OC Entwicklung - Bug #121

Sonderzeichen in GPX-Dateien

05/05/2013 12:17 - following

Status:	erledigt	**% Done:**	100%
Priority name:	2 mittel	**Estimated time:**	0.00 hour
Assignee:	cmanie		
Target version:	Version 9		
Ticket Referenz:	4455 following 7/2012	**Kategorien:**	export

Description

Es gibt mehrere Hinweise darauf, dass Umlaute etc. in GPX-Dateien überflüssigerweise HTML-codiert werden, z.B. ö für ö, und dann in codierter Form in Garmin-Geräten erscheinen. Siehe z.B.

http://www.geoclub.de/viewtopic.php?f=126&t=56539
http://forum.geocaching-network.com/index.php?topic=2086.0

Umgekehrtes Problem bei Zeichen, die in XML-Daten codiert sein müssen (vgl. http://de.selfhtml.org/xml/regeln/zeichen.htm) - hier fehlt (teilweise?) die Codierung:

http://forum.geocaching-network.com/index.php?topic=2622

Related issues:

Related to OC Entwicklung - Bug #222: Falsches Hint-Format in GPX und TXT	**erledigt**

Associated revisions

Revision 57a74fab - 06/25/2013 18:53 - cmanie

Special Characters in GPX-/TXT-Files ; update #121 ; update #222

Revision 11b1cd18 - 06/25/2013 18:53 - cmanie

Special Characters in GPX-/TXT-Files ; update #121 ; update #222

Revision 2bc2d85f - 06/25/2013 19:34 - cmanie

Special Characters in GPX-/TXT-Files ; update #121 ; update #222

Revision 99e638e0 - 07/08/2013 17:05 - cmanie

Special Characters in GPX-/TXT-Files ; update #121 ; update #222

Revision c1a9efd6 - 07/08/2013 17:06 - cmanie

Special Characters in GPX-/TXT-Files ; update #121 ; update #222

Revision 55a9a7a5 - 07/08/2013 17:06 - cmanie

Special Characters in GPX-/TXT-Files ; update #121 ; update #222

Revision 5feac313 - 07/08/2013 17:06 - following

optimized GPX entity translation
and decoded "ö" in cache desc deletion message

update #121, update #222

Revision f5565a91 - 07/13/2013 12:38 - following

ported search.php & tools to lib2, and ...

- fixed logentry sorting for logs with identical date
- fixed ä etc. display in short descriptions on search results page
- try to select a matching language for the short descriptions
- added some translations
- hide download and map links if no search results (updates #235)
- nicer display of selectlocid page
- use site-dependent urls in GPX, LOC, TXT and KML
- unified XML encoding, now all done via two functions in lib2/util.inc.php (updates #121)
- some preventive XML encoding adjustments in GPX
- removed XML encoding in LOC CDATA section
- improved charset conversion for OVL and OV2 output
- some optimizations
- discarded lots of obsolete code
- disabled debug mode force_compile in OcSmarty class (performance)

History

#1 - 05/05/2013 12:18 - following

- Priority name changed from 0 keine to 2 mittel

Ist das noch aktuell / ist es überhaupt lösbar?

In search.gpx.inc.php gibt es eine Sonderzeichen-Codiertabelle, aber soweit ich mich erinnere wird im polnischen Code wesentlich mehr codiert ...?

#2 - 05/07/2013 13:50 - following

- Kategorien set to export

#3 - 05/27/2013 17:25 - cmanie

Ist noch aktuell.
Das Problem ist, das die Umlaute schon als HTML-Entities vorhanden sind. Das '&' wird dann noch mal zusätzlich als HTML-Entity umgewandelt.
Das daraus entstehende 'ö' ist natürlich für kein Gerät mehr lesbar.

#4 - 05/29/2013 09:55 - cmanie

- Assignee set to cmanie

#5 - 06/03/2013 16:15 - cmanie

- Status changed from offen to in Arbeit 20%

#6 - 06/03/2013 17:30 - cmanie

- Target version set to Version 9

#7 - 06/04/2013 11:37 - cmanie

- % Done changed from 0 to 50

Info:
Tatsächlich ist das ein Bug der Garmin Software für das etrex 30 (möglicherweise auch 10/20 und andere Geräte).

