dscl

Directory Service command line utility.
Create, read, and manage Directory Service data. If invoked without any commands, dscl runs in an interactive mode, reading commands from standard input. Interactive processing is terminated by the quit command. (dscl and the GUI 'Directory utility' replace the older 'NetInfo Manager')

```
Syntax
      dscl [options] datasource [command]

Options
   -p           Prompt for password
   -u user      Authenticate as user
   -P password  Authentication password
   -f filepath  targeted local node database file path
   -raw         Don't strip off prefix from DirectoryService API constants
   -plist       Print out record(s) or attribute(s) in XML plist format
   -url         Print record attribute values in URL-style encoding
   -q           quiet - no interactive prompt

Commands:
   -read [path [key ...]]
   -readall [path [key ...]]
   -readpl path key plist_pat
   -readpli path key value_in
   -list path [key]
   -search path key val
   -create record_path [key [
   -createpl record_path key
   -createpli record_path key
   -append record_path key va
   -merge record_path key val
   -delete path [key [val ...
   -deletepl record_path key
   -deletepli record_path key
   -change record_path key ol
   -changei record_path key v
   -diff path1 path2 [key ...
   -passwd user_path [new_pas

Available only in interactive
   -cd dir
   -pushd [dir]
   -popd
   -auth [user [password]]
   -authonly [user [password]
   -quit
```

Leading dashes ("-") are optional for a

dscl operates on a datasource specif).2 or later) host specified by DNS hostname or IP address. Node r n paths beginning with a dot (".") character, which specifies the local dc dress form is used then the user must specify the -u option and either the -P or -p options to specify an administrative user and password on the remote host to authenticate with to the remote host. The exception to this is if "localhost" is specified.

Passing passwords on the command line is inherently insecure and can cause password exposure. For better security do not provide the password as part of the command and you will be securely prompted.

Path Specification

There are two modes of operation when specifying paths to operate on. The two modes correspond to whether the datasource is a node or a host. In the case of specifying a node, the top level of paths will be record types. Example top level paths would be:

```
/Users/alice
/Groups/admin
```

In the case of specifying a host as a data source, the top level of paths correspond to Open Directory plug-ins and Search Paths. One can specify the plug-in to traverse to a node name, after which the paths are equivalent to the former usage. The following might be the equivalent paths as the above paths:

```
/NetInfo/root/Users/alice
/LDAPv3/10.0.1.42/Groups/admin
```

If path components contain keys or values with embedded slash characters, the slash characters must be escaped with a leading backslash character. Since the shell also processes escape characters, an extra backslash is required to correctly specify an escape. For example, to read a mount record with the name "ldapserver:/Users" in the "/Mounts" path, the following path would be used:

```
dscl . -read /Mounts/ldaphost:\/Users
```

```
 All pathnames are case-sensitive.
```

Commands

The action of each command is described below. Some commands have aliases. For example, "cat" and "." are aliases for "read".

`read [path [key ...]]`

Prints a directory. Each of the properties are printed one per line. The property key is followed by a colon, then a space-separated list of the values for that property. Note that a value which contains embedded spaces will appear identical to a pair of values.

If The `-raw` flag for raw output has been given, then read prints the full DirectoryService API constant for record and attribute types.

If the -url flag has been specified then printed record path attribute values are encoded in the style of URLs. This is useful if a script or program is trying to process the output since values will not have any spaces or other control characters. Alias: cat

`readall [path [key ...]]`

readall prints all the records of a given type. The output of readall is formatted in the same way as read with a "-" on a line as a delimeter between records.

`readpl path key plist_path`

Prints the contents of plist_path
e for the path.

If the plist_path is the key for a c
_path. If plist_path is
the key for a string, number, boo

`readpli path key value_inde`

Prints the contents of plist_path
then a whitespace, and then
the value for the path.

If the plist_path is the key for a c
_path. If plist_path is
the key for a string, number, boo

`list path`

Lists the subdirectories of the g
search path, the names are
preceded by an index number th
ath. Alias: ls

When used in interactive mode,

`search path key val`

Searches for records that match
path or a record type path. Valid keys
are Directory Service record att

`create record_path [key [va`

Creates a record, property, or va
d if it does not exist. If a key is given,
then a property with that key will be created. Alias: mk

WARNING - If a property with the given key already exists, it will be destroyed and a new property will be created in its place. To add values to an existing property, use the append or merge commands.

If values are included in the command, these values will be set for the given key.

