

## WHALE FUNCTION

Melinda Varian

Office of Information Technology  
Princeton University  
87 Prospect Avenue  
Princeton, NJ 08544 USA

—.—  
Email: [melinda@princeton.edu](mailto:melinda@princeton.edu)  
Web: <http://pucc.princeton.edu/~melinda/>  
Telephone: 1-609-258-6016

February 02, 2005

### I. INTRODUCTION

This document describes a portion of the function implemented in Whale, the program used to populate account information for OIT's Active Directory, LDAP, IMAP, Webmail, Exchange, unix, P-synch, Kerberos, and mainframe systems. The document does not discuss the portions of Whale that are used solely for supporting the mainframe systems, VM/CMS and MVS, nor does it discuss the logic of CARPROC, the program that drives most Whale updates.

Whale stores information about active users directly in the main LDAP directory. It signals processes running on other systems to update other databases, including the NIS maps, Active Directory, and the Kerberos database.

Whale also maintains its own database (the "\$File"). Some of the information in the Whale database is not currently stored in LDAP or AD or the NIS map and will need to be stored somewhere by the Whale replacement. These items of information are referred to in this document as the "meta-data". They include all of the information needed for billing for system use. They also include historical records for inactive or obsolete users who no longer have LDAP entries. (Retaining these records allows permanent information to be retrieved for returning users, such as their unix uid and gid numbers.)

In addition, Whale writes a permanent transaction file detailing each transaction. Tools exist to archive and scan the transaction data.

Whale signals updates for AD by issuing a CGI GET for `/apps/UserAdd/UserAdd.aspx` on winadmin. In each case a parameter string is specified with an action and a list of other arguments. The CGI returns an indication of success or failure.

Rules for LDAP attribute values and other data maintained by Whale are given in the appendix. Variable attribute values are specified in boldface in this document. The LDAP attributes maintained by Whale have distinguished names in one of two forms:

"cn=**cn**,o=Princeton University,c=US" or "cn=**cn (mail)**,o=Princeton University,c=US"

The former is referred to as a "base entry".

## II. ADDING A NETID

Whale first becomes aware of a person or other University entity that should be given access to campus computer systems when it receives a command to add a netid. The Whale operator or the program driving Whale (such as CARPROC) selects the netid (username) and provides information about the person or entity represented by that netid.

### Determining Whether the Netid Can Be Added

Whale operators are encouraged to issue the INUSE command before trying to add a new netid. INUSE scans two meta-data files (recent.userids and alumuids.file) containing, respectively, a historical list of netids on the OIT systems and a list of usernames in the alumni system. With this information, the Whale operator can select a netid that was not issued in the past (rather than one that has been used, even though it may no longer be in use).

When Whale receives a command to add a netid, it must first determine whether the netid conflicts with one currently in LDAP. It prompts for the universityid number and Social Security number (SSN) of the person being added (or of the person responsible for the non-human entity being added). With this information, it does a series of tests to determine whether the proposed netid would conflict with an existing LDAP entry.

The LDAP entries for computer/email users contain a single uid attribute, which specifies the person's netid, and one or more campusid attributes. There is always a campusid attribute with the same value as the uid attribute (the netid), but there may be other campusid attributes corresponding to the person's usernames on other campus systems. Multiple LDAP entries can have the same SSN attribute. There can be no more than one LDAP entry with a given universityid attribute; any other entries for the same person contain a universityid attribute with a dummy value (*i.e.*, 09nnnnnnn) and a universityidref attribute with the real universityid number.

Searches are done for all records with a campusid equal to the proposed netid, with a universityid equal to the specified universityid, and an SSN equal to the specified SSN. (SSN matches are ignored when the universityid attribute in the LDAP entry holds a dummy universityid number.) Based on the results of these three searches, the logic for deciding whether a new netid can be added is:

- If no match is found in any of the three searches (campusid, universityid, SSN), the netid can be added.
- If an existing LDAP entry has a campusid that matches the proposed netid but a different uid, the netid is already in use and cannot be added.
- If an LDAP entry with the specified universityid is found but it has a different SSN than the one specified and neither of the SSNs is a dummy SSN (*i.e.*, 999nnnnnn), the proposed netid cannot be added.
- If an LDAP entry with the specified SSN is found but it has a different universityid than the one specified and neither of the universityids is a dummy, the proposed netid cannot be added.
- If two or more LDAP entries have the specified campusid or universityid, the proposed netid cannot be added. (This should never happen.)

