>> Env.d HOWTO

[Bitte Kapitel auswählen]

1. Einführung in env.d und Generische Variablen

1.1 Die Wichtigsten Gentoo Umgebungsvariablen

Eine der verbreitesten Fragen von Gentoo-Nutzern ist: "Wo sind meine Umgebungsvariablen definiert, und was ist deren Inhalt?". Um diese Frage zu verstehen, ist es erst einmal notwendig zu wissen, was eine Umgebungsvariable eigentlich ist. Eine Umgebungsvariable wird immer dann gesetzt, wenn der *export* Befehl ausgeführt wird. Nehmen wir beispielsweise an, Sie führen *export EDITOR="/usr/bin/vim"* aus; dabei setzen Sie eine Umgebungsvariable, die Ihrem System mitteilt, dass Ihr Standardeditor vim ist. Es gibt viele verschiedene Arten von Umgebungsvariablen, die Sie in Ihrem Gentoo-System setzen können. Im Folgenden einige Beispiele hierzu:

Notiz
export setzt eine Umgebungsvariable innerhalb Ihrer Shell **und** in allen Kindprozessen. Sollten Sie nur die Variable setzen wollen, ohne dabei eventuelle Kindprozesse zu beeinflussen, führen sie etwas wie **FOO="bar"** aus, d.h. lassen Sie das *export* Kommando weg.
Notiz
Ihre **00basic** Datei mag anders aussehen. Dies ist kein Problem, da diese Variablen nur die Gentoo-Voreinstellungen wider spiegeln.
Notiz
Sie finden diese Beispiele in /etc/env.d/00basic.

Variable	Funktion
PATH	Dies setzt den Standardpfad, in dem nach Programmen gesucht wird. Trennzeichen ist ':'. Eine Pfadangabe sähe ungefähr so aus: *PATH=/usr/local/bin:/opt/bin*.
ROOTPATH	Diese Variable entspricht der oben genannten Variable, nur wird hier der Standardpfad für den *root* Benutzer gesetzt. Ein Beispiel hierfür wäre etwas wie *ROOTPATH=/usr/local/bin:/opt/bin*.
LDPATH	Diese Variable spezifiziert den Suchpfad des Linkers für Bibliotheken. Auch diese Variable benutzt den ':' als Trennzeichen.
MANPATH	Dies gibt an, wo auf Ihrer Maschine die Manpages liegen; genau wie **PATH**, nur für Manpages anstatt Programmen. Ein Beispiel hierfür wäre *MANPATH=/usr/share/man:/usr/local/share/man*.
INFODIR	Dies gibt an, wo die Info-Dateien abgelegt sind. Ein Beispiel wäre *INFODIR=/usr/share/info*.
PAGER	Dies gibt an welcher Pager zu benutzen ist. Ein Beispiel wäre *PAGER=/usr/bin/less*.
EDITOR	Dies gibt den Standardeditor in Ihrem System an. Eine Angäbe sähe ungefähr so aus: *EDITOR=/usr/bin/vim*.

Dies ist ein Beispiel einer **00basic** Datei. Es soll Ihnen eine Vorstellung davon vermitteln, wie eine solche Datei aussieht. Sollte einmal Ihre **00basic** Datei zerstört werden, so können Sie diese hier benutzen!

Befehlsauflistung 1: 00basic

```
# /etc/env.d/00basic:
# $Header: /home/cvsroot/gentoo-src/rc-scripts/etc/env.d/00basic \
           ,v 1.11 2003/02/17 02:48:39 azarah Exp $

PATH="/usr/local/bin:/opt/bin"
ROOTPATH="/usr/local/bin:/opt/bin"
LDPATH="/usr/local/lib"
MANPATH="/usr/share/man:/usr/local/share/man"
INFODIR="/usr/share/info"
INFOPATH="/usr/share/info"
CVS_RSH="ssh"
PAGER="/usr/bin/less"
LESSOPEN="|lesspipe.sh %s"
```

1.2 env.d Formatierung

Die Dateien in *env.d* sind in einer logischen Reihenfolge angeordnet, so dass wenn *env-update* ausgeführt wird, die Variablen richtig angeordnet werden. Die Ziffern am Anfang des Dateinamens geben die Reihenfolge an, in denen diese durchlaufen werden. Folglich wird **00basic** als erstes durchlaufen, daraufhin **01irgendetwas** und so weiter. Auch der Inhalt der Dateien hat ein besonderes Format:

Befehlsauflistung 2: Dateiformat

```
# Dies ist ein Kommentar, genauso wie in der BASH
VARIABLE1=/Pfad/auf/irgendetwas
# Einige der Variablen können durch : getrennte Argumente besitzen.
VAR1A=/Pfad/auf/etwas:/Pfad/auf/nochein/etwas
VARIABLE2=irgendein_Name
```

2. Spezielle Variablen und Wie Alles Zusammenpasst

2.1 Spezielle Variablen

Es gibt einige spezielle Variablen, die in *env.d* aufgeführt sind. Dies sind: **KDEDIRS, PATH, CLASSPATH, LDPATH, MANPATH, INFODIR, ROOTPATH, CONFIG_PROTECT, CONFIG_PROTECT_MASK**. Diese Variablen sind insofern speziell, als sie in einer besonderen Art und Weise verarbeitet und in spezielle Dateien geschrieben werden, die Ich später erklären werde.

Notiz
Einige der unten stehenden Variablen sind oben schon in **00basic** aufgeführt worden. Dies stellt kein Problem dar, da diese Variablen mehr als einmal verwendet werden können, allerdings nur in verschiedenen Dateien.

Variable	**Funktion**
KDEDIRS	Dies ist der Pfad für alle zum KDE gehörenden Dateien
PATH	Schon zuvor aufgeführt, durch ':' getrennt
CLASSPATH	Setzt den Pfad zu den Java Klassen, durch ':' getrennt
LDPATH	Schon zuvor erwähnt (s.o.)
MANPATH	Schon zuvor erwähnt (s.o.), durch ':' getrennt
INFODIR	Schon zuvor erwähnt (s.o.), durch ':' getrennt
ROOTPATH	Schon zuvor erwähnt (s.o.), durch ':' getrennt
CONFIG_PROTECT	Dies gibt an welche Konfigurationsdateien in /etc von Änderungen durch neue Ebuilds geschützt werden sollen. Die Argumente werden durch Freizeichen ' ' getrennt.
CONFIG_PROTECT_MASK	Dies ist im Grunde das Inverse zu CONFIG_PROTECT, die angegebenen Dateien unter /etc werden freigegeben, sie werden beim Emerge-Prozess automatisch geändert.

2.2 Wie Dies Alles Zusammenpasst

So, nun da ich alles definiert habe, wundern Sie sich wahrscheinlich was um alles in der Welt dies mit Ihnen und Ihrer Gentoo Maschine zu tun hat. Die Macht von **env.d** liegt darin alle diese Variablen Ihren speziellen Bedürfnissen anpassen zu können. Mit den oben angegebenen Definitionen können sie dies durchführen, hoffe ich.

Da wir mittlerweile ein wenig Halt gefunden haben, ist es an der Zeit zu erklären, was mit diesen Variablen geschieht. Wenn *env-update* aufgerufen wird, linkt es alle diese Dateien in ihrer Reihenfolge. Erinnern Sie sich noch, dass ich gesagt habe es wäre in Ordnung eine Variable mehrfach zu definieren, so lange sich die Definitionen in unterschiedlichen Dateien befinden? Dies liegt daran, dass *env-update* sequenziell die Dateien durchgeht und all die Pfadangaben zusammenaddiert. Hätten Sie zum Beispiel eine **PATH** Variable für KDE, GNOME, X und Ihren Standardpfad, würde *env-update* all diese aneinander hängen, so dass ihr Standardwert für **PATH** die Summe all jener Einzelpfade annehmen wird. Nett, nicht wahr?

Nun da diese Variablen zusammengefügt wurden, ist es noch wichtig zu wissen, wo sie gespeichert werden.

Notiz
Die unten aufgeführten Dateien werden bei einem Lauf von *env-update* erzeugt.
Befehlsauflistung 3: Dateien, die bei der Addition aller Dateien aus env.d erzeugt werden

```
LDPATH ----> /etc/ld.so.conf
CONFIG_PROTECT und CONFIG_PROTECT_MASK ----> /etc/profile.env (als exportierte Variablen)
Spezielle und Andere ----> /etc/profile.env
```

Wenn Sie sich die Datei /etc/ld.so.conf einmal ansehen, können sie eindeutig erkennen, dass das Verzeichnis **env.d** deren Quelle war.