NOTE - Not all directory nodes support a property without a value. An error will be given if you attempt to create a property with no value in such a directory node.

`createpl record_path key plist_path val1 [val2 ...]`

Creates a string, or array of strings at plist_path.

If you are creating a value at the root of a plist that is an array, use "0" as the plist_path.

If only val1 is specified, a string will be created at plist_path. If val1 val2 ... are specified, an array of strings will be created at plist_path.

WARNING - If a value with the given plist_path already exists, it will be destroyed and a new value will be created in its place.

`createpli record_path key value_index plist_path val1 [val2 ...]`

Creates a string, or array of strings at plist_path for the plist at value_index of the key.

If you are creating a value at the root of a plist that is an array, use "0" as the plist_path.

If only val1 is specified, a string will be created at plist_path. If val1 val2 ... are specified, an array of strings will be created at plist_path.

WARNING - If a value with the given plist_path already exists, it will be destroyed and a new value will be created in its place.

append *record_path key val ...*

Appends one or more values to a property in a given record. The property is created if it does not exist.

merge *record_path key val ...*

Appends one or more values to a property in a given directory if the property does not already have those values. The property is created if it does not exist.

change *record_path key old_val new_val*

Replaces the given old value in the list of values of the given key with the new value in the specified record.

changei *path key index val*

Replaces the value at the given index in the list of values of the given key with the new value in the specified record. index is an integer value. An index of 1 specifies the first valu
r.

diff *path1 path2 key ...*

Compares the data from path1

delete path [key [val ...]]

Delete a directory, property, or v
ry. This can only be used on record
type and record paths. If a key is
re given, those values will be
removed from the property with

deletepl *record_path key pl*

Deletes a value in a plist. If no v
en, deletepl deletes the values within
plist_path.

deletepli *record_path key v*

Deletes a value for the plist at va
If one or more values are given,
deletepli deletes the values with

passwd *user_path [new_passw*

Changes a password for a user
henticated to the node (either by
specifying the -u and -P flags or
a new password. If you are not
authenticated then the user's old
active mode, you will be prompted for
them.

Interactive Commands

cd *dir*

Set the current directory. Path names for other dscl commands can be relative to the current directory.

pushd *path*

Similar to the pushd command commonly found in Unix shells. When a path is specified it sets the current directory while pushing the previous directory on to the directory stack. If no path is specified it exchanges the top two elements of the directory stack. It will also print the final directory stack. Alias: pd

popd

Pops the directory stack and returns to the new top directory. It will also print the final directory stack.

auth [*user* [*password*]]

Authenticate as the named user, or as "root" if no user is specified. If a password is supplied, then that password is used for authentication, otherwise the command prompts for a password. Alias: su

If dscl is run in host mode, then when this command is run the current directory must be in the subdirectories of a node.

authonly [*user* [*password*]]

Used to verify the password of a named user, or of "root" if no user is specified. If a password is supplied, then that password is used for authentication, otherwise the command prompts for a password.

If dscl is run in host mode, then when this command is run the current directory must be in the subdirectories of a node.

`quit`

Ends processing of interactive commands and terminates the program. Alias: q

`command history`

The up and down arrow keys will scan through the command history.

`tab completion`

When pathnames are being typed, pressing the tab key will result in a search to auto-complete the typed partial subdirectory name. It will also attempt to correct capitilization in the process.

Diagnostics

dscl will return -1 (255) on error.

Examples

View a record in the local directory no

```
$ dscl . -read /Users/ww
```

Create or replace the UserShell attrib

```
$ dscl . -create /Users/
```

Create or replace the test key of the m ttribute of the `user64` user record:

```
$ dscl . -createpl /User                                t value
```

List the uniqueID values for all user re

```
$ dscl /LDAPv3/ldap.comp
```

Append a value that has spaces in it:

```
$ dscl . -append /Users/
```

Check the SecureToken Status for us

```
$ dscl . -read /Users/us
```

“Employees must be given responsil ple won’t stay without them” ~ Robert Heller

Related macOS commands

Local man page: dscl - Command line
createhomedir - Create and populate home directories on the local computer.
diskutil - Disk utilities.
profiles - Profiles Tool.
sysadminctl - Administer system user accounts.
Stack Exchange - Create a local macOS administrator account from the command line.
DirectoryService(8)
DirectoryServiceAttributes(7)
Equivalent (Linux) bash command: ldapmodify - LDAP modify entry and LDAP add entry tools.