- If the three searches find one matching LDAP entry and the uid in that LDAP entry is the same as the proposed netid, the netid add request is treated like a request to update that LDAP entry.
- If the SSN and universityid searches find one matching LDAP entry that has no uid attribute and the campusid search does not find a match, the netid add request is treated like a request to update that incomplete LDAP entry. (This should never happen.)
- If a match is found in more than one of the three searches and the searches that find a match do not all find the *same* LDAP entry or if an LDAP entry that matches on the SSN or the universityid has a uid that is not the same as the proposed netid, a conflict exists. The Whale operator is shown the netid of the conflicting LDAP entry and is presented with a prompt to decide which of these three cases applies:
  1. *Netid Swap*: The new netid will become this person's only netid, replacing the old one after a few days.
  2. *Alter-ego Netid*: The new netid will be an additional personal netid for this person. (This option is available only to Whale super-users.)
  3. *Non-personal Netid*: The new netid will be a non-personal netid for which this person will be responsible for the time being.
- If an Xalias LDAP entry (ou=Xaliases,o=Princeton University,c=US) with a matching campusid is found, there may also be a conflict. The proposed netid can be added to replace the Xalias LDAP entry unless the Xalias entry has an mailboxalternate attribute, in which case it is the entry for a mailing list and must be left intact.

### **Adding/Updating a Netid When There is No Conflict**

1. Assign a dummy universityid for the user if the real universityid is not known.
2. If an existing LDAP entry is being updated as a result of the request to add a netid:
  - Make sure that there are campusid and emailrewrite attributes containing the netid.
  - Make sure that the following attributes contain current values:
    - displayname
    - ou
    - pdisplayname
    - puhomedepartmentnumber
    - puimapfilesystemdept
    - pupwstring
    - pupwstringtype
    - pstatus
    - puunixfilesystemdept
    - uid
  - Replace a dummy universityid with the real one, if known.
  - Replace a dummy or non-existent SSN with the real one, if known. (This may require an update to the pupwstring attribute, which should always be based on the SSN rather than the birthday if the SSN is known.)

3. Create a base LDAP entry if none exists by populating the following attributes in an entry with a distinguished name of the format “cn=**cn**,o=Princeton University,c=US”:

```

.   campusid: campusid
.   displayname: displayname
.   givenname: givenname                                (humans only)
.   objectclass: inetOrgPerson
.   objectclass: inetUser
.   objectclass: ipUser
.   objectclass: organizationalPerson
.   objectclass: person
.   objectclass: puPerson
.   objectclass: top
.   ou: ou
.   pudisplayname: pudisplayname
.   puhomedepartmentnumber: puhomedepartmentnumber
.   puimapfilesystemdept: puimapfilesystemdept         (optional)
.   pupwstring: pupwstring                             (optional)
.   pupwstringtype: pupwstringtype                     (optional)
.   pustatus: pustatus
.   puunixfilesystemdept: puunixfilesystemdept        (optional)
.   sn: sn
.   ssn: ssn                                           (optional)
.   street: street
.   telephonenumber: telephonenumber                 (optional)
.   uid: uid
.   universityembref: universityembref                (alter-ego entries only)
.   universityid: universityid
.   universityidref: universityidref                  (non-primary entries only)

```

If the LDAP add completes with the message “ldap\_add: Already exists”, the LDAP directory already has a cn attribute with the proposed value. Tell the Whale operator to use a different value for the name of the person or entity being added.

4. Signal kprime to add a Kerberos principal for this netid. (A message “create **uid**” is sent via TCP to port 756 on kprime (or to port 757 on csgtest2 for testing Whale).) Note that Whale has an option not to signal kprime; this is used when large numbers of netids are being added. Kprime will find all new users in a once-a-day run anyhow.

### Special Processing When There is a Netid Conflict

#### Existing Xalias:

In the case of an existing non-mailinglist Xalias entry, delete the Xalias entry and remember the captured mailbox attribute. When creating the new base LDAP entry, set the mailbox attribute to the saved value, unless running under CARPROC. (The thinking here is that if a human Whale operator wants to replace an Xalias entry, it is very likely a temporary re-add of the netid to allow files to be recovered. However, if CARPROC is driving Whale, it is more likely to be re-adding a user who has come back to the University, in which case the Xalias entry’s mail forwarding information should not be preserved.)

**Netid Swap:**

1. Update the old LDAP entry:
  - Remove the SSN.
  - Remove the pupwstring and pupwstringtype.
  - Set the universityidref attribute to the real universityid.
  - Set the universityembref (emailbox reference) attribute to the real universityid.
  - Replace the universityid with a dummy universityid.
2. Update meta-data for the old netid:
  - Set the netid termination date to two weeks in the future.
3. Add the new netid to LDAP (as above) with the real universityid and SSN.

