This function displays the relative distinguished names of the children below your current position in the DIT. Typically, if there are a large number of children, you will not be able to list them all, as a "sizelimit" will prevent you. For example

```
Dish -> list
1   commonName=Camayoc
2   commonName=Colin Robbins
3   commonName=Jess
4   commonName=Julian Onions
5   commonName=lancaster
6   commonName=PP Support
7   commonName=QUIPU Support
8   commonName=Tony Roadknight
9   commonName=Hugh Smith
10  commonName=Peter Cowen
(Limit problem)
Dish ->
```

shows a size limit of 10. Using the `-sizelimit` service control you can often increase this limit up to some administrative limit (typically 100) set by the DSA you are connected to.

`-nocache` tells list to reread the directory — do not use cached information.

`-noshow` tells list not to display the names found, but still construct sequences (see Section 4.2).

`-move` is used to tell DISH to change the current position in the DIT to the object specified.

4.1.4 Search

```
search  [[-object] <object>]
        [-baseobject] [-singlelevel] [-subtree]
        [[-filter] <filter>]
        [-[no]relative] [-[no]searchaliases]
        [-[no]partial] [-hitone]
        [<any of the read arguments>]
        [-fred [-expand] [-full] [-summary]
               [-nofredseq] [-subdisplay]]
```

Search the DIT, starting at the object specified, for entries that match the given filter. When an entry is found to match the filter, only distinguished name is printed unless a `-show` option is specified, when the attributes are displayed as well.

If no filter is specified, a default filter (which matches any object in the DIT) is used. This operation has a similar effect to a list operation, but attributes can be shown at the same time.

If no flags are given to a search command, then only one parameter is allowed. This is taken to be the filter and NOT the base object as in ALL the other commands. The flags `-filter` and `-object` are supplied to allow the user to specify if both a filter and base object are required. (*Note:* only one need be flagged.) For example, the following are all valid search commands:

```
search
search <filter>
search -filter <filter>
search -object <object>
search -filter <filter> -object <object>
search <filter> -object <object>
search -filter <filter> <object>
```

whereas

```
search <object>           (interpreted as a filter !)
search <filter> <object>
```

are not valid.

The three flags `-baseobject`, `-singlelevel`, and `-subtree` are mutually exclusive. They are used to control the depth of the search operation. The `-baseobject` option searches only the specified object. The `-singlelevel` option specifies all siblings of the specified object should be searched. The flag `-subtree` searches all the levels below and including the current position recursively. The default is `-singlelevel`.

For example:

```
search -object
   "@c=GB@o=University College London@ou=Computer Science"
   -subtree -filter "cn=colin robbins" -type telephoneNumber
   -show
```

Class	Characters
1	BFPV
2	SCGJKQXZ
3	DT
4	L
5	MN
6	R
0	all others (ignored)

Figure 4.1: Soundex Character Classes

will search the DIT subtree below "c=GB @ o=University College London @ ou=Computer Science" for an exact match of "colin robbins", and then call showentry to display the *telephoneNumber* attribute.

The `-norelative` flag is used to tell `showname` to print the full distinguished name of the results; otherwise, the name relative to the current position is printed.

The `-[no]searchaliases` flag directs the search as to whether aliases encountered in the search should be dereferenced and thus searched. This is different to the `-[dont]dereferencealias` service control which defines what happens to aliases found when locating the base object of the search, whilst the `-[no]searchaliases` controls aliases encountered in the siblings of base search object. The default is "`-dereferencealias -nosearchaliases`".

The `-hitone` flag is used to tell DISH to consider it an error if the search returns more than one entry. This is particularly useful when using DISH from shell scripts (see Section 4.7).

From time to time, search will not be able to do the entire search, and will return partial results; the `-nopartial` directs DISH not to print the partial result information. To see the full set of partial results, both `-partial` and `-show` are needed; otherwise only a summary is printed. Unlike referral errors, DISH does not follow these partial references.

More on Matching

When searching for an exact match ("="), the "`*`" character is used as a wildcard, and a substring match takes place (unless only a "`*`" is specified, when a *presence* match is used; hence any entry containing that attribute will

be matched). For example, "`Colin*`" will match any attribute value starting with the string "`Colin`" and "`*Robbins`" any attribute value ending in the string "`Robbins`". When constructing a substring filter, it should be noted that a QUIPU DSA can resolve searches of the form `string*` significantly faster than searches of the `*string*` or `*string` form.

Remember that a large number of the standard attributes match case independently; for example "x-tel" will match "X-Tel".

