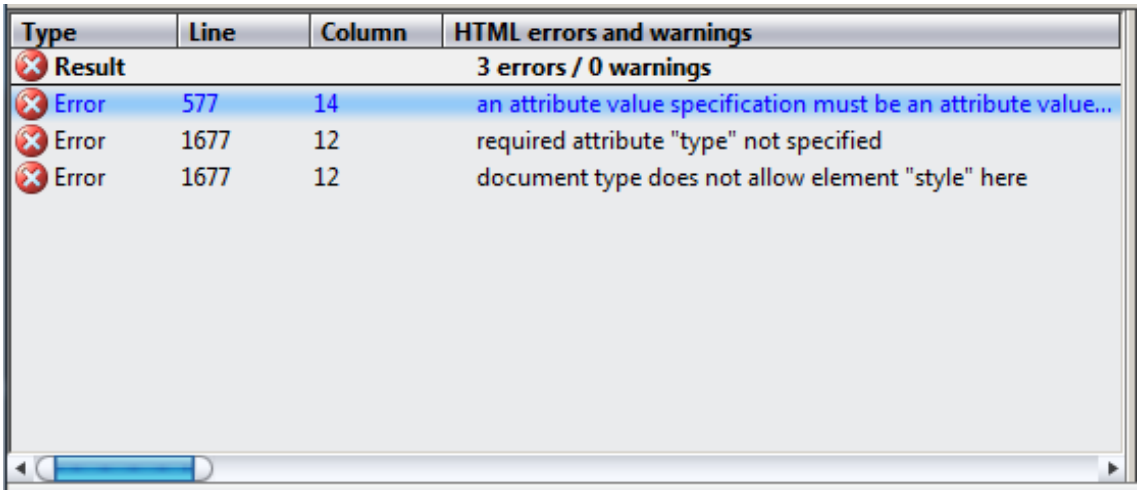
Type	Line	Column	HTML errors and warnings
Result			66 errors / 1 warning
Error	66	116	required attribute "alt" not specified
Error	66	324	required attribute "alt" not specified
Error	66	548	required attribute "alt" not specified
Error	67	126	required attribute "alt" not specified
Error	68	113	required attribute "alt" not specified
Error	69	131	required attribute "alt" not specified
Error	95	116	required attribute "alt" not specified
Error	95	351	required attribute "alt" not specified
Error	95	575	required attribute "alt" not specified
Error	96	126	required attribute "alt" not specified

Damit haben sich die Fehler bereits auf 66 reduziert. Der Validator beklagt sich nun noch über eine ganze Reihe von fehlenden alt Attributesn in img's und deutet noch darauf hin, dass es noch ein unerledigtes & für startRow gibt. Am Besten wir suchen jetzt einmal nach "startrow". 2 Treffer in der index.cfm. Dort wird eine URL für einen Link programmatisch zusammengestellt. Ausgebessert. Behoben. Weitere zwei Fehler verschwunden.

Jetzt zu den alt Attributen in den img Tags. Suchen wir mal alle img Tags und prüfen wir, ob "alt" Attribute vorhanden sind. Wenn nicht, dann fügen wir diese ein. Wo es möglich ist, kopieren wir ganz einfach den Inhalt eines vorhandenen title attributes.



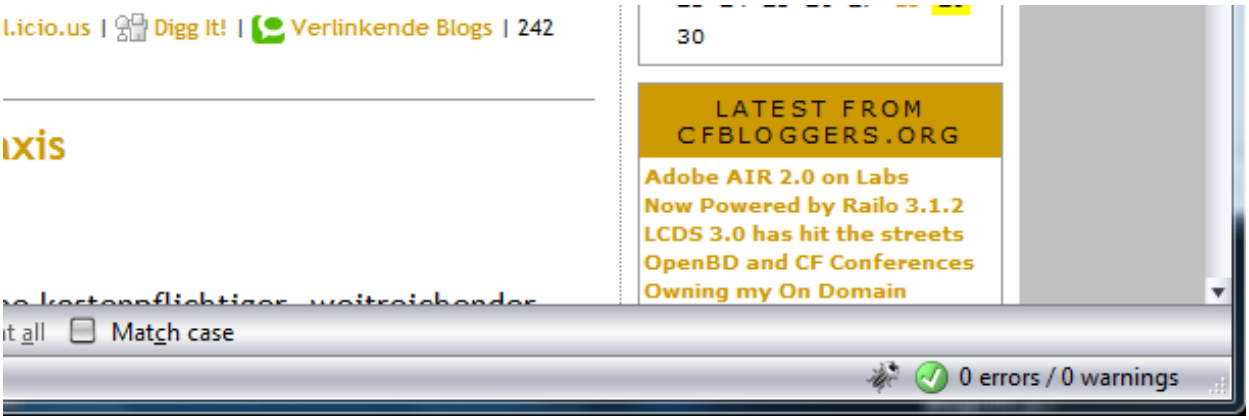
Damit sind wir auf der Einstiegsseite bei nur mehr 3 Fehlern angelangt. Wir sind nah dran.



