```
masterDSA, slaveDSA, treeStructure
```

defines `quipuObject` to be an objectClass represented by the oid "quipuObjectClass.2". It is a *subclass* of the object class `top`. An entry having this object class *must* have the attribute `acl`, and *may* have the attributes `masterDSA`, `slaveDSA` and `treeStructure` (Note: The similarity between this definition and the ASN.1 definition of a `quipuObject` in Appendix D).

To allow for definitions of "attribute sets", there is a simple macro facility provided, for example, using the macro

```
localeAttributeSet = localityName, streetAddress ...
```

every occurrence of `localeAttributeSet` will be replaced by the right-hand side expression.

With the definition of the "string" form of the oid it is also possible to specify *one* abbreviation for the name. The standard tables use the following abbreviations:

c	countryName
o	organizationName
ou	organizationalUnitName
cn	commonName
co	friendlyCountryName

The abbreviated name may be used anywhere the full name would be allowed.

Every entry stored in the QUIPU DSA is checked against these tables to see if the attributes supplied are allowed in the entry, to make sure the mandatory attributes are present, and the attributes themselves have the correct syntax.

If you wish to add you own definitions to the tables you should add them to the files *isode/dsap/oidtable.at.local*, *isode/dsap/oidtable.oc.local* and *isode/dsap/oidtable.gen.local* in that way, if a new set of tables is issued, your local entries will be preserved.

14.12 More Help Installing QUIPU

Contact "QUIPU-support". The mail address is given in Section 2.6.

Chapter 15

Security Management

15.1 QUIPU Userid

The DSA should not run under the userid of the super-user (root). The reason for this is that no application should be run as root unless it is absolutely necessary, just in case an undetected bug causes it to malfunction. (A process running as an ordinary user will be halted if it starts to misbehave; one running as root might carry on and delete important files, for example).

Ideally, the DSA should be assigned a special userid to run under, called, for example, "quipu". If this is not possible, it should run under the userid of the DSA manager. Whichever option is taken, we shall henceforth refer to the selected userid as "quipu" in the documentation.

15.2 Shared Libraries

If using QUIPU, in particular the DUAs, compiled which shared libraries (under SunOS-4) care need to be taken when installing. The ISODE Makefiles use *-L $(TOPDIR) -ldsap*. *TOPDIR* will be defined as something like "../..". Normally, when installed, there will not be a *libdsap* in the "../.." directory so there is no problem. However if a user were to install their own *libdsap* in the "../.." directory there are potential security problems.

The problem can be solved by relinking binaries without the *-L $(TOPDIR)* switch, once the new libraries are installed. We are looking at ways of solving this within the ISODE Makefile.

15.3 File Permissions

The EDB files are where QUIPU stores the contents of the directory. As these files potentially contain sensitive information, they should be readable and writable only by the QUIPU userid.

There are two log files: one for debugging information and one for audit. Both of these files should be readable and writable only by the QUIPU userid. There is a potential problem here, in that ISODE's logging code usually makes log files writable by everybody. To be on the safe side, make a separate directory for QUIPU's logs (*/etc/isode/quipu-logs*, for example). Make this directory only accessible by the QUIPU userid, and change the tailoring parameter *logdir* so that the logs are written there.

15.4 Discretionary Access Control

The principles of discretionary access control are explained in the Chapter 11 of this manual. This section gives guidelines on setting the access control lists for entries in the DIT.

15.4.1 What Must be Publicly Readable

Some attributes are used by the directory itself for routing and other purposes. If these attributes are not publicly readable (and hence readable by all DSAs) then the directory's internal communications may fail. If a DUA gives messages such as "Unavailable" this is one possible cause. Note that there are many other possible causes of such failures; Network congestion or a machine being down is the most likely explanation.

The following attributes ought to be publicly readable:

- masterDSA.
- slaveDSA.
- presentationAddress.
- userCertificate.
- treeStructure.

The *userPassword* attribute ought to be comparable by everybody, but not readable (unless the entry is a spot-shadowed DSA entry, in which case the userPassword must be publically readable, see Section 14.4).

15.5 Audit

15.5.1 Enabling Auditing

The *stats* parameter in the *quiputailor* file controls how much audit information is kept. It is advisable to enable recording of audit events at the *notice* level. More detailed information is given at the *trace* level. Both are enabled by default.

15.5.2 Relating Events to Users

Most events in the usage log contain an *association descriptor* instead of the name of the user who caused the event. An association descriptor is a (small) number which identifies a connection to QUIPU. (It is rather like knowing which terminal line a command came in on.) To discover the user name, it is necessary to scan back through the log to find the record for the start of the association. This will contain the name of the user and how he or she became authenticated.

15.5.3 Format of Audit Records

Each record in the log file is formatted as follows:

```
<AuditRecord> ::= <month> "/" <day> <time> <process>
                  <pid> "(" <userid> ")" <Event>
```

`time`: The time of the event in hh:mm:ss format.

`process`: The name of the program ("quipu").

`pid`: The process id of the DSA.

`userid`: The id of the user who started the DSA running. It is *not* the id of the DUA which caused the event!

`Event:` is the rest of the message. The following sections describe most of the common messages.

15.5.4 Start of an Association

```
<BindEvent> ::= "Bind" "(" <Integer> ")"
                "(" <AuthType> ")"
                ":" <DN>
<AuthType> ::= "none" | "simple" | "protected" |
               "strong" | "noauth"
```

For example:

```
Bind (4) (simple): c=GB@o=University College
         London@ou=Computer Science@cn=Steve Kille
```

This means that Steve Kille has started using association descriptor 4, and proved his identity using simple authentication (i.e., a password).

