

## >> Gentoo Linux Dokumentationsleitfaden

[Bitte Kapitel auswählen] ▀

### 1. Grundlagen

#### 1.1 Ziele des Gentoo-Guide-XML-Designs

Die Guide-XML-Syntax ist zugleich minimalistisch als auch ausdrucksstark, so dass sie schnell erlernbar ist, gleichzeitig aber alle notwendigen Eigenschaften bereitstellt, die wir für das Erstellen von Internet-Dokumenten brauchen. Die Anzahl der Tags ist auf ein Minimum beschränkt -- nur die die wir brauchen. Dadurch wird es einfacher Dokumente in andere Formate, wie z.B. DocBook, XML/SGML oder HTML umzuwandeln.

Das Ziel ist, dass **Erstellen** und **Umwandeln** von GuideXML Dokumenten zu vereinfachen.

#### 1.2 Weitere Quellen

Wenn Sie planen Dokumentation zu Gentoo beizutragen oder GuideXML testen wollen, lesen Sie bitte den [Dokumentations Entwickler Leitfaden](#) der Tipps und Tricks zur Erstellung von Dokumentation beinhaltet.

### 2. Guide-XML

#### 2.1 Grundstruktur

Jetzt, da Sie wissen, wie man Guide-XML umwandelt, sind Sie so weit, um die GuideXML Syntax zu erlernen. Wir beginnen mit den einleitenden Tags, die in einem Guide-XML-Dokument verwendet werden:

**Befehlsauflistung 1:** Der einleitende Teil eines Guide-XML-Dokuments

```
<?xml version='1.0' encoding="UTF-8"?>
<guide link="Relativer Link zu Ihrem Guide">
<title>Gentoo Linux Dokumentationsleitfaden</title>
<author title="Chief Architect">
  <mail link="drobbins@gentoo.org">Daniel Robbins</mail>
</author>
<author title="Editor">
  <mail link="thomasfl@gentoo.org">Thomas Flavel</mail>
</author>
<author title="Übersetzung">
  <mail link="kontakt@hendrik-brandt.de">Hendrik Brandt</mail>
</author>

<abstract>
Dieser Leitfaden erklärt Ihnen, wie Sie Dokumente mit Hilfe der neuen
minimalistischen Gentoo GuideXML Syntax erstellen können.
Diese Syntax stellt das offizielle Format für Gentoo-Linux-Dokumente dar -
auch dieses Dokument wurde mit Hilfe der Gentoo's Guide-XML erstellt.
Dieser Leitfaden setzt Grundkenntnisse in XML und HTML voraus.
</abstract>

<license/>

<version>1.0</version>
<date>29. März 2001</date>
```

In der ersten Zeile sehen wir den notwendigen Tag der festlegt, dass das Dokument in XML beschrieben ist. Darauf folgt der Tag `<guide>` -- das gesamte Dokument ist von einem `<guide>` `</guide>` -Paar umschlossen. Schließlich gibt es noch den Tag `<title>`, mit dem der Titel für das Dokument festgelegt wird.

Als nächstes kommen wir zu den `<author>`-Tags, welche Informationen über die unterschiedlichen Autoren eines Dokumentes enthalten. Jeder `<author>`-Tag besitzt ein optionales Element `title=`. Dieses wird benutzt, um die Beziehungen der Autoren zum Dokument (Autor, Co-Autor, Bearbeiter,

Übersetzer usw.) zu beschreiben. Im vorliegenden Beispiel sind die Namen der Autoren in einem weiteren Tag eingeschlossen -- dem Tag `<mail>`. Mit diesem kann eine Email-Adresse mit der betreffenden Person verbunden werden. Der Tag `<mail>` ist optional und kann weggelassen werden. Weiterhin wird nicht mehr als ein Element `<author>` pro Dokument benötigt.

Es folgen die Tags `<abstract>`, `<version>` und `<date>`, die benutzt werden, um die Überschrift, die gegenwärtige Versionsnummer und das gegenwärtige Versionsdatum (im Format TT MMM JJJJ) des Dokuments festzulegen. Damit sind alle Tags, die zu Beginn des Dokuments stehen sollten genannt. Ebenso wie die Tags `<title>` und `<mail>`, sollten diese nur direkt im Tag `<guide>` und nicht irgendwo anders stehen. Weiterhin wird empfohlen (aber nicht gefordert), dass diese Tags vor dem eigentlichen Inhalt im Dokument stehen.

