

Universität des Saarlandes
Fachrichtung 6.2 Informatik

Zufallstesten von Programmen, Diensten und
Handys

WMLGEN

Handbuch

Phase	Phasenverantwortlicher	E-Mail
Pflichtenheft	Stephan Schlicker	stephanschlicker@web.de
Grobentwurf	Holger Hewener	holger@hewener.net
Spezifikation	Andreas Schlicker	andreasschlicker@web.de
Implementierung	Dominik Gummel	bofh@krypt1.cs.uni-sb.de
Testbericht	Sascha Kiefer	sk@intertivity.com
Demonstration	Markus Uhl	markus.uhl@gmx.de

23. Juli 2003



Inhaltsverzeichnis

1	Einführung	3
2	Installation	4
2.1	Systemvoraussetzungen	4
2.1.1	Minimalanforderungen	4
2.1.2	Anforderungen zum manuellen Validieren und Testen	4
2.1.3	Anforderungen zum automatischen Testen	4
2.1.4	Anforderungen zum Testen von Endgeräten	4
2.1.5	Anforderungen zum Compilieren des Quellcodes von WMLGEN	4
2.2	Installation	5
2.3	WMLGEN starten	5
3	Bedienung	6
3.1	Benutzen der Standardkonfiguration	6
3.1.1	Ausgabe auf den Bildschirm	6
3.1.2	Ausgabe in eine Datei	6
3.2	Benutzen einer eigenen Konfiguration	7
3.2.1	Eigenschaften der Konfigurationsdatei	7
3.2.2	Ausgabe auf den Bildschirm	9
3.2.3	Ausgabe in einer eigenen Datei	9
3.3	Benutzen einer eigenen Konfiguration und Seed	9
3.4	Validieren der erzeugten Dokumente	9
3.5	Testen der erzeugten Dokumente	10
3.5.1	Lokales, manuelles Testen	10
3.5.2	Automatisiertes Testen mit der WMLGEN-Testdatenbank	10
3.5.3	Testen von Endgeräten	10
4	Tutorial	11
4.1	Erzeugen und Speichern eines WML-Dokumentes mit der Standardkonfiguration	11
4.2	Erzeugen und Speichern eines WML-Dokumentes mit eigener Konfiguration	11
4.3	Erzeugen und Speichern eines WML-Dokumentes mit eigener Konfiguration und Seed	12
5	Probleme und Fehlermeldungen	13
5.1	Fehlermeldungen bei der Erzeugung der Dokumente:	13
5.2	Probleme mit den Zeichenkodierungen	13
A	Abkürzungsverzeichnis	14
B	Glossar	15

1 Einführung

Ein verbreitetes Verfahren, Fehler in Programmen aufzuspüren, ist das Erzeugen zufälliger Eingaben. WMLGEN ist ein Programm, das ein solches Verfahren zum Testen von Programmen, Diensten und Handys für WML-Dokumente implementiert.

Sie können mit WMLGEN zufällige, syntaktisch korrekte¹ WML-Dokumente generieren und diese einem Programm zur Weiterverarbeitung schicken, natürlich mit dem Ziel, das Programm möglichst zu einem Fehlverhalten zu bringen.

WMLGEN richtet sich an alle erfahrenden Computerbenutzer, ob es nun Handyhersteller, Handy-Netzbetreiber, Entwickler von WAP-Browsern/-Emulatoren, Tester von Softwaresicherheit oder einfach nur interessierte Handynutzer sind, die den WAP-Browser ihres Handys testen wollen.

Sie können entweder eine voreingestellte Konfiguration zur Erzeugung der WML-Dokumente benutzen oder in einer Konfigurationsdatei selbst festlegen, welche Eigenschaften die WML-Dokumente besitzen sollen. Die in der Konfigurationsdatei festgelegten Eigenschaften bilden den Rahmen für die durch einen Zufallsgenerator ansonsten beliebig erzeugten Dokumente.

Sie können sich die Dokumente auf den Bildschirm ausgeben lassen oder in einer Datei speichern .

Zusätzlich zum erzeugten WML-Dokument speichert WMLGEN in jeweils eigenen Dateien die zur Erzeugung benutzte Konfiguration und die zugrundeliegende Seed des Zufallsgenerators. Wird eine Konfigurationsdatei verwendet, zu der bereits ein WML-Dokument durch WMLGEN generiert und die Seed des Zufallsgenerators gespeichert wurde, so wird automatisch diese Seed verwendet und ein identisches Dokument erzeugt. Existiert zu einer Konfigurationsdatei jedoch keine Seed-Datei , so wird automatisch eine neue Seed berechnet und ein neues zufälliges Dokument im Rahmen der angegebenen Konfiguration erstellt.