As well as an exact equality match, the following types of match can be specified:

```
~=      Approximately equals.
>=      Greater than or equal.
<=      Less than or equal.
```

The Approximate match method used is DSA dependent. A QUIPU DSA uses a soundex based algorithm. This works by grouping similar-sounding characters into classes. The classes and their corresponding characters are shown in Figure 4.1. The first character of a word is always used as the first character of its corresponding soundex code. Adjacent similar characters are ignored. Thus, the word "Robbins" has soundex code "R152". Since the word "Robens" also has soundex code "R152", these two words are approximately equal. To match multiple words each of the target words must appear in order in the string to which it is being compared. There may, however, be other items in between the words matched. For example: "Tim Howes" would match "Timothy Alan Howes" since "Tim" matches "Timothy," "Howes" matches "Howes," and the matched words are in the proper order.

Filter items can be linked with *and* "`&`", or *or* "`|`". Brackets "`( )`" should be used to enforce the Boolean ordering of the expressions, otherwise the evaluation is left to right. A *not* "`!`" can be used to specify the Boolean *not* operator.

If a filter does not have a type specified, e.g., "-filter steve", then an approximate match for the specified common name is assumed (in this case "-filter cn~=steve" is assumed).

A more complex example of a filter that searches for anybody whose is a member of staff or a student, who has a "drink" attribute, whose name approximately matches "steve", but whose surname is not "Kille":

```
cn~=steve & (userClass = staff | userClass = student) &
            (! surname = Kille) & drink = *
```

A list of all the attributes that can be used to search for an entry is given in Chapter 10, together with a description of the type matching allowed.

FRED flags

These flags are used by FRED, and are activated when the `-fred` flag is given.

The `-expand` flag indicates that any matched entry should be displayed in full followed by children entries.

The `-full` flag indicates that if more than one entry is matched on a search, then all entries should be displayed in full.

The `-summary` flag indicates that a one-line display should be used for the entry.

The `-subdisplay` flag indicates that a one-line display should be used for the entry, followed by the entry's children.

The `-nofredseq` flag indicates that the entry (and children) should not be added to the current sequence.

The `-fredlist` flag indicates that any attributes containing DNs should be flagged when using the DA service.

4.1.5 Add Entry

```
add [<object>] [-draft <draft> [-noedit]]
    [-template <draft>]
    [-newdraft] [-objectclass <objectclass>]
```

Add is used to add entries to the DIT. This invokes editentry on the draft entry. If there is not a draft entry specified, a draft entry of the specified objectclass (default — thornperson & quipuObject) is created. The default draft file is *.dishdraft* in your home directory, this can be altered with the `-draft` option.

When editing has finished, an add operation is sent to the directory (the `-noedit` flag following a `-draft` stops the editor being invoked).

`-newdraft` causes the current draft to be overwritten with a new template of the appropriate objectclass.

`-template` is used to specify a template that should be used during editing.

On successful completion of the add, the draft entry is renamed with a suffix of ".old".

4.1.6 Editentry

```
editentry <draft>
```

This option is only available from the shell, and not from the DISH program. It is generally not invoked directly by a user, but by the other DISH commands.

editentry invokes an editor (as defined by the users $EDITOR environment variable) on the current draft entry. In a future release it is hoped to have an editor that knows about the syntax of an entry, and thus ensures that the entry you have supplied is syntactically correct.

4.1.7 Delete Entry

```
delete [<object>]
```

The specified entry is removed from the DIT (remember that X.500 only allows leaf entries to be deleted).

4.1.8 Modify Entry

```
modify [<object>] [-draft <draft> [-noedit]] [-newdraft]
       [-add <attribute type>=<attribute value>]
       [-remove <attribute type>=<attribute value>]
       [<any of the showentry arguments>]
```

Modify is used to modify existing entries in the DIT, and has two modes of operation. The first is used to modify specific attributes and uses the `-add` and `-remove` flags to add and remove single attributes and values. Many of these can be strung together in the same command, e.g.,

```
modify -add "description=new attribute" -remove "drink=Chocolate".
```

The second method is used for altering many attributes at the same time. Initially "`showentry -all -edb -nocache`" is used to get the current DIT entry, and place a copy of it in your .dishdraft file. If the `-draft` option is specified then the given file is used. After editing, any changes to the entry are sent to the directory (the `-noedit` flag following a `-draft` stops the editor being invoked).

The draft file is handled in the same way as with add, except that when a new draft is created, the current values of the attributes are read from the directory.

4.1.9 ModifyRDN

```
modifyrdn [<object>] -name <newrdn> [-[no]delete]
```

This is used to modify the RDN of an entry. The RDN cannot be changed using the modify operation (remember, as with delete X.500 only allows this operation on leaf nodes).

The `-nodelete` flag is used to prevent the old RDN being removed as an attribute of the entry.

4.1.10 Showname

```
showname [<object>] [-[no]compact] [-[no]cache] [-[no]ufn]
         [<any of the read arguments>]
```

A name can be printed either in compact form (as seen in EDB or .dishdraft files), just showing the distinguished name, in a UFN format (default), or by showing the distinguished attributes one per line.

