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       Authentica
   -P password   Authentica
   -f filepath   targeted l
   -raw          Don't stri
   -plist        Print out
   -url          Print reco
   -q            quiet - no

Commands:
   -read [path [key ...]]
   -readall [path [key ...
   -readpl path key plist_
   -readpli path key value_
   -list path [key]
   -search path key val
   -create record_path [ke
   -createpl record_path k
   -createpli record_path
   -append record_path key
   -merge record_path key
   -delete path [key [val
   -deletepl record_path k
   -deletepli record_path
   -change record_path key
   -changei record_path key
   -diff path1 path2 [key
   -passwd user_path [new_

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

Leading dashes ("-") are optional for all commands.

dscl operates on a datasource specified on the command line. This can be a node name or a macOS Server (10.2 or later) host specified by DNS hostname or IP address. Node names can be absolute paths beginning with a slash ("/"), or relative domain paths beginning with a dot (".") character, which specifies the local domain, or "..", specifying the local domain's parent. If the hostname or IP address 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 k   e escaped with a leading backslash character. Since th   to correctly specify an escape. For example, to read a mount rec   path would be used:

```
dscl . -read /Mounts/l
```

All pathnames are cas

Commands

The action of each command   nd "." are aliases for "read".

read [*path* [*key* ...]]

Prints a directory. Each of the   n, then a space-separated list of the values for that property. N   a pair of values.

If The -raw flag for raw outpu   record and attribute types.

If the -url flag has been speci   Ls. This is useful if a script or program is trying to process   . Alias: cat

readall [*path* [*key* ...]]

readall prints all the records   with a "-" on a line as a delimeter between records.

readpl *path key plist_pa*

Prints the contents of plist_pa   alue for the path.

If the plist_path is the key for a dictionary or array, the contents of it are displayed in plist form after the plist_path. If plist_path is the key for a string, number, bool, date, or data object, only the value is printed out after the plist_path.

readpli *path key value_index plist_path*

Prints the contents of plist_path for the plist at value_index of the key. The plist_path is followed by a colon, then a whitespace, and then the value for the path.

If the plist_path is the key for a dictionary or array, the contents of it are displayed in plist form after the plist_path. If plist_path is the key for a string, number, bool, date, or data object, only the value is printed out after the plist_path.

list *path*

Lists the subdirectories of the given directory. Subdirectories are listed one per line. In the case of listing a search path, the names are preceded by an index number that can act as a shortcut and used in place of the name when specifying a path. Alias: ls

When used in interactive mode, the path is optional. With no path given, the current directory will be used.

search *path key val*

Searches for records that match a pattern. The search is rooted at the given path. The path can be a node path or a record type path. Valid keys are Directory Service record attribute types.

create *record_path* [*key* [*val* ...]]

Creates a record, property, or value. If only a record path is given, the create command will create the record 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   created in its place. To add values to an existing property

If values are included in the c

NOTE - Not all directory node   create a property with no value in such a directory node.

createpl *record_path key*

Creates a string, or array of s

If you are creating a value at

If only val1 is specified, a stri   gs will be created at plist_path.

WARNING - If a value with the   created in its place.

createpli *record_path key*

Creates a string, or array of s

If you are creating a value at

If only val1 is specified, a stri   gs will be created at plist_path.

WARNING - If a value with the   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 value. An index greater than the number of values in the list will result in an error.

`diff` *path1 path2 key ...*

Compares the data from path1 and path2 looking at the specified keys (or all if no keys are specified).

`delete path [key [val ...]]`

Delete a directory, property, or value. If a directory path is given, the delete command will delete the directory. This can only be used on record type and record paths. If a key is given, then a property with that key will be deleted. If one or more values are given, those values will be removed from the property with the given key. Alias: rm

`deletepl` *record_path key plist_path* [*val ...*]

Deletes a value in a plist. If n̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲given, deletepl deletes the values within plist_path.

`deletepli` *record_path key*

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

`passwd` *user_path* [*new_pas*

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

Interactive Commands

`cd` *dir*

Set the current directory. Pat ̲ ̲ ̲

`pushd` *path*

Similar to the pushd comman ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ nt directory while pushing the previous directory on to the d ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ ̲ he directory stack. It will also print the final directory stack. Alias ̲

`popd`

Pops the directory stack and ̲ ̲ ̲

`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

    $ dscl . -read /Users/

Create or replace the UserShell att

    $ dscl . -create /User

Create or replace the test key of th   s attribute of the `user64` user record:

    $ dscl . -createpl /Us   est value

List the uniqueID values for all user

    $ dscl /LDAPv3/ldap.co

Append a value that has spaces in

    $ dscl . -append /User

Check the SecureToken Status for

    $ dscl . -read /Users/

"Employees must be given respon   *people won't stay without them"*
~ Robert Heller

Related macOS commands

Local man page: dscl - Command line help page on your local machine.
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.

ⓘ

(SS64)

SS64.com asks for your consent to use your personal data to:

Personalised advertising and content, advertising and content measurement, audience research and services development

Store and/or access information on a device

Your personal data will be processed and information from your device (cookies, unique identifiers, and other device data) may be stored by, accessed by and shared with 134 TCF vendor(s) and 63 ad partner(s), or used specifically by this site or app.

Some vendors may process your personal data on the basis of legitimate interest, which you can object to by managing your options below. Look for a link at the bottom of this page to manage or withdraw consent in privacy and cookie settings.