¹gemäß der WML-Spezifikation 1.3 vom 19. Februar 2000

2 Installation

In diesem Kapitel werden die Hard- und Softwarevoraussetzungen, die zu einem problemlosen Einsatz von WMLGEN benötigt werden, aufgezeigt. Darüber hinaus erhalten Sie Hinweise zur Installation und zum Start des Programms.

2.1 Systemvoraussetzungen

2.1.1 Minimalanforderungen

- Standard Desktop PC oder einen Laptop
- 5 MB freien Festplattenspeicher
- zusätzlichen Festplattenspeicher zum Speichern von erzeugten WML-Dokumenten
- Betriebssystem Windows 98 oder neuere Versionen (Zusätzlich benötigen Sie die Programmbibliothek `cygwin1.dll`, welche im Cygwin-Paket (www.cygwin.com) enthalten ist) bzw. Linux (ab Kernel 2.2)

2.1.2 Anforderungen zum manuellen Validieren und Testen

- XML-Validator (z.B. `xmlvalid` von ElCel Technology (www.elcel.com/products/xmlvalid.html))
- WML-Browser (z.B. Opera (www.opera.com), wApua (fsinfo.cs.uni-sb.de/~abe/wApua/), WINWAP (www.winwap.org))

2.1.3 Anforderungen zum automatischen Testen

- MySQL-Datenbank
- Apache-Webserver
- Perl

2.1.4 Anforderungen zum Testen von Endgeräten

- Server-Computer mit Internetzugang
- Webserver, der das Abrufen entsprechender WML-Dokumente unterstützt (z.B. Apache)
- Endgerät (z.B. Handy)

2.1.5 Anforderungen zum Compilieren des Quellcodes von WMLGEN

- C++ Compiler, gcc 3.2 oder kompatible Versionen

2.2 Installation

Sie erhalten WMLGEN unter folgender Adresse:

<http://alan.cs.uni-sb.de/wml/index.html>

Dort haben Sie die Wahl zwischen zwei Dateien: Einem komprimierten Tar-Archiv (für Linux-Betriebssysteme) und einem Zip-Archiv (für Windows-Betriebssysteme). Entscheiden Sie sich bitte für die Datei, die zu Ihrem Betriebssystem passt.

Nachdem Sie WMLGEN heruntergeladen haben, müssen Sie das Archiv entpacken. Dazu benötigen Sie ein geeignetes Programm, wie zum Beispiel tar und gzip (Linux) oder WinZip (Windows 98 oder neuere Versionen). Zum Installieren entpacken Sie bitte die heruntergeladene Datei in das gewünschte Verzeichnis. Dort wird ein Unterverzeichnis `wmlgen` erstellt, in das die WMLGEN-Dateien entpackt werden. Sie erhalten den Quellcode von WMLGEN und eine für Ihr Betriebssystem compilierte Version.

2.3 WMLGEN starten

Um WMLGEN zu starten, wechseln Sie in das Verzeichnis, in dem WMLGEN installiert ist und geben `wmlgen` ein. Dadurch wird das Programm mit der Standardkonfiguration gestartet und gibt ein zufällig erzeugtes WML-Dokument auf dem Bildschirm aus.

Wenn Sie WMLGEN unter Windows (ab Version Windows 98) benutzen möchten, benötigen Sie zusätzlich die Programmbibliothek `cygwin1.dll`, welche im Cygwin-Paket (*www.cygwin.com*) enthalten ist. Zum Starten von WMLGEN geben Sie in der Konsole `wmlgen` ein.

3 Bedienung

Sie haben mehrere Möglichkeiten WMLGEN zu benutzen. Sie können WMLGEN entweder mit der Standardkonfiguration starten, eine eigene Konfiguration benutzen oder zusätzlich eine Seed für den Zufallsgenerator vorgeben. Darüber hinaus können Sie sich das erzeugte WML-Dokument auf dem Bildschirm anzeigen lassen oder in einer Datei speichern.