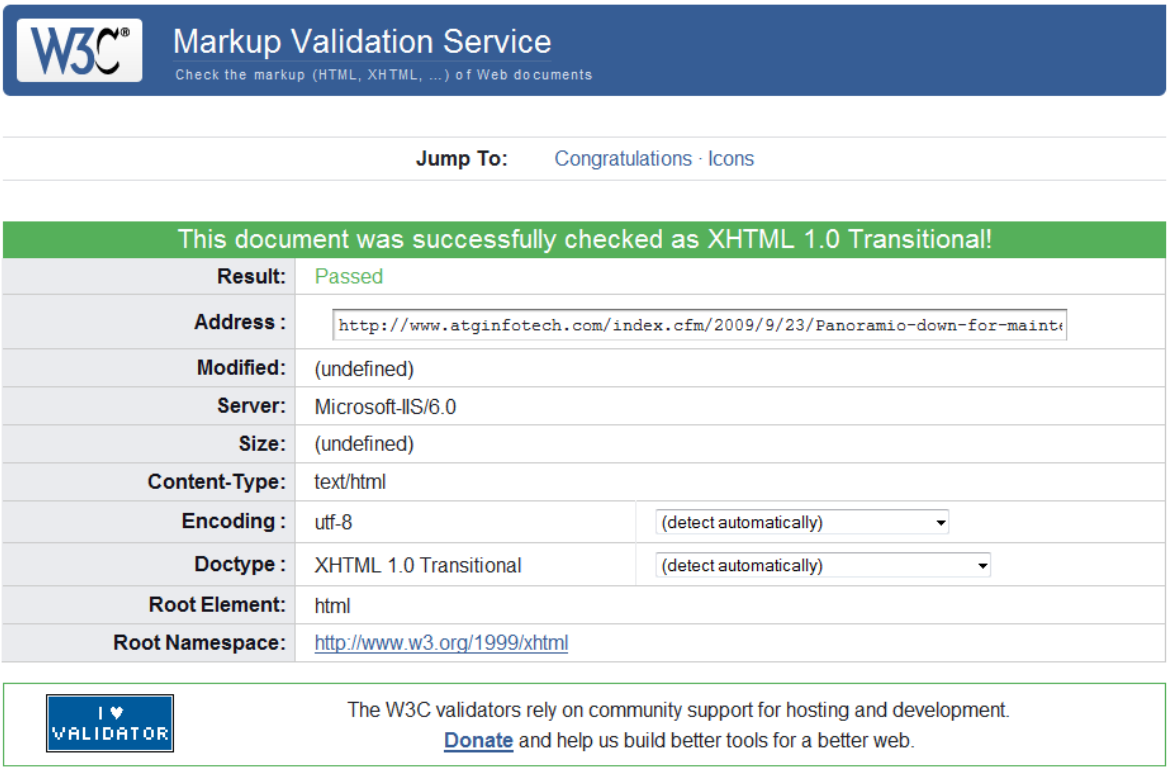
Type	Line	Column	HTML errors and warnings
Result			3 errors / 0 warnings
Error	577	14	an attribute value specification must be an attribute value...
Error	1677	12	required attribute "type" not specified
Error	1677	12	document type does not allow element "style" here

In einem table Tag im der calendar.cfm findet sich noch ein border=0. Dieses wird durch border="0" ersetzt. Der Validator mokiert sich über eine style Definition mitten im body der Seite. Diese Definition in search.cfm und tagcloud.cfm lagern wir in das Stylesheet aus.

Geschafft!