**Alter-ego Netid:**

If the Whale user is privileged, add the LDAP entry for the new netid:

- Assign a dummy universityid for the universityid attribute.
- Use the real universityid for the universityidref attribute.
- Use the real universityid for the universityembref attribute.
- Use the real SSN for the ssn attribute.

**Non-personal Netid:**

1. Force the value of the pustatus attribute to be one of those appropriate for non-human entities (adm, cwk, lsv, org, oth, out, svm, or svr).
2. Ask the Whale operator to provide a “user name” that is a description of the function of the netid, rather than the name of the person who is to be responsible for this non-personal netid for the nonce.
3. Add the LDAP entry for the new netid:
  - Assign a dummy universityid for the universityid attribute.
  - Use the real universityid for the universityidref attribute.
  - Do not specify an ssn attribute.
  - Do not specify pupwstring or pupwstringtype attributes. (Starter passwords for non-personal netids are determined by attributes in the referenced LDAP entry.)

**Meta-data Stored for Netids**

A netid can have a “termination date”, which causes it to be deactivated once that date is past.

Whenever a netid is removed from LDAP, the data necessary to recreate it are stored in the meta-data.

### III. ADDING A DEFAULT ACCOUNT

Once a default account (a “.STUDY account”) is added for a netid, the user can use LDAP to authenticate to various campus services. In most cases (currently everybody other than trustees, employees of the Princeton University Press, and participants in the Community Auditor Program), the user also has access to IMAP for email and is enrolled in the Princeton AD domain. Everybody enrolled in the AD domain must be given access to the public unix systems (which host the Samba shares and the Web pages). However, the unix login shell is set to /bin/nologin for all new users other than undergraduates. (Anyone who has a unix account can change his shell to a real value via a Web page.)

There must be an LDAP entry with a uid equal to the netid before the default account can be added.

The LDAP entry is updated to enable access to the appropriate OIT systems:

- If the user should have access to email services and the distinguished name does not contain an at-sign (that is, it does not contain an email address), an LDAP modify rdn command is issued to change the dn to the form “cn=**cn (mail)**,o=Princeton University,c=US”. Because the modify rdn command deletes the cn attribute that has a value of just the user’s name, that must be re-added. Because changing the rdn makes the paburi attribute and the user’s Private Address Book entries obsolete, they must be changed as specified in Section IX below.

If this is a new email user (the user should have email privileges but the dn did not contain an at-sign or the LDAP entry did not contain an mailbox attribute), the LDAP entry must be updated to enable IMAP mail use:

— The following standard attributes are set:

```

datasource: iPlanet Messaging Server 5.0 Admin Console
emailrewrite: emailrewrite
inetuserstatus: active
mailallowedserviceaccess: +all:*
maildeliveryoption: mailbox
mailhost: imap1.Princeton.EDU      (for undergraduates)
mailhost: imap2.Princeton.EDU    (for all others)
mailuserstatus: active
objectclass: inetLocalMailRecipient
objectclass: inetMailUser
objectclass: nsManagedPerson
objectclass: nsMessagingServerUser
objectclass: userPresenceProfile
preferredlanguage: en
pumailstore: IMAPVP1             (for undergraduates)
pumailstore: IMAPVP2             (for all others)

```

If the distinguished name does contain an email address and the user should have email privileges, a check is made to assure that the standard mailalternateaddress attributes are set (they may not be if the user was in LDAP just to get mail-forwarding to a non-Princeton email address).