Die Syntax zum Starten von WMLGEN lautet folgendermaßen:

```
wmlgen [--help] [config file] [output filename] [--debug]
```

[config file] gibt eine Konfigurationsdatei an, die zum Generieren des WML-Dokuments benutzt wird. Wenn Sie *zusätzlich* einen [output filename] angeben, dann wird das erzeugte WML-Dokument in einer Datei, die mit output filename beginnt, gespeichert. Zusätzliche Debug-Informationen während des Programmablauf erhalten Sie, wenn Sie als letztes Argument --debug eingeben. Starten Sie WMLGEN mit dem Parameter --help so erhalten Sie Hinweise zum Start des Programms sowie zur Konfigurationsdatei.

Im Folgenden werden die unterschiedlichen Möglichkeiten, WMLGEN zu starten, beschrieben.

3.1 Benutzen der Standardkonfiguration

3.1.1 Ausgabe auf den Bildschirm

Um WMLGEN mit der Standardkonfiguration zu starten, müssen Sie lediglich

```
wmlgen
```

im Installationsverzeichnis eingeben. Die Ausgabe der erzeugten Datei erfolgt auf dem Bildschirm.

3.1.2 Ausgabe in eine Datei

Wenn Sie WMLGEN mit der Standardkonfiguration nutzen und das erzeugte WML-Dokument in einer Datei speichern möchten, die mit `meinedatei` beginnen soll, geben Sie im Installationsverzeichnis folgendes ein:

```
wmlgen default.cfg meinedatei
```

WMLGEN erzeugt drei Dateien, die mit dem von Ihnen gewünschten Namen `meinedatei` beginnen:

- `meinedatei_timestamp.wml`
- `meinedatei_timestamp.cfg`
- `meinedatei_timestamp.seed`

WMLGEN hängt an den von Ihnen angegebenen Namen einen sog. `timestamp` an, der den Zeitpunkt zu dem die Ausgabe erstellt wurde angibt. Dadurch werden alle erzeugten Dokumente eindeutig unterscheidbar.

Die `name_timestamp.wml`-Datei ist das von WMLGEN erzeugte WML-Dokument.

Dieses können Sie nun in einem Programm Ihrer Wahl validieren oder testen (siehe auch Abschnitte 3.4 oder 3.5). Zusätzlich speichert WMLGEN in der Datei `name_timestamp.cfg` die von Ihnen zur Erstellung des WML-Dokuments benutzte Konfiguration. Darüber hinaus wird in `name_timestamp.seed` die Seed des Zufallsgenerators, die zur Erstellung des WML-Dokumentes genutzt wurde, abgelegt. Ein ausführlich erläutertes Beispiel finden Sie in Kapitel 4.1.

3.2 Benutzen einer eigenen Konfiguration

Neben der Standardkonfiguration in der Datei `default.cfg` können Sie auch selbst Vorgaben zur Erzeugung der WML-Dokumente machen. Diese legen Sie in einer eigenen Konfigurationsdatei, z.B. `meineconfig.cfg`, ab.

3.2.1 Eigenschaften der Konfigurationsdatei

Die Konfigurationsdatei ist in Sektionen unterteilt. Diese werden durch eckige Klammern gekennzeichnet und tragen die Namen:

- `[global]`: für die allgemeinen Einstellungen
- `[dtd]`: für die Dokumentstruktur
- `[direct_parents]`: für direkte Kindregeln
- `[further_parents]`: für entfernte Kindregeln

Die einzelnen Sektionen werden im Folgenden näher erläutert:

An erster Stelle in der Konfigurationsdatei befindet sich ein Abschnitt zur Festlegung von *allgemeinen Eigenschaften*. Dieser wird als `[global]` bezeichnet.

Hier ist es möglich folgende Einstellungen festzulegen:

- `minlengthfile`: Anzahl der minimalen Bytes (Zeichen), welche die Ausgabe umfassen soll. Diese Angabe wird als Richtmaß für die Ausgabe genutzt. So ist die Ausgabe meist nicht mehr als 20% länger als die angegebene minimale Dateilänge.
- `minlengthstring`: Minimale Länge für Textabschnitte im Dokument.
- `encoding`: Wahl der Zeichenkodierung. Auswahl: `US-ASCII`, `UTF-8`, `UCS-4`, `ISO-8859-1 (Latin-1)`.
- `averagelengthstring`: Durchschnittslänge der Textabschnitte gefolgt von der maximalen Abweichung. Wird für jeden Textabschnitt neu berechnet.
- `starttag`: Das Element, welches als Initialelement genutzt wird.
- `autorefresh`: Diese Einstellungsmöglichkeit ist hauptsächlich zum automatischen Testen der Dokumente gedacht, da hiermit in dem WML-Ausgabedokument eine Weiterleitung nach einer Sekunde auf eine konfigurierbare URL eingerichtet wird.
- `refreshurl`: URL, welche zum automatischen Neuladen für den automatischen Test genutzt werden soll.