15.5.5 End of an Association

```
<UnbindEvent> ::= "Unbind" "(" <Integer> ")" <WhoBy>
      ":" <DN> <WhoBy> ::= "(by this)" | "(by that)"
```

For example:

```
Unbind (4) (by that): c=GB@o=University College
           London@ou=Computer Science@cn=Steve Kille
```

This means that Steve Kille's DUA has disconnected from the DSA, and descriptor 4 is left free for use by someone else. The "(by that)" means that the DUA, rather than this DSA, decided to close the connection.

15.5.6 DAP Operation

```
<DAPOperationEvent> ::= <OpType> "(" <Integer> ")"
                            ":" <DN>
<OpType> ::= "Read" | "Search" | "List" | "Compare" |
             "Add" | "Remove" | "Modify" | "ModifyRDN" |
             "Getedb"
```

For example:

```
Read (4): c=gb@o=Nottingham University
```

This means that whoever is using association 4 (Steve Kille in this example) has read the entry for Nottingham University.

15.5.7 DAP Result

```
<DAPResultEvent> ::= "Result sent" "(" <Integer> ")"
<DAPErrorEvent>  ::= "Error sent"  "(" <Integer> ")"
```

Each operation will normally be answered by either a result or an error.

15.5.8 Chaining

```
<ChainingEvent> ::= "Chain" "(" <Integer> "):" <OpType>
```

This means that the DSA has decided to contact another DSA in order to perform an operation received previously.

Normally, nothing is logged when a chain operation is returned. However, by increasing the logging level to "LLOG_DEBUG" a *Result Recieved* is logged.

15.5.9 Updates

```
<UpdateEvent> ::= "Slave update" <DN> |
                  "Shadow update" <DN>
```

These indicate an EDB file or spot shadow entry have been updated.

15.5.10 Other Events

There are a few other less common messages that can be written to the audit log. The text messages should be self-explanatory.

15.5.11 Processing the Log Files

A script called *dsastats* may be used to process the log file and produce a report summarising the usage of a DSA. The report shows the following:

- The operations performed by the DSA
- Who has accessed the DSA
- Which parts of the DIT have been accessed

The report summarises the calls received by the DSA according to the degree of authentication. This analysis is given for remote and local access. The report also tries to distinguish between system usage, which includes QUIPU getedb operations, DSA probing and testing by directory system and interface developers, and real usage. The ability to make this distinction rests on user names being supplied in bind requests. A large percentage of binds are currently anonymous — it is hoped that this report encourages directory administrators to install systems such that the provision of user names becomes the norm.

If the `-summary` flag is used when the script is run, a more concise report is produced.

Configuring dsastats

The usefulness of the report relies on accurate configuration. The key configuration details to be noted are:

`Local Organisation:` Analysis of which parts of the DIT have been accessed is given by organisational unit for the pre-configured local organisation. (Note that this does not apply if the -summary flag is used.) For all other parts of the DIT, analysis is restricted to being by organisation.

`Local addresses:` A file should be set up (a skeleton is provided) which contains the leading substrings from local DUA and DSA addresses, as they appear in the quipu.log file. This file, which is called *quipulocaladds,* might look something like the following:

```
LOCAL-ETHER
Janet=000012345678
Internet=128.9
# file must end with this line
```

`Filtering out system usage:` A file should be set up (again a skeleton is provided) which contains the distinguished names of system users. This will probably include the names of directory software developers, directory administrators and dsa probes. Distinguished names can be specified case-independently. The file, which is called *quiputechusers*, might look something like the following:

```
c=GB@o=X-Tel Services Ltd@cn=Camayoc
c=GB@o=University College London@cn=DSA Probe
# File must end with this line
```

Running the script

The script can be configured to find the tailor files and log files. In this case, the script can be run by something like the following:

```
dsastats >usage.report
```

The above will not be possible if you run more than 1 DSA. The recommendation is then that each DSA logs to a file with a name of the form dsaname.usage. Usage would now be:

```
dsastats llama.usage >llama.report
```

A report produced by this script should resemble the example given in Figure 15.1.

To produce a more concise report, specify the -summary (or -s) flag.

```
dsastats -s llama.usage >llama.summary
```

```
Summary of calls to DSA <giant armadillo>
From  2:48:46 on 04 feb to 13:24:09 on 07 feb

No. of binds                      local  remote

Anonymous                             2       8
Unauth name DAP                       1       2
Unauth name DSP                      11     137
Simple                                8      25
Protected                             0       0

No. of operations

Reads                                 2     136
Compares                             28     141
Lists                                 0       4
Searches                             41     214
Modifies                              0       0
ModifyRDNs                            0       0
Adds                                  0       0
GetEDBs                                     363


System usage (calls received)

Binds by Directory technicians              169
Reads of DSA entries                          4
Other ops on DSA entries                      4
Getedb operations                           363
Spot shadows                                  0

Who has used the directory?
*Real* usage by organisation
No. users No. binds
        1       145    anonymous
        1         5    c=gb, o=x-tel services ltd
        ...

Which parts of the Directory have been accessed - real usage?
No. ops  Subtree
Local subtree
     67  c=gb@o=university college london
     14  c=gb@o=university college london@ou=computer science
      1  c=gb@o=university college london@ou=genetics and biometry
      1  c=gb@o=university college london@ou=mathematics
      ...

Other parts of the DIT
      0  root
     66  c=gb
      1  c=gb@o=brunel university
      1  c=gb@o=edinburgh university
      1  c=gb@o=glasgow university
      ...
```