- The mailquota attribute is set to the default IMAP mail quota or to -1 if the user's department has a private IMAP store, in which case the memberofmanagedgroup attribute is also set.
- If there is no mail attribute in the LDAP entry, the default attribute is added.
- If there are no mailalternateaddress attributes in the LDAP entry, the three default values are added.
- If there is no mailbox attribute in the LDAP entry, the default value is added.
- If there is a mailmessagestore attribute in the LDAP entry, it is deleted. If the user's department has a private IMAP mailstore, a new mailmessagestore attribute is added specifying that mailstore (which is determined by examining the department record in Whale's meta-data); otherwise, the mailservers are queried to determine whether they have a mailstore for the user and, if not, which mailstore should be set for the user. A query ("uid: **uid**") is first sent via TCP to port 8101 (imap1) or 8102 (imap2) on the "wrong" mailserver (imap2 for undergraduates and imap1 for all others) in case the user already has a mailstore there. If the response contains a value for "mailmessagestorecurrentlyinuse", the mailmessagestore attribute is set to the specified value and the mailhost, pumailstore, and mailalternateaddress attributes are set to specify that mailserver. If the user is not found on the wrong mailserver, a message is sent via TCP to port 8101 or 8102 on the "right" mailserver (imap1 for undergraduates or imap2 for all others) to request a mailstore. The response is "mailmessagestorecurrentlyinuse" or "mailmessagestorefornewaccounts" followed by the value to use for the mailmessagestore. (If the mailserver doesn't reply, the mailmessagestore is set to "primary".)
- If either response includes a value for "mailmessagestorecurrentlyinuse", the user is one who was deleted and is being re-added. Although his LDAP directory entry was deleted at some point, his mail had not yet been deleted. Therefore, his mailmessagestore, mailhost, mailalternateaddress, and pumailstore attributes must be set correctly to allow him to retrieve his existing mail.
- If the user is to be allowed access to the public unix systems, the LDAP entry must be updated with several parameters. An objectclass attribute is added with the value "posixAccount" if there is none, and the following attributes are added or replaced:
  - gecos
  - gidnumber
  - homedirectory
  - loginshell
  - pumacosxhomedirectory
  - pumacosxcifsmount
  - puunixquota
  - uidnumber

Once LDAP has been updated with the unix information, a signal is sent to nismaster.Princeton.EDU to indicate that there is a new unix user. The port number is 2823 for production and 2923 for testing. The message is "create **uid**" followed by a linefeed. The response begins with a banner indicating a successful connection ("CRUD Version x.y ready"); that is followed by "OK account **uid** created" or an error message starting "ERR".

- Pupsynch attributes are added for all of the systems to which the netid is enabled; this allows the user to change the passwords on any of those systems using P-synch.
- The LDAP and P-synch passwords and their timestamp attributes are set (see the logic in Section VI. below). If a real AD password is being set, the pupwadmodifytimestamp attribute is also set. (Note that the AD password is not stored in the LDAP entry.)

Because Whale does not directly populate/depopulate AD, it must take care to determine whether AD already knows about the netid. (Giving AD a signal to add a user who already exists has the effect of re-establishing the AD account, which wipes out existing information.) Winadmin is queried as to the existence of a user by issuing a CGI GET for:

action=USEREXISTS&netid=**uid**

If this is a new AD user, an AD update is signalled by sending a (one-line) message to winadmin via CGI GET:

action=ADDWITHPASSWORD&netid=**uid**&fullname=**cn**&password=**plaintextpassword**  
&uaid=**universityid**&status=**ADstatus**&dept=**puhomedepartmentnumber**

The “action” above is “ADDWITHSSN” if the password is a starter password. (See Section VI.)

#### Meta-data Stored for Default Accounts

- Account deletion date (optional)
- Account registration date
- IMAP disk quota (optional)
- Unix disk quota (optional)

#### IV. ADDING A NON-DEFAULT ACCOUNT

Whale implements the concept of subaccounts in the form “netid.subid”. Two subid names are “reserved”, .STUDY and .EXCHANGE. .STUDY is used as the subid of the default account; .EXCHANGE is used as the subid of the main Exchange account. The default account must exist before any other accounts can be added. The value of “subid” for other accounts can be any convenient 1-8 character alphanumeric string meaningful to the account owner.

Non-default accounts exist for the purpose of adding disk quotas and providing information on how to charge for the added quotas. When a non-default account is added, Whale searches its meta-data for a record describing the University account that is to be charged. This must be present and must have a budget sufficient to cover the amount budgeted for the new user account.

When a non-default account is added, these LDAP attributes are modified, if appropriate:

- mailquota
- puunixquota

When a non-default account is added with unix quota specified, nismaster is signaled that there has been a change in the LDAP entry. The signal (to port 2823) is “update **uid**” followed by a

| linefeed. The response is the usual banner followed by “OK account **uid** updated” or an error  
| message beginning “ERR”.

. When a non-default account is added with Exchange quota specified, winadmin is signaled to  
. increase the Exchange mailbox size by CGI GET:

. action=EXCHANGEQUOTA&netid=**uid**&increment=kilobytes

. Non-default accounts are not allowed to have Exchange quota unless a .EXCHANGE subid has  
. already been set up, but before issuing the EXCHANGEQUOTA call, Whale issues a  
. MAILBOXEXISTS call to confirm that AD agrees that this is an Exchange user.