Mit der Version 3.0 für die etrex Serie ist dieser Bug behoben worden. Dafür sind das Kartenscrolling, der Gerätestart und andere Dinge extrem langsam geworden weswegen viele Nutzer noch eine ältere Version einsetzen.

Sidenote:
Die GC-PQs haben das gleiche Problem der Doppelcodierung.

#8 - 06/04/2013 20:25 - cmanie

- % Done changed from 50 to 100

Konvertierung der XML-/HTML-Entities geändert.
Statt Regex werden jetzt die PHP-Standardmethoden verwendet.

Um die Doppelcodierungen zu vermeiden werden evtl. bereits kodierte XML-/HTML-Entities vor der Umwandlung in ihre Original-Schriftzeichen zurückgewandelt und NICHT wieder in HTML-Entities zurückgewandelt. Auch vorher schon waren die Umlaute in den Log-Kommentaren nicht immer in XML-/HTML-Entites umgewandelt und wurden es beim Export auch nicht.

Da der Content-Type im XML als UTF-8 definiert ist und die Umlaute auch UTF-8 kodiert sind sollte das bei den Geräten keine Probleme geben. Wäre das ein Problem gewesen, wäre das auch schon in der Vergangenheit aufgefallen, was aber wohl nicht der Fall ist.

Auch in CacheWolf, MacCaching und Basecamp lassen sich die Dateien problemlos importieren.

Info:
Die gleiche Methode gibt es noch acht weitere Mal in gleicher oder ähnlicher Form.
Da dieser Bug hiermit erstmal behoben ist mache ich dafür ein neues Ticket auf.

Getestet mit Garmin etrex 30 / SW 2.60, 2.80, 3.00, BaseCamp, CacheWolf, MacCaching

#9 - 06/04/2013 20:26 - cmanie

- Status changed from in Arbeit 20% to im Test

#10 - 06/04/2013 20:26 - cmanie

- Status changed from im Test to in Arbeit 20%

#11 - 06/05/2013 01:37 - following

following schrieb:

> Umgekehrtes Problem bei Zeichen, die in XML-Daten codiert sein müssen

> (vgl. http://de.selfhtml.org/xml/regeln/zeichen.htm) - hier fehlt (teilweise?) die Codierung:
>
> http://forum.geocaching-network.com/index.php?topic=2622

Dieses Problem mit den Geokret-Namen ist mittlerweie behoben.

#12 - 06/05/2013 01:39 - following

Von allen GPX-Daten, die über die Funktion xmlentities() laufen, sind meines Wissens nur die Logtexte bereits vorcodiert; siehe dazu auch https://github.com/OpencachingDeutschland/oc-server3/blob/master/htdocs/okapi/services/logs/submit.php#LC176.

Die Decodierung sollte also nur auf die Logtexte angewendet werden; bei allen anderen Feldern kann sie Fehler produzieren.

Noch eine Formalie: Die OC-Quelltexte sind (mit wenigen Ausnahmen) mit Tabs eingerückt, wobei ein Tab = 2 Zeichen ist. Finde ich selbst einigermaßen nervig, aber solange das so einheitlich gehandhabt wird müssen wir der Lesbarkeit halber dabei bleiben.

#13 - 06/05/2013 01:48 - following

following schrieb:

> Von allen GPX-Daten, die über die Funktion xmlentities() laufen, sind meines Wissens nur die Logtexte bereits vorcodiert;

Oops, das gilt natürlich auch für die Cachebeschreibungen. Alles, was mit dem Wysywyg-Editor bearbeitet werden kann (Logtexte, Cachebeschreibungen, Benutzerprofil-Text) steht als HTML in der Datenbank (auch wenn man es als Plaintext editiert hat), alles andere uncodiert.

#14 - 06/05/2013 11:15 - cmanie

Die Cachebeschreibung ist vorcodiert, die Logs manchmal.
Anscheinend werden nur OKAPI-Logs vorcodiert. Bei Eintragung von Logs über die Oberfläche werden Umlaute nicht in Entities umgewandelt.

Ich kann zwar nicht nachvollziehen, was bei der Decodierung

`&ouml; -> ö`

für Fehler auftreten sollten, setze das aber gerne wunschgemäß um.