Und schliesslich haben wir das `<license/>` Tag, mit dem wir das Dokument unter der [Creative Commons - Attribution / Share Alike](#) Lizenz veröffentlichen, wie es die [Dokumentations Richtlinie](#) erfordert.

## 2.2 Kapitel und Abschnitte

Sobald Sie die einleitenden Tags spezifiziert haben, können Sie damit beginnen, Strukturelemente des Dokuments hinzuzufügen. Guide-Dokumente sind unterteilt in Kapitel (chapter) und jedes Kapitel kann ein oder mehr Abschnitte (section) enthalten. Jedes Kapitel und jeder Abschnitt hat eine Überschrift. Hier ist ein Beispielkapitel mit einem Abschnitt, der einen Absatz enthält. Wenn Sie dieses XML an das XML im [vorherigen Auszug](#) anhängen und ein `</guide>` am Ende hinzufügen, haben Sie ein (minimales) Guide-Dokument:

### Befehlsauflistung 2

```
<chapter>
<title>Das ist mein Kapitel</title>
<section>
<title>Hier ist Abschnitt eins in meinem Kapitel</title>
<body>

<p>
Dies ist der eigentliche Textinhalt meines Abschnitts.
</p>

</body>
</section>
</chapter>
```

Wie Sie sehen haben ich dem Kapitel durch das hinzufügen eines Kindelements `<title>` in das Element `<chapter>` eine Überschrift gegeben. Anschliessend habe ich einen Abschnitt durch das hinzufügen des Elements `<section>` erzeugt. Wenn Sie sich das Element `<section>` anschauen, werden Sie zwei Kindelemente erkennen -- ein `<title>` und ein `<body>` Element. Da Ihnen das Element `<title>` bereits bekannt ist, sei hier das Element `<body>` erläutert; dieses enthält den eigenlichen Inhalt des derzeitigen Abschnitts. Wir betrachten nun kurz die Tags die innerhalb von `<body>` erlaubt sind.

### Notiz

Das Element `<guide>` kann mehrere Elemente `<chapter>`, und das Element `<chapter>` kann mehrere Elemente `<section>` enthalten. Allerdings kann das Element `<section>` nur einmal das Element `<body>` enthalten.

## 2.3 Ein Beispiel zu `<body>`

Jetzt ist es an der Zeit zu lernen, wie man den eigentlichen Inhalt auszeichnet. Hier ist der XML-Code zu einem Beispiel-`<body>`-Element:

### Befehlsauflistung 3

```
<p>
Das ist ein Absatz. <path>/etc/passwd</path> ist eine Datei.
<uri>http://www.gentoo.de</uri> ist meine Lieblings-Web-Seite.
Geben Sie <c>ls</c> ein - wenn sie wollen. Ich sollte jetzt <e>wirklich</e> schlafen gehen.
</p>

<pre>
Das ist Textausgabe oder Quelltext.
# <i>das sind Benutzereingaben</i>
```

Um HTML/XML lesbarer zu machen, sollten Sie verschiedene Hervorhebungen verwenden:

```
<foo><i>bla</i></foo>
```

```
<codenote>So wird eine Notiz in einen Quelltextabschnitt eingefügt</codenote>
</pre>
<note>
Das ist eine Anmerkung.
</note>
<warn>
Das ist eine Warnung.
</warn>
<impo>
Das ist wichtig.
</impo>
```

Und so wird dieses `<body>`-Element dargestellt:

Das ist ein Absatz. `/etc/passwd` ist eine Datei. <http://www.gentoo.de> ist meine Lieblings-Web-Seite. Geben Sie `ls` ein - wenn Sie wollen. Ich sollte jetzt **wirklich** schlafen gehen.

#### Befehlsauflistung 4

Das ist Textausgabe oder Quelltext.  
# **das sind Benutzereingaben**

Um HTML/XML lesbarer zu machen, sollten Sie verschiedene Hervorhebungen verwenden:  
<foo><i>bla</i></foo>

```
// So wird eine Notiz in einen Quelltextabschnitt eingefügt
```

#### Notiz

Das ist eine Anmerkung.

#### Warnung

Das ist eine Warnung.