### Meta-data Stored for Non-default Accounts

- Account deletion date (optional)
- Account registration date
- Dollar amount budgeted
- Dollar amount used
- Billed University account
- Exchange disk quota (optional)
- Exchange disk quota charge date (optional)
- IMAP disk quota (optional)
- IMAP disk quota charge date (optional)
- Unix disk quota (optional)
- Unix disk quota charge date (optional)

## V. ADDING AN EXCHANGE ACCOUNT

The default account must exist before a .EXCHANGE subid can be added. Although subids other than .EXCHANGE can be responsible for Exchange quotas, the .EXCHANGE subid must exist for all Exchange users.

. When a .EXCHANGE subid is added for a user, Whale signals winadmin to establish the  
. Exchange mailbox by CGI GET:

. action=CREATEMAILBOX&netid=**uid**&exchangeserver=servername&storagegroup=groupname  
. &store=storename&quota=kilobytes

. Whale then updates the user’s LDAP entry with two puexchangeservice attributes (“mail” and  
. “calendar”), a puexchangeenabled attribute (a zulu timestamp), and a maildeliveryoption attribute  
. (“forward”). In addition, it replaces the mailbox and mailforwardingaddress attributes with the  
. value **uid**@exchange.Princeton.EDU.

. The .EXCHANGE subid records the Exchange quota and the University account number (project  
. grant number) to which the Exchange quota is to be billed. When a new Exchange account is  
. added, Whale sets the Exchange quota equal to the user’s IMAP quota to ensure successful  
. migration of existing IMAP mail to Exchange.

Whale's meta-data must contain a record for the specified University account number, which must have sufficient funds to cover the budget for the .EXCHANGE account. Whale's departmental meta-data must contain a record for the user's home department and that record may specify the Exchange server, Exchange storagegroup, and Exchange mailstore to be used for new Exchange users for that department. If the department record does not contain Exchange information, the default values specified in the record for Department 000 are used.

### Meta-data Stored for Exchange Accounts

- Account deletion date (optional)
- Account registration date
- Dollar amount budgeted
- Billed University account
- Exchange disk quota
- Exchange disk quota charge date

## VI. SETTING AND ALTERING PASSWORDS

The Whale operator may set or change a user's passwords on one or more kinds of systems. When a default account is being added, a password is set on all of the systems the user is entitled to use (although this "password" may be simply an indication that the user must set a real password using P-synch). When the Whale operator issues a password change command, a prompt is given to allow the choice of one or more of the AD, LDAP (IMAP), P-synch, and Kerberos passwords.

### Validating the Password

**User-selected Password:** The Whale operator can specify a real password to be used on one or more systems, but this is discouraged except in emergencies. The preferred procedure is for the Whale operator to set the P-synch password and encourage the user to use P-synch to set the other passwords.

The proposed password is subjected to a number of tests:

- The string specified is 6-8 characters chosen from the numerals, upper- and lower-case alphabets, and these special characters: @ ( ) \$ ! - % \_ . ` ' ~ { }
- The string is not all numeric.
- If the AD password is being set, the string includes at least one character from three of the four sets, upper-case alphabets, lower-case alphabets, numerals, and special characters.
- Unless the Whale operator has Whale super-user privileges and has used the "loose password" option, the string passes the cracklib rules.

If any of these tests fails, the Whale operator is given a message explaining why the proposed password is unacceptable and is prompted for another password. Once an acceptable password is entered, Whale encrypts it using a standard unix crypt routine with a "random" seed to produce a 13-character *cryptstring*. The plaintext password is not stored on disk anywhere.

**Default Starter Password (“Password Reset”):** If the Whale operator specifies a password of “reset”, Whale calls the WHALGENP script to return a password consisting of four “random” alphanumeric characters followed by either the last four digits of the user’s non-dummy SSN or the MMDD value of the user’s birthday. (If the LDAP entry for the netid has a universityidref attribute, the SSN or birthday is obtained from the referenced LDAP entry.) WHALGENP enforces the rules for AD passwords by making certain that there are at least a mixture of upper- and lower-case alphabetic characters and numerals in the resulting string. The first four characters generated by WHALGENP are reported to the Whale operator along with the rule for determining the other four characters. The password is flagged as being a “starter password”, one that must be changed following the first use.

**CARPROC-selected Starter Password:** When Whale is being operated by CARPROC, the password it specifies is accepted without being subjected to the guessability tests, but it is flagged as a starter password. (CARPROC has selected the password by invoking WHALGENP itself, so the password is a safe one to use as a starter password.)