Die *Struktur* des zu erzeugenden WML-Dokuments wird im Abschnitt [dtd] spezifiziert. Die Strukturdefinition ist an eine *Document Type Definition* (DTD) à la XML angelehnt.

Für die Konvertierung einer regulären XML-DTD in das von WMLGEN verwendete Konfigurationsformat kann ein Programm geschrieben werden. So wäre es möglich, den Generator von WMLGEN zur Erzeugung beliebiger XML-Dokumente zu nutzen. Für jedes Element kann man ...

- den Namen festlegen, der in der Ausgabe erscheinen soll (*name*)
- die Namen der Elemente festlegen, welche als Kinder eines Elements auftreten können (*children*)
- die Namen und den Typ der Attribute festlegen, welche zu einem Element zwingend angegeben werden müssen (*requiredattributes*)
- die Namen und den Typ der Attribute festlegen, welche zu einem Element optional angegeben werden können (*impliedattributes*)
- die Regel angeben, nach der die möglichen Kinder des Elements erstellt werden können (*rule*). Sie Syntax der Regeln ähnelt der von regulären Ausdrücken.
- angeben, ob ein Element leer ist, d.h. nach XML-Spezifikation keine Kinder hat und deswegen auf eine andere Weise ausgegeben wird (*empty*)
- die Wahrscheinlichkeit bestimmen, mit der ein Element erstellt wird, wenn es ausgewählt wird.

Die mitgelieferte Konfigurationsdatei `default.cfg` enthält alle Elemente gemäß der WML-Spezifikation 1.3.

Um Regeln zu beschreiben, welche die Beziehungen zwischen einem aktuellen Element und seinen *Vorgängern* zu beschreiben, gibt es ...

- einen Teil namens [*direct_parents*], welcher die Regeln für *direkte Nachfolger* beschreibt. Dazu gibt man den Namen des übergeordneten Elementes an gefolgt von Kinderelementen, für die Veränderungen der Erzeugungswahrscheinlichkeit angegeben werden
- einen Teil namens [*further_parents*], welcher die Regeln für *entfernte Vorgänger-Kind Beziehungen* beschreibt. Als erstes gibt man das Kindelement an. Daraufhin kann man verschiedene Elemente mit Wahrscheinlichkeiten angeben, welche für das aktuelle Element übernommen werden, wenn dieses Element als Vorgänger irgendwann aufgetreten ist. Wenn mehrere Elemente in dieser Auflistung als Vorgänger auftreten, so wird die kleinste angegebene Wahrscheinlichkeit für das aktuelle Element übernommen.

Um Werte festzulegen, gibt man die oben beschriebenen Eigenschaften mit Wertzuweisungen an. Beispielweise `name = wml`.

In der Sektion [*global*] werden die Werteingaben direkt nach der Sektionsdeklaration angegeben. In den anderen Sektionen muss zuerst eine Sub-Sektion angegeben werden, welche durch spitze Klammern beschrieben wird. Beispielsweise beschreibt `<wml>` die Sub-Sektion WML. Die dann folgenden Wertdeklarationen werden dann immer auf die letzte eingeleitete Subsektion bezogen.

Wenn Sie für eine Eigenschaft keinen Wert definieren wollen, so lassen Sie diese Deklaration einfach aus und entfernen Sie diese komplett aus der Konfigurationsdatei .

3.2.2 Ausgabe auf den Bildschirm

Angenommen Ihre Konfigurationsdatei heißt `meineconfig.cfg`. Sie starten WMLGEN dann mit folgendem Befehl:

```
wmlgen meineconfig.cfg
```

Das erzeugte WML-Dokument wird auf dem Bildschirm ausgegeben.