Befehlsauflistung 4: /etc/ld.so.conf

```
# ld.so.conf autogenerated by env-update; make all changes to
# contents of /etc/env.d directory
/usr/local/lib
/usr/lib/gcc-lib/i686-pc-linux-gnu/3.2.3
/usr/lib/opengl/xfree/lib
/usr/lib/mozilla
/usr/X11R6/lib
/opt/blackdown-jdk-1.4.1/jre/lib/i386/
/opt/blackdown-jdk-1.4.1/jre/lib/i386/native_threads/
/opt/blackdown-jdk-1.4.1/jre/lib/i386/classic/
/usr/qt/3/lib
/usr/kde/3.1/lib
```

Es ist wichtig zu beachten, dass /etc/profile.env von Ihrer Shell gesourced wird. Dies ist natürlich sinnvoll, denn so können Sie all Ihre neu definierten Variablen benutzen, sobald Sie sich das nächste mal einloggen.

Befehlsauflistung 5: Reihenfolge beim Ablauf von env-update

```
/etc/env.d Einstellungen ---> env-update ---> /etc/profile.env
---> /etc/profile ---> bash ---> all Ihre Programme
```

Nun, da Sie ein neues **profile.env** erzeugt haben, sollte es im Groben so aussehen:

Befehlsauflistung 6: /etc/profile.env

```
(Um die Lesbarkeit zu erhöhen, in diesem Beispiel gekürzt und umgebrochen)
# THIS FILE IS AUTOMATICALLY GENERATED BY env-update.
# DO NOT EDIT THIS FILE. CHANGES TO STARTUP PROFILES
# GO INTO /etc/profile NOT /etc/profile.env
export KDEDIRS='/usr'
export INFODIR='/usr/share/info:/usr/X11R6/info'
export INFOPATH='/usr/share/info:/usr/share/gcc-data/i686-pc-linux-gnu/3.2/
info'
export CONFIG_PROTECT_MASK='/etc/gconf'
export CLASSPATH='/opt/blackdown-jdk-1.4.1/jre/lib/rt.jar:.:.'
export ROOTPATH='/usr/local/bin:/opt/bin:/usr/i686-pc-linux-gnu/gcc-bin/3.2:/
usr/X11R6/bin:usr/qt/3/bin:/usr/kde/3.1/sbin:/usr/kde/3.1/bin'
export CONFIG_PROTECT='/usr/X11R6/lib/X11/xkb /usr/kde/3.1/share/config /usr/
share/texmf/tex/generic/config/ /usr/share/texmf/tex/platex/config/ /usr/
share/config'
export MANPATH='/usr/share/man:/usr/local/share/man:/usr/share/gcc-data/i686-
pc-linux-gnu/3.2/man:/usr/X11R6/man:/usr/qt/3/doc/man'
export PATH='/usr/local/bin:/opt/bin:/usr/i686-pc-linux-gnu/gcc-bin/3.2:/usr/
X11R6/bin:/usr/qt/3/bin:/usr/kde/3.1/bin'
export KDEDIR='/usr/kde/3.1'
export JDK_HOME='/opt/blackdown-jdk-1.4.1'
export JAVAC='/opt/blackdown-jdk-1.4.1/bin/javac'
export LESS='-R'
export CC='gcc'
export PAGER='/usr/bin/less'
export HOSTNAME='fabula'
export G_BROKEN_FILENAMES='1'
export QMAKESPEC='linux-g++'
export LESSOPEN='|lesspipe.sh %s'
export CXX='g++'
export CVS_RSH='ssh'
export QTDIR='/usr/qt/3'
export JAVA_HOME='/opt/blackdown-jdk-1.4.1'
export XINITRC='/etc/X11/xinit/xinitrc'
export GDK_USE_XFT='1'
export MOZILLA_FIVE_HOME='/usr/lib/mozilla'
```

Nun, das war was ich zu sagen hatte, also auf und viel Spaß; nur zerstören Sie sich bitte nicht allzu viel.


W3C
HTML
4.01