**Whale-generated Password:** If the Whale operator specifies a password of “genpass” (typically because the user’s SSN and birthday are unknown), Whale invokes the WHALGENP script to generate a “random” eight-character password that will pass the rules for AD passwords. This password is not currently flagged as a starter password.

### Updating the LDAP Entry When a Password is Changed

Whale first calculates a “zulu timestamp” to use to timestamp password changes in the LDAP entry. (This is the Gregorian timestamp in UTC, *i.e.*, YYYYMMDDhhmmssZ.) The password modification timestamp attributes are set only when a real password is being set, and they are removed when a starter password is being set. (Note that the P-synch password is always considered to be a real password.)

The userpassword and pupsynchpw attributes are updated, respectively, if the LDAP or P-synch passwords are being set. The attribute value for userpassword contains “init” if a starter password is being set. Unix and IMAP recognize this as an indication that the user cannot login until a real password is set using P-synch. The attribute value for pupsynchpw is the unix crypt of the password, whether or not it is a starter password. The attribute value for userpassword contains the unix crypt of the password if it is a real password.

The pupwXXmodifytimestamp attributes are set or removed as appropriate depending on whether the LDAP, P-synch, and/or AD passwords are being set to real or starter passwords.

| The unix systems always check the LDAP userpassword attribute when a user is attempting to  
| login, so no notification of a password change need be sent to nismaster.

### Updating AD When a Password is Changed

To set an AD password, a message in one of the following two formats is sent to winadmin via a CGI GET (for setting a real and a starter password, respectively):

action=PASSWORD&netid=**uid**&password=**plaintextpassword**  
action=RESET&netid=**uid**&password=**plaintextpassword**

If this operation returns a “doesn’t exist” response, a forced add of the user to AD is done as described in Section III. above.

### Updating Kerberos When a Password is Reset

Whale is not used to set new Kerberos passwords, but it can be used to signal that the existing Kerberos password should be discarded (thus requiring the user to establish a new one using P-synch). A message in the form “resetpw **uid**” is sent via TCP to port 756 on kprime (or to port 757 on csgtest2 for testing Whale).

## VII. ALTERING NETID DATA FIELDS

### “Alter Ego” State:

- If the netid is no longer an alter ego, delete the universityembref attribute.
- If the netid is becoming an alter ego, ensure that there is a universityidref attribute and set the universityembref attribute to the same value.

### Birthdate:

- If the birthday is being deleted and the pupwstringtype is “B”, remove the pupwstring and pupwstringtype attributes.
- If the birthday is being set and the pupwstringtype is not “S”, set the pupwstring to the birthday (MMDD) and set the pupwstringtype to “B”.

**Gecos Fields (Name, Address, and Phone):** Update the gecos attribute in the LDAP entry and send an “update **uid**” signal to nismaster.

**Unix Uidnumber, Gidnumber, or Loginshell:** Update the appropriate attribute in the LDAP entry and send an “update **uid**” signal to nismaster.

**Obsolete and Inactive Flags:** If these flags (in the meta-data) are altered so that neither is on, the netid is re-added to LDAP as though it were a new netid. Only Whale super-users are allowed to turn off the “obsolete” flag.

**Private Departmental Filesystems:** When a user is changed to or from a private IMAP mailstore, a message “storeNN” is sent via TCP to port 8301 on imap1 (undergraduates) or port 8302 on imap2 (all others) to cause the user’s mail to be moved to the specified mailstore. (“NN” is the new mailstore number. For the purposes of testing Whale, ports 8311 and 8312 are used.) An “OK” response indicates that the move was successful. If the user is being moved from a public to a private mailstore, his LDAP entry must be updated to set mailquota to “-1” and to set the correct values for the puimapfilesystemdept and memberofmanagedgroup attributes. If he is being moved from a private mailstore to a public one, his mailquota must be set to the default value and those two attributes must be deleted.

When a user is changed to or from a private unix filesystem, Martin Harriss must be notified to perform a manual update. The puunixquota attribute should be set to 0 (if going to a private filesystem) or to the default value. The puunixfilesystemdept attribute is set or deleted, as appropriate.

**SSN:**

- If an SSN is being added or altered, replace the `ssn`, `pupwstring`, and `pupwstringtype` attributes.
- If the SSN is being cleared, remove the `ssn` attribute and update the `pupwstring` and `pupwstringtype` attributes to reflect the birthday (or remove them if the birthday is not known).

**Status or Department Number:**

- Signal an AD update by sending a message to winadmin via CGI GET:  
`action=STATUSCHANGE&netid=uid&status=ADstatus&dept=puhomedepartmentnumber`
- Update the LDAP entry to reflect new values for the `pustatus`, `puhomedepartmentnumber`, and `ou` attributes.