#15 - 06/05/2013 11:16 - cmanie

- % Done changed from 100 to 80

#16 - 06/05/2013 11:17 - cmanie

- Private changed from No to Yes

following schrieb:

> Noch eine Formalie: Die OC-Quelltexte sind (mit wenigen Ausnahmen) mit Tabs eingerückt, wobei ein Tab = 2 Zeichen ist. Finde ich selbst einigermaßen nervig, aber solange das so einheitlich gehandhabt wird müssen wir der Lesbarkeit halber dabei bleiben.

Tut mir leid. Ich nutze eine IDE. Da fallen solche "Nebensächlichkeiten" nicht auf ;-)
Werde ich einstellen.

#17 - 06/05/2013 11:17 - cmanie

- Private changed from Yes to No

#18 - 06/05/2013 12:30 - following

cmanie schrieb:

> Die Cachebeschreibung ist vorcodiert, die Logs manchmal.
> Anscheinend werden nur OKAPI-Logs vorcodiert. Bei Eintragung von Logs über die Oberfläche werden Umlaute nicht in Entities umgewandelt.

Alle Logtexte (und Beschreibungstexte) sind im HTML-Format gespeichert - also wenn der Benutzer Zeichen wie < oder & eingegeben hat, sind die als < und & codiert. Umlaute brauchen nicht codiert sein, weil es keine speziellen XML-Zeichen sind. Die Codierung macht je nach Editor-Modus entweder der TinyMce oder log.php/editlog.php/submit.php via htmlspecialchars(), daher kann es Unterschiede im Umfang der Codierung geben.

> Ich kann zwar nicht nachvollziehen, was bei der Decodierung [...] für Fehler auftreten sollten, setze das aber gerne wunschgemäß um.

Fehler würden auftreten, wenn z.B. ein Benutzer die Zeichenfolge ">" im Cachename, der Kurzbeschreibung etc. verwendet - die würde dann zu ">" verfälscht. Zugegeben ein sehr pathologischer Fall, aber wer weiß was es alles für Smilies oder Rätselcachedaten usw. gibt ;)

#19 - 06/05/2013 13:43 - cmanie

> Alle Logtexte (und Beschreibungstexte) sind im HTML-Format gespeichert [...]
> Umlaute brauchen nicht codiert sein

Hab mich falsch ausgedrückt.
Umlaute sind manchmal in Entities codiert, manchmal nicht.
In der Cachebeschreibung immer, im Log nur manchmal. OKAPI und lop.php nutzem die Methode htmlspecialchars(), die Entities nicht codiert. Die Cachedeschreibung wird durch den HTML-Purifier gejagt, vielleicht macht der das.
Zuerst hatte ich auch TinyMCE im Verdacht, der ist es aber nicht.

> [...] wenn z.B. ein Benutzer die Zeichenfolge ">" im Cachename, der Kurzbeschreibung etc. verwendet [...]

o.k., meinetwegen ;-)

#20 - 06/05/2013 16:10 - following

Beim Testen bin ich auf ein weiteres Problem gestoßen; hatte oben leider nicht weit genung gedacht.

Angenommen ein Benutzer editiert eine Cachebeschreibung im Wysywyg-Editor und gibt im Text ein: "Es muss <hier> gesucht werden." Dann wird das in der Datenbank als " Es muss <hier> gesucht werden. " gespeichert (mit dem Flag desc_html = 1).

html_entity_decode macht daraus: " Es muss <hier> gesucht werden. "
htmlspecialchars macht daraus: "<p>Es muss <hier> gesucht werden.</p>"
Beim Decodieren der XML-Daten wird daraus: " Es muss <hier> gesucht werden. "
... und angezeigt wird: "Es muss gesucht werden."

Demnach darf xmlentities() wohl nicht alles decodieren, sondern nur gezielt solche Zeichen die nicht zu den "htmlspecialchars" zählen - Umlaute etc.?

#21 - 06/05/2013 16:18 - cmanie

Hm, ja, das ist nicht schön.
Ich schaue mir das noch mal an.

Übrigens betraf dieser Bug, wenigstens auf meinem Garmin etrex, nicht nur Umlaute sondern auch Quotes und wahrscheinlich auch andere Entities.

Das werden wir dann nicht verhindern können.