3.2.3 Ausgabe in einer eigenen Datei

Angenommen Ihre Konfigurationsdatei heißt `meineconfig.cfg` und die Datei, in der Sie das erzeugte WML-Dokument speichern möchten, soll mit `meinedatei` beginnen. Dann starten Sie WMLGEN mit folgendem Befehl:

```
wmlgen meineconfig.cfg meinedatei
```

Existiert zur Konfigurationsdatei `meineconfig.cfg` eine Seed-Datei `meineconfig.seed`, so verwendet der Zufallsgenerator von WMLGEN diese Seed zur Generierung des neuen WML-Dokumentes. Wurde die Konfigurationsdatei `meineconfig.cfg` bereits durch WMLGEN bei der früheren Erzeugung eines WML-Dokumentes gespeichert, so ist das nun generierte WML-Dokument genau identisch mit dem bereits existierenden WML-Dokument.

WMLGEN wird die drei folgenden Dateien erzeugen (siehe auch (3.1.2)):

- `meinedatei_timestamp.wml`
- `meinedatei_timestamp.cfg`
- `meinedatei_timestamp.seed`

Ein ausführliches Beispiel hierzu finden Sie in Kapitel 4.2.

3.3 Benutzen einer eigenen Konfiguration und Seed

Wenn Sie WMLGEN mit einer eigenen Konfiguration und einer vorgegebenen Seed starten möchten, gehen Sie genauso vor wie in Abschnitt 3.2 beschrieben. Erstellen Sie sich Ihre Konfigurationsdatei und speichern Sie die Seed in einer `.seed`-Datei unter identischem Namen. WMLGEN benutzt automatisch die in der `.seed`-Datei gespeicherte Seed zur Generierung des neuen WML-Dokuments.

In Kapitel 4.3 wird die Vorgehensweise für diese Benutzungsmöglichkeit in einem Beispiel beschrieben.

3.4 Validieren der erzeugten Dokumente

Wir empfehlen Ihnen zur Validierung der erzeugten WML-Dokumente den XML-Validator `xmlvalid` von ElCel Technology. Er ist erhältlich unter www.elcel.com/products/xmlvalid.html. Sie müssen sich zuerst registrieren, bevor Sie kostenlos den Validator runterladen können. Für alle gängigen Betriebssysteme ist eine Version verfügbar.

3.5 Testen der erzeugten Dokumente

3.5.1 Lokales, manuelles Testen

Zum manuellen, lokalen Anzeigen und Testen der erzeugten WML-Dokumente eignen sich unserer Erfahrung nach insbesondere der WAP-fähige Browser Opera (www.opera.com) und der WAP-Browser WINWAP (www.winwap.org). Installieren Sie einfach einen WAP-Browser Ihrer Wahl und schon können Sie mit dem Testen beginnen.

3.5.2 Automatisiertes Testen mit der WMLGEN-Testdatenbank

Zusätzlich zum manuellen Testen, können Sie die erzeugten WML-Dokumente auch automatisch testen. Dazu bieten wir Ihnen die WMLGEN-Testdatenbank an. Sie basiert auf einer MySQL-Datenbank und speichert alle WML-Dokumente, bei denen der Test nicht erfolgreich war, also der getestete Browser ein Fehlverhalten gezeigt hat. Die WMLGEN-Testdatenbank finden Sie im Verzeichnis `/wmlgen/testdb/`. Um sie zu benutzen, benötigen Sie allerdings zusätzliche Programme:

- Apache-Websserver
- Perl
- MySQL

Eine vorkonfigurierte Version der Datenbank existiert bereits. Benutzer, Passwörter etc. können im Perl-Script konfiguriert werden. Ein genaue Systembeschreibung, eine Bedienungsanleitung und Benutzungsbeispiele finden Sie im *Handbuch für die WMLGEN-Testdatenbank* und in einschlägiger Literatur.

3.5.3 Testen von Endgeräten

Um Endgeräte (z.B. Handys) zu testen, müssen Sie lediglich im Microbrowser des Endgerätes folgende Adresse aufrufen:

<http://alan.cs.uni-sb.de/wml/cgi-bin/mysqlwmlgen.pl>

4 Tutorial

4.1 Erzeugen und Speichern eines WML-Dokumentes mit der Standardkonfiguration

Sie möchten ein WML-Dokument mithilfe der Standardkonfiguration erzeugen und es speichern. Dabei soll der Dateiname mit `test1` beginnen.

1. Geben Sie in der Befehlszeile `wmlgen default.cfg test1` ein.
2. WMLGEN erzeugt die drei in Abschnitt 3.2.3 erläuterten Dateien.