**Universityid:**

- If the `universityid` is being deleted, assign a dummy one.
- Update the `universityid` attribute in the LDAP entry.

**Universityid Reference:**

- If a `universityidref` is being cleared or set, delete any `sealso` and `universityembref` attributes, as well as the `universityidref` attribute.
- If a new `universityidref` is being set, update it and replace the `universityembref` attribute if there is one.

Only Whale super-users are allowed to specify a `universityidref` attribute when adding a `netid`, but all Whale users can alter the `universityidref`.

**User's Name:**

- Signal an AD update by sending a message to winadmin via CGI GET:  
`action=NAMECHANGE&netid=uid&fullname=cn`
- Modify the distinguished name of the LDAP entry to contain the new `cn`. (Note that if the new and old `dn` differ only in case, the `modrdn` operation will fail; in that case, do two `modrdn` operations, using a dummy `cn` for the first one.)
- Replace the `cn` attribute if necessary. (The `modrdn` operation replaces the `dn` and the `cn` that matches it up to the comma, but it may leave another `cn` unchanged.)
- : • Replace any `paburi` attribute and modify the user's Private Address Book entries as specified in Section IX.
- : • Delete any `givenname` attribute and add the new one, if appropriate.
- Replace the `sn` attribute.
- Replace the `displayname` and `pudisplayname` attributes.

## VIII. ALTERING ACCOUNT DATA FIELDS

**Disk quotas (Exchange, IMAP, or unix):** When disk quotas are changed, the appropriate LDAP attribute is modified but the meta-data are also modified to set the “last charged” date for that category of disk quota, so that the monthly billing run will charge for the new quota size only from that date. This has proved to be an unsatisfactory arrangement. More useful would be to keep a running total of megabyte-days for Exchange, IMAP, and unix quotas by University account number that could be queried and reset by the billing program. Obviously, this would require attributes containing the current quotas by University account number and a daily run to increment the megabyte-day counts.

When the unix quota is altered, Whale sends an “update **uid**” signal to port 2823 on nismaster after the LDAP entry has been updated.

When the Exchange quota is altered, Whale signals winadmin via CGI GET:

action=EXCHANGEQUOTA&netid=**uid**&increment=kilobytes

Before issuing that call, however, it issues a MAILBOXEXISTS query to confirm that AD agrees that this is an Exchange user.

When a request to increase an IMAP quota is entered, the appropriate IMAP server is queried to determine whether sufficient space is available to allow the request. The message “uid: **uid**” is sent to port 8121 on imap1.Princeton.EDU (for undergraduates) or to port 8122 on imap2.Princeton.EDU (for all others). That server returns a message indicating the mailstore and the message “maxmailquotaincrease: nnnnn” (in bytes).

**Public Unix Access:** If public unix access is being turned off for the default account, Whale removes these LDAP attributes:

- objectclass: posixAccount
- gecos
- gidnumber
- homedirectory
- loginshell
- puauthpassword
- pumacosxhomedirectory
- pumacosxcifsmount
- pupsynch: NIS
- uidnumber

Similarly, if public unix access is being turned on for the default account, Whale populates those attributes in the LDAP entry.

Once the LDAP entry has been modified, a signal is sent to nismaster. The message is “create **uid**” or “delete **uid**”, as appropriate.

**Windows Access:** Access to the Princeton AD domain can be enabled and disabled by altering the default account for a user. If it is being disabled, Whale sends a message to winadmin via CGI GET:

action=DELETE&netid=**uid**

It then deletes any pupsynch attribute with a value of “AD”.

If access is being enabled, an ADDWITHPASSWORD or ADDWITHSSN message is sent to winadmin (as in Section V.) and the LDAP entry is updated to include a pupsynch attribute with the value of “AD”. If the password is real, the LDAP entry is also updated to have a pupwadmodifytimestamp attribute.

. **Email Access:** Access to IMAP email can be enabled and disabled by altering the default  
: account for a user. If access is being disabled, Whale removes these LDAP attributes:

- . • objectclass: inetLocalMailRecipient
- . • objectclass: inetMailUser
- . • objectclass: nsManagedPerson
- . • objectclass: nsMessagingServerUser
- . • objectclass: userPresenceProfile
- . • datasource
- . • mailbox
- . • emailrewrite
- . • inetuserstatus
- . • mail
- . • mailallowedserviceaccess
- . • mailalternateaddress
- . • mailautoreplytext
- . • maildeliveryoption
- . • mailhost
- . • mailmessagestore
- . • mailquota
- . • mailsieverulesource
- . • mailuserstatus
- . • memberofmanagedgroup
- . • nswmextendeduserprefs
- . • preferredlanguage
- . • pumailautoreplymode
- . • pumailstore

. In addition, the distinguished name is modified to the base form (the form without the email  
: address). The paburi attribute and the user’s PAB entries are modified as defined in Section IX.