#### Wichtig

Das ist wichtig.

## 2.4 Die Tags `<body>`

Im vorherigen Abschnitt wurde eine Menge neuer Tags eingeführt -- jetzt werden sie erläutert. Die Tags `<p>` (Absatz), `<pre>` (Quelltextabschnitt), `<note>` (Anmerkung), `<warn>` (Warnung) und `<impo>` (Wichtig), können alle ein oder mehrere Zeilen Text enthalten. Neben dem Element `<table>` (auf das wir später eingehen), sollten das die einzigen Tags, die innerhalb des Elements `<body>` stehen, sein. Weiterhin sollten diese Tags **nie** verschachtelt werden -- mit anderen Worten, Sie sollten das Element `<note>` nicht innerhalb des Elements `<p>` verwenden! Wie Sie vielleicht bemerken, gibt das Element `<pre>` Leerzeichen und Zeilenumbrüche exakt wieder - dadurch empfiehlt es sich für Quelltextabschnitte. Sie können auch das `<pre>` Tag benutzen:

#### Befehlsauflistung 5: `<pre>`

```
<pre caption = "Ausgabe von uptime">
# <i>uptime</i>
16:50:47 up 164 days,  2:06,  5 users,  load average: 0.23, 0.20, 0.25
</pre>
```

## 2.5 `<path>`, `<c>` und `<e>`

Die Elemente `<path>`, `<c>` und `<e>` können innerhalb jedes `<body>`-Kindelements verwendet werden, ausgenommen ist `<pre>`.

Das Element `<path>` wird verwendet um Text hervorzuheben, der auf eine **Datei auf der Festplatte** verweist -- genauso wie für einen **absoluten oder realtiven Pfad** oder einfach einen **Dateinamen**. Dieses Element wird normalerweise mit einer Maschinentextschrift dargestellt, um es vom Rest des Textes abzugrenzen.

Das Element `<c>` wird verwendet, um **Befehle** oder **Benutzereingaben** zu markieren. Sie sollten sich `<c>` als eine Möglichkeit vorstellen, dem Leser mitzuteilen, dass er etwas eingeben kann, was zu irgendeiner Aktion führt. Zum Beispiel sind alle XML-Tags in diesem Dokument von `<c>` umschlossen, um zu zeigen, dass der Benutzer etwas anderes als einen Pfad eingeben kann. Indem Sie das Element `<c>` verwenden, erleichtern Sie es Ihren Lesern, Befehle die sie eingeben müssen schneller zu erkennen. Desweiteren ist es auf Grund dessen, dass `<c>`-Elemente sich vom regulären Text abheben,

**nicht zwingend notwendig, Benutzereingaben mit doppelten Anführungszeichen zu umgeben.** Verwenden Sie zum Beispiel das Element "<c>" nicht wie in diesem Satz. Vermeiden Sie die Benutzung von nicht-notwendigen doppelten Anführungszeichen, um das Dokument leserlicher und ansprechender zu gestalten.

<e> wird benutzt, um Wörter oder Wortgruppen zu betonen; zum Beispiel: Ich sollte **wirklich** öfters Semikola verwenden. Wie Sie sehen können, hebt sich dieser Text zur Betonung von der normalen Absatzschrift ab. Dadurch wird das von ihnen Gesagte **schlagkräftiger!**

## 2.6 <mail> und <uri>

Wir hatten uns den Tag <mail> schon vorhin angesehen; er wird benutzt, um einen Text mit einer speziellen Email-Adresse zu verbinden; er hat die Form <mail link="mm@bla.de">Herr Martin Mustermann </mail>.

Der Tag <uri> wird verwendet, um auf Dateien oder Seiten im Internet zu verweisen. Es gibt zwei Varianten. Bei der ersten wird die URI direkt im Text angezeigt, so wie dieser Link zu <http://www.gentoo.de>. Um diesen Link zu erzeugen, habe ich <uri>http://www.gentoo.de </uri> eingegeben. Die Alternative wird verwendet, wenn Sie eine URI mit einem abweichenden Text verknüpfen wollen -- z.B. [das deutschsprachige Gentoo Linux Portal](http://www.gentoo.de). Um **diesen** Link zu erzeugen, wurde <uri link="http://www.gentoo.de">das deutschsprachige Gentoo Linux Portal </uri> eingegeben.