#22 - 06/09/2013 02:24 - following

Die Hints sind überraschenderweise auch HTML-codiert, aber Umlaute sind wohl nicht betroffen. Dafür gibt es dort ein anderes Doppelcodierungsproblem -> #222.

Vielleicht stammen die codierten Umlaute aus von GC rüberkopierten HTML-Beschreibungen? In OC kann ich sie jedenfalls nicht reproduzieren.

#23 - 06/13/2013 19:24 - cmanie

- % Done changed from 80 to 100

Eigentlich war schon der Ansatz falsch.

Wenn HTML zur Anzeige übermittelt werden soll, dann ist das ganze in XML in **<![CDATA[ ]]>** Blocks einzufassen. Wenn man HTML escaped, dann kann man eigentlich nicht erwarten, dass das Zielgerät das vernünftig darstellt. Zudem kommt es zu Doppelcodierungen da beispielsweise ä kein valides XML-Entity ist.
CDATA-Blocks werden vom XML-Parser nicht behandelt und können daher selbst auch XML, eine Untermenge davon oder sogar Binärdaten enthalten.

Trotzdem hat beispielsweise der Garmin etrex 30 in der SW-Version bis 2.80 ein Problem mit HTML-Entities (Umlauten). Auch andere Tools wie Cache-Wolf können mit codierten Entities nicht umgehen. Daher werden jetzt alle HTML-Entities dekodiert und als UTF-8-Zeichen ins GPX geschrieben. Einzig spitze Klammern werden wieder kodiert, da der enthaltene Text sonst als Tag erkannt und ausgeblendet werden könnte (da nicht valide). Da das File-Encoding im XML-Header angegeben ist sollte das keine Schwierigkeiten machen.

Getestet mit Garmin etrex 30 (SW 2.60, 2.80, 3.00), CacheWolf, MacCaching und BaseCamp.

#24 - 06/14/2013 03:16 - following

Test positiv mit GSAK, aber negativ mit Garmin Mapsource: Die Datei ist nicht einlesbar (Gegentest mit der aktuellen Stable-Codeversion ist positiv). Bascamp kam Ende 2010 raus, d.h. bei vielen Leuten dürfte noch Mapsource laufen, u.a. bei mir.

Das Garmin Dakota 20 soll auch Probleme mit CDATA haben: http://forum.locusmap.eu/viewtopic.php?f=10&t=1967

Gefunden im Groundspeak-Forum, http://forums.groundspeak.com/GC/index.php?showtopic=286053, November 2011:

> Fixing this problem would require that both the short description and long description sections of the GPX file be changed to CDATA sections. This type of change to the file structure of the GPX file would cause issues on many devices since the CDATA section is not part of the specification of the GPX file format. At this time this is something we cannot fix.

Ich fürchte damit scheidet diese Lösung aus.

Man muss auch immer damit rechnen, dass irgendwelche Basteltools XML-Daten nicht per XML-Parser sondern sonstwie einlesen. Die orentieren sich dann am bislang üblichen GPX-Format der Geocaching-Websites und -Tools, die wohl alle kein CDATA erzeugen.

#25 - 06/14/2013 10:13 - cmanie

following schrieb:

> Test positiv mit GSAK, aber negativ mit Garmin Mapsource: Die Datei ist nicht einlesbar (Gegentest mit der aktuellen Stable-Codeversion ist positiv). Bascamp kam Ende 2010 raus, d.h. bei vielen Leuten dürfte noch Mapsource laufen, u.a. bei mir.
>
> Das Garmin Dakota 20 soll auch Probleme mit CDATA haben: http://forum.locusmap.eu/viewtopic.php?f=10&t=1967

Und ich hab mich gewundert, warum Garmin selbst in ihrem Exports keine CDATA-Blocks verwendet, da sie es ja als professionelles Unternehmen eigentlich besser wissen müssten. Jetzt vermute ich sie wissen es besser, nämlich das ihre (alte) Software keinen (ordentlichen) XML-Parser verwendet.

> Gefunden im Groundspeak-Forum, http://forums.groundspeak.com/GC/index.php?showtopic=286053, November 2011:
>
> > Fixing this problem would require that both the short description and long description sections of the GPX file be changed to CDATA sections. This type of change to the file structure of the GPX file would cause issues on many devices since the CDATA section is not part of the specification of the GPX file format. At this time this is something we cannot fix.

