

Q: How do I access the individual 'mounts' database entries using niutil? Because of the slashes in the property values, niutil doesn't seem to understand the input.

A: Indeed, if you try something straightforward like:

```
niutil -read . /mounts//dev/sd0a
```

you get an error message such as

```
niutil: can't open ./mounts//dev/rsd0a
```

because niutil is using / to separate properties. So, let's say we have the following:

```
% niutil -list . /mounts
133  /dev/sd0a
134  /dev/sd0b
137  /dev/sd1a
```

and we want to execute `niutil -read . /dev/sd0a` entry. We have to find some way to escape those slashes in the property value (`^/dev/sd0a^`) or we're sunk. The simple ways don't work:

```
% niutil -read . /mounts\dev\rsd0a
niutil: can't open ./mounts/dev/rsd0a
```

```
% niutil -read . /mounts//dev/rsd0a'
niutil: can't open ./mounts//dev/rsd0a
```

Turns out it's necessary to use a DOUBLE backslash to get the proper escaping:

```
% niutil -read . /mounts/\dev\sd0a
name: /dev/sd0a
dir: /
type: 4.3
opts: rw noquota noauto
freq: 1
```

passno: 1

Why? Because if one types simply '\V' the shell gobbles the escape character (the \), and transform this into '/'. If one types TWO slashes, then the shell turns them into one, and there is one in the string passed to niutil as its argument.

QA344

Valid for 1.0, 2.0,3.0,3.1