## 2.7 Grafiken

Und so können Sie Grafiken in ein Dokument einfügen: <figure link="meinbild.png" short="mein Bild" caption="für immer mein Lieblingsbild"/ >. Das Attribut link= verweist auf die gewünschte Grafikdatei, das Attribut short= legt eine Kurzbeschreibung fest (wird derzeit für das Attribute alt= im HTML-Tag <img> verwendet) und mit caption= wird schließlich eine Bildunterschrift festgelegt -- alles in allem nicht allzu schwer :) Ausserdem wird noch der Tag  aus HTML unterstützt, um Grafiken ohne Unterschrift, Ränder u.a. einzufügen.

## 2.8 Tabellen und Listen

Guide stellt eine vereinfachte Tabellensyntax, vergleichbar mit der von HTML, bereit. Um eine Tabelle zu erzeugen, benutzen Sie den Tag <table>. Eine Reihe wird mit <tr> begonnen. Allerdings unterstützt GuideXML **keinen** Tag <td> wie in HTML, um die eigentlichen Tabelleninhalte einzufügen; verwenden Sie stattdessen <th> für den Tabellenkopf und <ti> wenn Sie normale Informationen einfügen wollen. Sie können <th> überall da verwenden, wo Sie auch <ti> verwenden können -- es gibt also keinen Zwang <th>-Elemente nur in der ersten Reihe zu verwenden. Derzeit stellen diese Tags noch keine Attribute bereit, allerdings werden bald welche hinzugefügt (wie z.B. ein Attribut caption= für <table>).

Um eine sortierte oder unsortierte Liste zu erzeugen, verwenden Sie einfach die gleichwertigen HTML-Tags <ol>, <ul> und <li>. Listen-Tags sollten nur innerhalb von <p>, <ti>, <note>, <warn> und <impo> verwendet werden.

## 2.9 Verweise innerhalb des Dokuments

In Guide ist es wirklich einfach, Verweise zu anderen Teilen des Dokuments mit Hilfe von Hyperlinks zu erstellen. Sie können einen Verweis zu [Kapitel 1](#) durch das Eingeben von <uri link="#doc\_chap1">Kapitel 1</uri> erzeugen. Um auf [Abschnitt 2 in Kapitel 1](#) zu verweisen, geben Sie <uri link="#doc\_chap1\_sect2">Abschnitt 2 in Kapitel 1</uri> ein. Um auf Grafik 3 zu verweisen, geben Sie <uri link="doc\_chap1\_fig3">Grafik 1.3</uri> ein; oder für einen Verweis auf [Quelltextbeispiel 2.2](#) einfach <uri link="doc\_chap2\_pre2">Quelltextbeispiel 2.2</uri>. In Zukunft werden weitere Möglichkeiten, der automatischen Verknüpfung (wie z.B. Unterstützung von Tabellen) hinzukommen.

Allerdings ändern sich viele Anleitungen öfters und solches "Zählen" kann dann zu fehlerhaften Links führen. Um dem abzuhelpen, kann ein Name für ein <chapter> oder <section> mit Hilfe des id Attributes hinzugefügt werden. Nun muss man nur noch auf dieses Attribut verweisen:

**Befehlsauflistung 6:** Verwendung des id Attributes

```
<chapter id="foo">
<title>Das ist foo!</title>
...
```

```
<p>
Weitere Informationen finden Sie im <uri link="#foo">foo Kapitel</uri>
</p>
```

## 3. Coding Style

### 3.1 Einleitung

Da jede Gentoo Dokumentation eine große Anstrengung ist und viele Leute die Dokumentation ändern, ist ein Coding Style von Nöten. Ein Coding Style beinhaltet zwei Teile. Der erste (Interner Coding Style) erläutert wie die xml-Tags platziert werden sollen, der zweite beschreibt den Inhalt; wie man die Leser nicht verwirrt.

Beide Teile werden jetzt erläutert.

### 3.2 Interner Coding Style

**Neue Zeilen** müssen direkt nach **jedem** GuideXML-Tag platziert werden (öffnende und schliessende), ausgenommen für `<version>`, `<date>`, `<title>`, `<th>`, `<ti>`, `<li>`, `<i>`, `<e>`, `<uri>`, `<path>`, `<b>`, `<comment>`, `<codenote>`, `<mail>`.