4.2 Erzeugen und Speichern eines WML-Dokumentes mit eigener Konfiguration

Sie möchten eine eigene Konfigurationsdatei `meineconfig.cfg` zur Erzeugung eines WML-Dokumentes erstellen. Das erzeugte WML-Dokument möchten Sie, dabei soll der Dateiname mit `test2` beginnen.

1. Legen Sie mit einem Editor Ihrer Wahl eine leere Datei mit dem Namen `meineconfig.cfg` an.
2. Öffnen Sie mit einem Editor Ihrer Wahl die Standardkonfigurationsdatei `default.cfg` im `wmlgen`-Verzeichnis.
3. Kopieren Sie den Inhalt von `default.cfg` in `meineconfig.cfg`.
4. Schließen Sie `default.cfg` und nehmen Sie an `meineconfig.cfg` die von Ihnen gewünschten Änderungen vor (Lesen Sie hierzu auch Kapitel 3.2).
5. Angenommen, Sie möchten die folgenden fünf Änderungen im Vergleich zur Standardkonfiguration vornehmen:
 - Die Zeichenkodierung soll in US-ASCII erfolgen
 - Die durchschnittliche Länge eines Textabschnittes soll 13 sein, die maximale Abweichung soll 7 betragen.
 - Die Wahrscheinlichkeit für ein Absatz-Tag `<p>` soll 90% betragen.
 - Die Wahrscheinlichkeit, dass auf ein Absatz-Tag `<p>` ein Bild `` folgt, soll 50% betragen.
 - In einem betonten Textabschnitt `` sollen keine Bilder `` erscheinen.
6. Setzen Sie in der Sektion `[global]` den Wert für `encoding` auf `us-ascii`.
7. Geben Sie ebenfalls in der Sektion `[global]` hinter `averagelengthstring=13, 7` ein.
8. In der Sektion `[dtd]` setzen Sie in der Sub-Sektion `<p>` den Wert für `prob` auf `0.9`.
9. Setzen Sie in der Sektion `[direct_parents]` den Wert für `img` unter `<p>` auf `0.5`.

10. Setzen Sie in der Sektion [`further_parents`] den Wert für `img` unter `` auf 0.
11. Speichern Sie die Datei `meineconfig.cfg`.
12. Starten Sie WMLGEN mit `wmlgen meineconfig.cfg test2`.
13. WMLGEN erzeugt die drei in Abschnitt 3.2.3 erläuterten Dateien.

4.3 Erzeugen und Speichern eines WML-Dokumentes mit eigener Konfiguration und Seed

Sie möchten die selbst erstellte Konfigurationsdatei `meineconfig.cfg` aus dem vorherigen Beispiel erneut verwenden. Allerdings möchten Sie ebenfalls eine Seed für den Zufallsgenerator vorgeben, die unterschiedlich zur Seed des im vorherigen Beispiel erzeugten WML-Dokumentes ist. Das neu zu erzeugende WML-Dokument möchten Sie , dabei soll der Datei-Name mit `test3` beginnen.

1. Legen Sie mit einem Editor Ihrer Wahl eine leere Datei mit dem Namen `meineconfig.seed` an. Bitte beachten Sie, dass der Dateiname (bis auf die Endung) genau mit dem Dateinamen der Konfigurationsdatei übereinstimmen muss.
2. Öffnen Sie mit einem Editor Ihrer Wahl die erzeugte Seed-Datei aus dem vorherigen Beispiel `test2_1058726979_170000.seed` und merken Sie sich die Seed . Nun können Sie diese Datei wieder schließen.
3. Denken Sie sich eine unterschiedliche Seed aus, schreiben diese in die Datei `meineconfig.seed` und Sie.
4. Starten Sie WMLGEN mit `wmlgen meineconfig.cfg test3`.
5. WMLGEN erzeugt die drei in Abschnitt 3.2.3 erläuterten Dateien.
6. Zur Kontrolle können Sie die abgelegte Seed in der nun erzeugten Datei `test3_1058727980_520000.seed` mit der von Ihnen eingegebenen Seed vergleichen.

5 Probleme und Fehlermeldungen

Starten Sie WMLGEN mit dem Parameter `--help` so erhalten Sie Hinweise zum Start des Programms sowie zur Konfigurationsdatei .

Für WMLGEN wird eine *manpage* mitgeliefert, die Sie bei Bedarf installieren können.

