



[FAQ](#)

[System](#)

Environment shell-unabhängig setzen

“Was mache ich, wenn ich irgendeine obskure Environment-Variable immer, für alle shells und unter allen Bedingungen setzen will?”

[\[Textversion\]](#)

Es gibt eine Datei `~/ .MacOSX/environment.plist`, in der man einfach (einfach?) mit dem *Property-List-Editor* oder mit einem beliebigen Editor einen entsprechenden Eintrag hinzufügen kann. Interessanterweise haben alle shells, die Apple mitliefert, entsprechende Funktionen an Bord, die diese Datei auswerten.

Beispiel: Fügt man die Zeilen

```
<key>LANG</key></code>
<string>de_DE.ISO8859-1</string>
```

in das `<dict></dict>` ein, so hat man großflächig Zeichensatzprobleme erledigt.

Falls die `environment.plist` nicht vorhanden sein sollte, kann man folgendes Beispiel als Grundgerüst verwenden:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://
<plist version="1.0">
  <dict>
    <key>LANG</key>
    <string>de_DE.ISO8859-1</string>
    <key>MM_CHARSET</key>
    <string>ISO8859-1</string>
  </dict>
</plist>
```

Abkürzen läßt sich diese Schreiarbeit, indem man einfach unter seinem normalen Benutzeraccount

```
defaults write "${HOME}/.MacOSX/environment" LANG de_DE.ISO8859-1
defaults write "${HOME}/.MacOSX/environment" MM_CHARSET ISO8859-1
```

ausführt, um zu obigem Ergebnis zu kommen. Das *defaults* Kommando in Mac OS X ist bestens dazu geeignet, Schlüssel-/Werte-Paare in Property Lists syntaktisch korrekt zu bearbeiten (Näheres in der Manual Page dazu).

Achtung: Erfolgt der Login an dem OS X-Rechner nicht lokal, sondern bspw. per *ssh*, werden die in der `~/ .MacOSX/environment.plist` gesetzten Variablen nicht

ausgewertet. Hier ist es dann dennoch notwendig, die verwendete Shell entsprechend zu konfigurieren.

Das kann sehr einfach geschehen, indem man mittels *defaults* die passenden Werte aus `environment.plist` einfach ausliest:

```
defaults read "${HOME}/.MacOSX/environment" LANG
```

Als Beispiel für die (ba)sh:

```
export LANG=$(defaults read "${HOME}/.MacOSX/environment" LANG)
```

Passendes Beispiel für (t)csh:

```
setenv LANG `defaults read "${HOME}/.MacOSX/environment" LANG`
```

Für ganz harte Fälle hat `macosxhints` noch einen [Vorschlag](#) - ein *ruby*-Programm, mit dem man `environment.plist` in das Environment einer shell zu übertragen.

Zudem existiert ein *Preference Pane* namens [RCEnvironment](#), welches das komfortable Editieren der `environment.plist` ermöglicht.

[Noses](#) und [Thomas Bodenmann](#) - Ergänzungen durch Thomas Kaiser und Clemens Beier per E-Mail.

eingestellt von [alex](#) am Mittwoch, 26. November 2003

Kommentare

[Dominik Schlütter](#) am 4.12.2003 02:18 Uhr:

Diese 'environment.plist' ist auch praktisch, wenn man Apples X11 benutzt - damit lassen sich die Umgebungsvariablen beispielsweise in den xterms nutzen (die haben per Default keine Login-Shells und kennen daher auch nicht die Pfade aus `.login` oder `.tcshrc/.profile & Co.`)

Die in der `.plist` gesetzten Umgebungsvariablen werden übrigens von den Werten aus den o.g. Startdateien überschrieben.

Dominik.

[Arthur Müller](#) am 15.01.2005 12:53 Uhr:

Kann man auf diese Weise tatsächlich Umgebungsvariablen setzen? Also:

- weder das Verzeichnis `.MacOSX` im homedirectory existiert
- noch existiert auf dem gesamten System eine Datei Namens "environment.plist".

Ich hab MacOSX in Version 10.3.6. Versteh ich da irgendwas nicht?

LG,
R2

[Ich](#) am 23.01.2005 13:05 Uhr:

Probieren geht über studieren.

- [↑ Seitenanfang](#)
- [← de-soc-mac.de](#)
- [→ Impressum](#)

URI des Dokuments: http://faq.de-soc-mac.de/system/environment_shellunabhaengig_setzen.shtml - letzte Änderung:
Sunday, 23.01.2005 - 13:05:44 Uhr