**Leere Zeilen** müssen direkt nach **jedem** `<body>` (nur öffnende Tags) und vor **jedem** `<chapter>`, `<p>`, `<table>`, `<author>` (Gruppe), `<pre>`, `<ul>`, `<ol>`, `<warn>`, `<note>` and `<impo>` (nur öffnende Tags) platziert werden.

Ein **Zeilenbruch** muss nach 80 Zeilen angewandt werden, ausser im `<pre>` Tag. Nur wenn es keine andere Wahl gibt kann von dieser Regel abgewichen werden (zum Beispiel wenn eine URL über das Maximum an Zeichen hinausgeht). Der Editor muss dann beim nächsten Leerzeichen umbrechen.

**Einrücken** soll nicht benutzt werden, ausser in XML Konstruktionen bei denen die Parent Tags `<tr>` (von `<table>`), `<ul>`, `<ol>` und `<author>` sind. Wenn Einrücken benutzt wird, **müssen** zwei Leerzeichen für jede Einrückung benutzt werden. Das heisst, **keine** Tabs und **nicht** mehr Leerzeichen.

Wenn ein Zeilenbruch in eines der folgenden `<ti>`, `<th>` oder `<li>` Konstrukte fällt, muss Einrückung für den Inhalt benutzt werden.

Ein Beispiel für Einrückung ist:

**Befehlsauflistung 7:** Beispiel: Einrückung

```
<table>
<tr>
  <th>Foo</th>
  <th>Bar</th>
</tr>
<tr>
  <ti>Das ist ein Beispiel für Einrückung.</ti>
  <ti>
    Wennn Text nicht in eine 80 Zeilen lange Zeile passt, müssen
    Sie Einrückung benutzen, wenn das Parent Tag es erlaubt.
  </ti>
</tr>
</table>

<ul>
  <li>Erste Option</li>
  <li>Zweite Option</li>
</ul>
```

**Attribute** sollen keine Leerzeichen zwischen den Attributen haben. "=" Ein Beispiel:

**Befehlsauflistung 8:** Attribute

```
Falsch :    <pre caption = "Attribute">
Richtig:    <pre caption="Attribute">
```

### 3.3 Externer Coding Style

Innerhalb von Tabellen (`<table>`) und Listings (`<ul>` und `<ol>`), sollten Punkte (".") nur benutzt werden, wenn mehrere Sätze benutzt werden. In diesem Fall sollte jeder Satz mit einem Punkt enden.

Jeder Satz, auch innerhalb von Tabellen und Listings sollte mit einem Großbuchstaben beginnen.

**Befehlsauflistung 9:** Punkte und Großbuchstaben

```
<ul>
  <li>Kein Punkt</li>
  <li>Mit Punkt. Mehrere Sätze, Sie erinnern sich?</li>
</ul>
```

Codelistings sollten **immer** eine *Beschreibung* haben.

Benutzen Sie soweit möglich `<uri>` mit dem *link* Attribut. In anderen Worten, [Gentoo Website](http://www.gentoo.org) wird <http://www.gentoo.org> vorgezogen.

Wenn Sie etwas innerhalb eines `<pre>` Konstruktes kommentieren möchten, benutzen Sie `<codenote>` wenn der Inhalt ein C or C++ Code Schnipsel ist. Andernfalls benutzen Sie `<comment>`. Platzieren Sie den Kommentar **vor** dem Bezug des Kommentars.

**Befehlsauflistung 10:** Kommentar Beispiel

```
(Ersetzen Sie "john" mit Ihrem Benutzer Namen)
# id john
```

## 4. Quellenangaben

### 4.1 Mit dem Schreiben beginnen

Guide wurde speziell dafür entwickelt, es Entwicklern zu ermöglichen, mehr Zeit mit dem Schreiben der Dokumentation und weniger mit dem Erlernen der eigentlichen XML-Syntax zu verbringen. Wir hoffen, dass dies den Entwicklern erlaubt, qualitativ hochwertige Gentoo-Linux-Dokumentation zu schreiben. Wenn Sie uns helfen möchten (oder irgendeine Fragen zu GuideXML haben), dann senden Sie eine Nachricht an die [gentoo-doc Mailing Liste](#) (in Englisch!). Wir wünschen viel Spass;!