5.1 Fehlermeldungen bei der Erzeugung der Dokumente:

Wenn Ihre Dokumente keine Elemente in höherer Dokument-Tiefe haben, so variieren Sie die Grösse der Ausgabedatei, so dass mehr Elemente - und somit auch mehr Elemente in tieferen Ebenen - erstellt werden. Dies macht sich bemerkbar, wenn Sie fast nur card-Elemente mit einem weiteren Kopfelement erhalten.

Wenn Sie während der Dokumenterzeugung eine Warnung des Konfigurationssparsers bekommen, so überprüfen Sie zuerst, ob zu jeder Eigenschaft, welche Sie in der Konfiguration angegeben haben, auch ein Wert angegeben haben. Wenn Sie keinen Wert definieren wollen, so lassen Sie diese Deklaration einfach aus und entfernen sie komplett aus der Konfigurationsdatei .

5.2 Probleme mit den Zeichenkodierungen

Diverse Validatoren und WML-Browser verstehen Zeichenkettencodierungen wie UCS-4 nicht. Um trotzdem Dokumente zu generieren und zu testen wählen sie bitte ein anderes `encoding` aus den unterstützten Kodierungen. Viele Browser ignorieren auch ihnen fremde Kodierungen und zeigen die Zeichen anders an, als Sie es vielleicht erwarten.

A Abkürzungsverzeichnis

- ASCII = **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
- DTD = **D**ocument **T**ype **D**efinition)
- UCS = **U**niversal **C**haracter **S**et
- UTF = **U**niversal **T**ransformation **F**ormat
- WAP = **W**ireless **A**pplication **P**rotocol
- WML = **W**ireless **M**arkup **L**anguage Specification
- XML = **E**xtensible **M**arkup **L**anguage

B Glossar

- **ASCII (American Standard Code for Information Interchange)**: 7 bit breiter Zeichencode, der keine Sonderzeichen erlaubt (auch bekannt als *ISO-646-US* oder *US-ASCII*).
- **Apache**: ein weit verbreiteter Open-Source Web-Server.
- **DTD (Document Type Definition)**: beschreibt die Struktur einer Klasse von SGML- oder XML-Dokumenten mit Hilfe einer Text-Datei, das alle Syntax-Regeln in einem von SGML vorgeschriebenen Format enthält.
- **ISO 8859-1**: auch *Latin-1*. Entspricht einem Zeichensatz mit den meisten europäischen Zeichen.
- **Seed** : ein Computer kann als deterministisches Gerät keine echten Zufallszahlen erzeugen, sondern nur so genannte Pseudozufallszahlen. Algorithmen zur Erzeugung dieser Pseudozufallszahlen müssen mit einer Zahl (*Seed*) initialisiert werden. Bei gleicher Seed werden die gleichen Pseudozufallszahlen erzeugt.
- **UCS-4 (Universal Character Set)**: Neben Unicode eine weitere Zeichenmenge, 2^{31} Zeichen, initiiert durch die International Organization for Standardization (ISO)
- **Unicode**: Menge von Zeichen, initiiert durch ein Konsortium von Herstellern mehrsprachiger Software.
- **UTF-8 (Universal Transformation Format)**: eine Kodierung des Unicode-Zeichensatzes
- **WAP (Wireless Application Protocol)**: Protokoll, das es erlaubt, speziell aufbereitete Seiten im Internet über mobile Endgeräte abzurufen
- **WML (Wireless Markup Language Specification)**: Sprache, in der WAP-Seiten geschrieben sind.
- **XML (Extensible Markup Language)** ist ein Standard des „World Wide Web Consortiums“ (W3C) für Internetsprachen. WML ist ein Beispiel für eine solche Sprache. XML ist eine eingeschränkte Teilmenge von SGML (**Standardised Generalised Markup Language**; Näheres hierzu siehe Internet).

Index

Konfiguration, 3, 6, 7, 9, 11–13

Konfigurationsdatei, 3, 6–9, 12, 13

Seed, 3, 6, 7, 9, 12, 15

Seed-Datei, 3, 9, 12

speichern, 3, 4, 6, 9, 11, 12

Standardkonfiguration, 5–7, 11

starten, 5, 6, 9, 12, 13

testen, 3, 4, 7, 10, 13

Zeichenkodierung, 7, 11

Zeichenkodierungen, 13