4.1.11 Compare

```
compare [<object>] -attribute <attribute> [-[no]print]
```

For example

```
compare userclass=student
```

This command has a return value of true or false (1 or 0).

If `-print` is specified the strings "TRUE" or "FALSE" are printed.

4.1.12 Squid

```
squid [-sequence [<name>]]
      [-alias <object>] [-version]
      [-user] [-syntax]
      [-fred]
```

The command *squid* (status QUIPU DISH) with no parameters will inform you of the current status of your DISH process for example:

```
Connected to Armadillo at Internet=128.40.16.220+2005
Current position: @c=GB@o=Nottingham University
User name: @c=GB@o=University College London@
           ou=Computer Science@cn=Colin Robbins
Current sequence 'default'
```

`-sequence` makes squid print the specified sequence. Sequences are described in Section 4.2.

`-alias` adds the given DN to the current sequence.

`-version` prints the version number of the DISH process and associated ISODE and DSAP libraries.

`-user` prints the DN you are bound as.

`-syntax` prints the directory syntaxes known to DISH.

`-fred` must be the first argument if supplied. It causes any Distinguished names printed by the command to be shown using the user-friendly naming notation rather than the DN notation.

4.1.13 Bind

```
bind [-noconnect] [[-user] [<username>]]
     [-password [<password>]] [-[no]refer]
     [-call <dsaname>]
     [-simple] [-protected] [-strong] [-noauthentication]
```

This is used to (re)connect to the directory. This is not normally required, as each command will bind as necessary, but allows advanced users to contact specified DSAs and change their username.

`-noconnect` is used to start DISH (and cache) without connection to a directory.

`-[no]refer` is used to tell DISH whether to automatically follow referrals issued by the DSA, by default they are followed.

To connect to the directory you need to be authenticated. There are four flags to control the level of authentication DISH attempts to use. The lowest level of authentication is "anonymous", which is used when you do not supply a distinguished name. A DSA may respond with "inappropriate authentication" which means you should use a higher level of authentication. The next level of authentication is achieved by supplying a Distinguished Name only, with no password. The `-noauthentication` is used to tell DISH not to prompt for a password. The next levels are "simple" and "protected simple" authentication which use the `-simple` and `-protected` flags respectively. These present a textual password to the directory which the directory the uses to authenticate you. Using `-simple` the password is sent across the network "in the clear," whereas `-protected` encrypts it; thus you are advised to used `-protected` if your directory supports it.

The `-strong` flag tells DISH to send strong authentication credentials to the directory. This is not fully implemented in this version of QUIPU.

To connect to the DSA using "simple" or "protected" authentication, you must supply a distinguished name (`-user`) and password (`-password`). This, however, reveals your password to anybody who is looking at your terminal but is useful for issuing bind operations from a shell script. To prevent this, you need to place the password in your UNIX protected *.quipurc* (see Section 4.4.1). Alternatively, if you do not supply a password, you will be prompted for one (with echoing turned off).

4.1.14 Unbind

```
unbind [-noquit]
```

This is used to break the connection to a DSA. If `-noquit` is specified, the DISH process does not die (This is used to maintain the cache).

To be "friendly" to other users, you should unbind when you have finished using the directory — put an `unbind` in your *.logout* file if you use DISH from the shell.

4.1.15 FRED

```
fred [-display <name>]
     [-dm2dn [-list] [-phone] [-photo] <domain-or-mailbox>]
     [-expand [-full] <DN>]
     [-ufn [-list,][-mailbox,][-phone,]
           [-photo,][-options %x,]<name...>]
     [-ufnrc <list...>]
```

This is used by the FRED program to provide some functions which would otherwise require complicated interactions between FRED and DISH.

`-display` is used to set the address of an X Window System[1] display.

`-dm2dn` is used to map a domain name from the DNS or a mailbox into a distinguished name in the directory. The `-list` suboption returns a list of possible matches, `-phone` returns the phone number attribute of the matched entry, and `-photo` returns the photo attribute of the matched entry.

`-expand` is used to see if a node has children.

`-ufn` is used to perform a user-friendly naming match. The `-list` suboption returns a list of possible matches, `-mailbox` returns the mailbox attribute of the matched entry, `-phone` returns the phone number attribute of the matched entry, `-photo` returns the photo attribute of the matched entry, and `-options` is used to customise the behavior of the matching algorithm.

`-ufnrc` is used to set the search list for user-friendly naming.

4.2 Sequences

Results returned by *search* and *list* operations may be long, and for this reason have a reference number printed beside them. The reference number can be used as the "object" in any of the calls to the directory. Thus

```
showentry 6
```

shows the entry labelled with the sequence number 6 by a previous *search* or *list* operation.

When you come into DISH by default, the resultant numbers of list and search operations are added to a sequence called "default".

Two flags

[1] The X Window System is a trademark of the Massachusettes Institute of Technology