Die GPX-XSD und die Groundspeak-Cache-XSD sind Schemadefinitionen, die auf XML aufsetzen. CDATA-Blocks sind integraler Bestandteil von XML und müssen nicht definiert werden. Sie kommen immer dann zum Einsatz wenn Daten vom XML-Parser nicht verarbeitet werden sollen, wie beispielsweise eingebettetes XML/HTML.

#26 - 06/14/2013 11:58 - following

cmanie schrieb:

> Die GPX-XSD und die Groundspeak-Cache-XSD sind Schemadefinitionen, die auf XML aufsetzen. CDATA-Blocks sind integraler Bestandteil von XML und müssen nicht definiert werden. Sie kommen immer dann zum Einsatz wenn Daten vom XML-Parser nicht verarbeitet werden sollen, wie beispielsweise eingebettetes XML/HTML.

Klar. Diese Aussage macht sich halt besser als "would cause issues on many devices, because Garmin's XML parser is broken". :-/

#27 - 06/14/2013 15:00 - cmanie

following schrieb:

> Klar. Diese Aussage macht sich halt besser als "would cause issues on many devices, because Garmin's XML parser is broken". :-/

Was *sich besser macht* ist mir egal. Ich benötige Informationen die bei der Entwicklung hilfreich sind.

Die Aussage, das auf machen Garmin Geräten, in mancher (alten) Garmin Software und möglicherweise in Basteltools CDATA-Blocks nicht unterstützt werden ist auf jeden Fall hilfreicher als die Aussage *"since the CDATA section is not part of the specification of the GPX file format"*, die schlicht falsch ist. Genauso wie alles andere was der "Admin" in dem verlinkten Thread da so schreibt totaler Quatsch ist.

Wenn einige Garmin Geräte das nicht unterstützen ist das ein Fakt, den man leider nicht außer acht lassen kann. Vollkommen unerheblich wie die Spezifikationen aussehen.

#28 - 06/14/2013 20:33 - bohrsty

cmanie schrieb:

> Wenn einige Garmin Geräte das nicht unterstützen ist das ein Fakt, den man leider nicht außer acht lassen kann. Vollkommen unerheblich wie die Spezifikationen aussehen.

waere hier eine auswahl beim download fuer den user denkbar? eine "volle" version mit CDATA und und eine vollstaendig escapete, die anhand einer ersetzungstabelle alle nicht-xml- und nicht von den geraeten anzeigbare zeichen ersetzt (wie beim chatusername #209)...

#29 - 06/15/2013 00:32 - following

bohrsty schrieb:

> waere hier eine auswahl beim download fuer den user denkbar?

Halte ich in Sachen Usability für ungünstig. Je nach Software oder Gerät muss man die eine oder die andere GPX-Variante verwenden; heruntergeladene und gespeicherte GPX-Dateien sind dann nicht mehr universell verwendbar, und im Zweifelsfall passt es gerade nicht.

Eigentlich müsste man so gut wie alles problemlos decodieren können:

" und ' machen im Text kein Problem
> auch nicht
< nur dann, wenn es Beginn eines syntaktisch gültigen Tags ist
& nur dann, wenn es Beginn eines HTML-Entities ist

Da bleibt praktisch fast nichts Codiertes mehr übrig. Die Decodierung ist dann nur nicht mehr trivial sondern braucht etwas Syntax-Parsing.

#30 - 06/15/2013 02:34 - following

... wobei nicht auszuschließen ist, dass bei einem einfachen < im Text auch wieder irgendwelche kaputten Parser auf die Nase fallen. Hm.

#31 - 06/20/2013 23:09 - following

Die Funktion decodeEntities() in Commit d3d3ab133 funktioniert nicht, weil changePlaceholder() immer mit $inverse=true aufgerufen wird. Außerdem ist noch ein Tippfehler drin: "$placeholter".

Müsste & nicht auch noch gesondert berücksichtigt werden?

Ich denke bei den Logtexten in search.txt.inc.php könnte man auch noch die HTML-Entities decodieren - ist ja das gleiche Format wie bei den Cachebeschreibungen.

#32 - 06/21/2013 10:31 - cmanie

following schrieb:

> Die Funktion decodeEntities() in Commit d3d3ab133 funktioniert nicht, weil changePlaceholder() immer mit $inverse=true aufgerufen wird. Außerdem ist noch ein Tippfehler drin: "$placeholter".