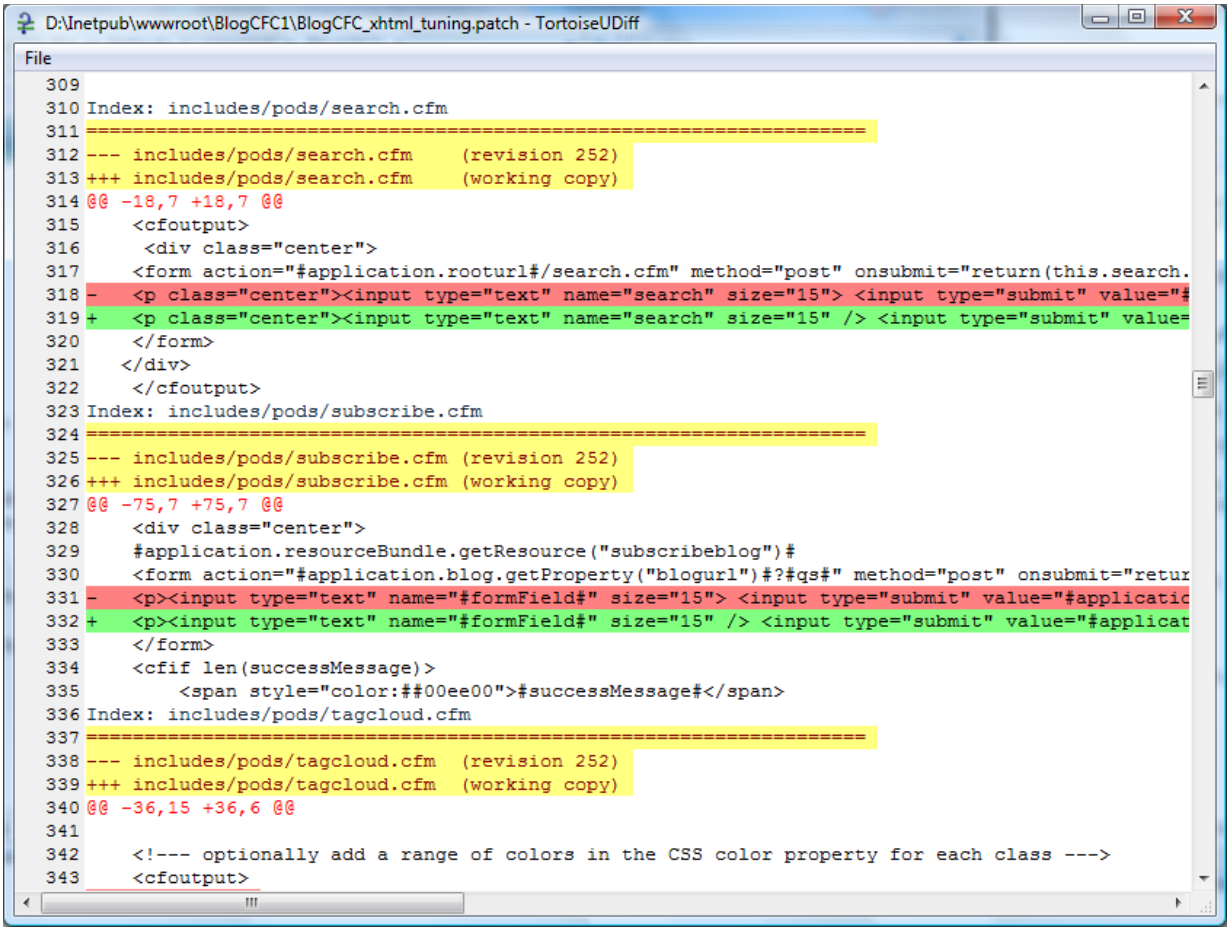
Jetzt noch mal die offizielle Validierung über http://validator.w3.org/ durchführen. Bestanden!



Wer ganze Arbeit leisten will untersucht auch noch die Formulare search.cfm, addcomments.cfm, addsub.cfm und passt sie standardkonform an.

jedes Auftreten von onClick umwandeln in Kleinschreibung.
jedes Auftreten von selected innerhalb von <option als selected="selected" darstellen.
jedes Auftreten von checked innerhalb von <input als checked="checked" darstellen.
textareas mit passenden col und row Attributen ergänzen.
<script> Tags mit type="text/javascript" ergänzen.
wer ganz genau sein will, stell noch sicher dass vor jedem /> ein Leerzeichen steht.

Zum Abschluss erstellen wir noch ein SVN Patch File, in dem wir die Änderungen in Ruhe auf Ihre Plausibilität prüfen können.



Für ganz Eilige haben wir den SVN Patch zum Download bereitgestellt (basierend auf SVN Rev.252): **BlogCFC_xhtml_tuning.patch** Dieser Patch ist nur zu Testzwecken und nicht für den Produktiveinsatz gedacht. Wir übernehmen keine Haftung für irgendwelche Schäden die nach Einspielen des Patches auftreten könnten und garantieren nicht die fehlerfreie Funktion.

Das Frontent besteht nun aus korrektem XHTML. Jetzt nur noch aufpassen, dass nicht im eingegebenen Content XHTML Fehler auftreten. So wie bei diesem Eintrag hier ;-), weil z.B.: das Colorcoding der Regexp Ausdrücke keinen optimalen Code erzeugt.