Shit. Wenn es zu spät wird...

> Müsste & nicht auch noch gesondert berücksichtigt werden?

Nein, weil:

& -decode-> & -htmlspecialchars-> &

& ist ein valides XML-Entity und muss nicht doppelt codiert werden (Wie auch?).

> Ich denke bei den Logtexten in search.txt.inc.php könnte man auch noch die HTML-Entities decodieren - ist ja das gleiche Format wie bei den Cachebeschreibungen.

Dachte ich auch. Deswegen hab ich das Decoding auch in die Methode html2text aufgenommen. Ich hab jetzt zusätzlich das Decoding für Logtexte aktiviert, bei denen das HTML-Flag nicht gesetzt ist. Zumindest in meinen Testdaten gab es Einträge die HTML-Tags und Entites enthielten aber das HTML-Flag nicht gesetzt hatten.

#33 - 06/21/2013 14:41 - following

cmanie schrieb:

> Ich baue es gerade selbst mit ein, der Rest passt ja.

Ich sehe gerade, dass bei der Hint-Codierung auch noch ein Bug drin ist. Dann ist es doch besser wenn du alles selbst korrigierst.

#34 - 06/21/2013 15:10 - following

Mist, nun hab ich irgendwie meinen Bugreport von eben überschrieben. Nochmal in Kurzfassung:

Eingabe im Editor: "ü", z.B. als Teil eines Rätsels
gespeichert als: "ü"
....
in der GPX-Ausgabe: "ü"
nach XML-Parsing: "ü"
Anzeige: "ü"

#35 - 06/25/2013 19:30 - cmanie

following schrieb:

> Eingabe im Editor: "ü", z.B. als Teil eines Rätsels

Nehmen wir mal an, es wird dann ü in das GPX geschrieben, so wie bisher.
Diese Eingabe wird aber dann zum Beispiel in
- MacCaching als ü
- CacheWolf als ? im HTML und ü im Textmodus
angezeigt.
Im Garmin und in Basecamp ist die Anzeige korrekt (ü).

MacCaching und CacheWolf egal?

#36 - 06/26/2013 14:08 - following

Hmm, verstehe. Wir müssen zwischen zwei Übeln abwägen und uns für eines entscheiden.

Mein Beispiel ist sehr konstruiert, sowas dürfte nur extrem selten vorkommen - und in der Regel wird es dann dem Owner auch auffallen, sodass er die Beschreibung umformulieren kann. Daher ist deine Lösung sicher günstiger.

Notfalls könnte man immer noch einen aufwändigeren Workaround nachliefern, z.B. solche Pseudo-Entities im Content erkennen und ein Leerzeichen einfügen.

#37 - 06/27/2013 22:59 - following

Huch, du hast ja nun doch eine Sonderbehandlung von & eingebaut, also es steht dann doppelt codiert im XML und wird vom XML-Parser zu & decodiert. War das Absicht?

Ansonsten sollte nun alles passen.

#38 - 06/28/2013 09:40 - cmanie

following schrieb:

> Huch, du hast ja nun doch eine Sonderbehandlung von & eingebaut, also es steht dann doppelt codiert im XML und wird vom XML-Parser zu & decodiert. War das Absicht?

Ja, um den von dir skizzierten Fall abzubilden. Der Einwand ist ja durchaus berechtigt.
Da es in diesem speziellen Fall dem jetzigen Verhalten entspricht und wohl auch nicht so häufig vorkommt, macht es jedoch nicht mehr kaputt als vorher. Die Anzeige im Garmin und Basecamp und teilweise in CacheWolf ist ja korrekt.

Wir können uns dann immer noch entscheiden, ob wir das "wieder ausbauen" und dann in Kauf nehmen das statt dem gewünschten "ü" ein "ü" angezeigt wird. Dazu muss nur eine Zeile entfernt werden.

#39 - 07/08/2013 17:13 - following

- Status changed from in Arbeit 20% to erledigt

Ist online.

Was mir noch aufgefallen ist: Codierte Umlaute gibt es nur in Cachebeschreibungen bis date_created 10.06.2011. Anscheinend wurde an diesem Tag etwas am HTML-Purifier geändert.