. If IMAP access is being turned on, the distinguished name is modified to include the email  
: address and the cn containing only the user’s name is added. The paburi attribute and the user’s  
: PAB entries are modified as specified in Section IX. The attributes listed above, except  
(possibly) memberofmanagedgroup, are added using the same rules as in Section III.

## IX. MODIFYING AN RDN

:  
 : When any operation is performed that causes the LDAP relative distinguished name (RDN) for a  
 : user to be changed, the user's Webmail PAB (Private Address Book) information in LDAP must  
 : be modified appropriately. Actions that cause the RDN to be modified include turning email  
 : access on or off, altering the user's name, and changing the netid.

: The value of the paburi attribute for an active Webmail user is typically of the form:

: ldap://ldap.Princeton.EDU:389/ou=cn=**cn (mail)**,o=Princeton University,c=US,o=pab

: The easiest way to find the associated PAB entries is to search LDAP with a base of "ou=cn=**cn (mail)**,o=Princeton University, c=US, o=pab" and a search argument of "objectclass=\*". All  
 : entries found in this way must be deleted and then replaced with new entries which have their  
 : RDNs modified to reflect the change in the user's **cn** and/or **mail** information. The WHALPAB  
 : script implements the necessary logic. In addition, the paburi attribute in the user's LDAP entry  
 : must be changed to point to the revised PAB.

## X. DELETING A NETID

Whale does not currently remove LDAP entries, other than in the case of replacing an Xalias entry with an entry for a live user. CARPROC does remove LDAP entries, but it first invokes Whale to delete all user accounts for the netid and to mark the netid either obsolete or inactive. (Obsolete netids are typically ones that were entered in error and were never actually used.)

## XI. DELETING AN ACCOUNT

Whenever an account is deleted, the account deletion date is stored in the meta-data. Other actions are taken depending on the type of account being deleted:

### Deleting a Default Account

All other accounts (subids) should be deleted before the default account (.STUDY) is deleted. Once the default account has been deleted, the user no longer has access to OIT systems, even though there may still be a base entry for the netid in LDAP. These are the steps to delete the default account:

1. Signal kprime to remove the Kerberos principal for this netid. (A message "delete **uid**" is sent via TCP to port 756 on kprime (or to port 757 on csgtest2 for testing Whale).)
2. Remove the userpassword, pupsynchpw, and pupsynch attributes from the LDAP entry.
3. If the account was enabled for public unix access, take the steps described above in Section VIII. for turning off unix access.
4. Currently, email access is not removed when the default account is deleted. Instead, it is removed when CARPROC deletes the LDAP entry.

### Deleting a Non-default Account

When a non-default account is deleted, the mailhard and puunixquota attributes may need to be updated to subtract the portion of those quotas that had been being paid for by the deleted account.

### Deleting an Exchange Account

. Deletion of an Exchange account has not yet been implemented on the AD and Exchange server  
 . side, but Whale implements the following logic in anticipation of code being written to update  
 . AD and the Exchange server to automate account deletion:

. When a .EXCHANGE account is deleted, the user is reverted to IMAP mail. Whale first updates  
 . the LDAP entry to remove any puexchangeservice and mailforwardingaddress attributes and to  
 . set the attributes puexchangedisabled (a zulu timestamp), maildeliveryoption (“mailbox”), and  
 . mailbox (“uid@Princeton.EDU”). It then queries winadmin to confirm that AD agrees that this  
 . is an Exchange user:

. action=MAILBOXEXISTS&netid=uid

. If the reply is “Success”, it signals winadmin to delete the Exchange mailbox and move the user’s  
 . mail to IMAP by issuing this CGI GET:

. action=DELETEMAILBOX&netid=uid

## : XII. RENAMING A NETID

: The logic to rename a netid is implemented in the WHALRENM script. When a user wishes to  
 : change to a different netid, tests are first done to determine whether the desired netid is available:

- : • If no existing LDAP entry has a campusid attribute with the value of the requested netid, it is  
 : available.
- : • If an existing LDAP entry has a matching campusid but a different uid, the requested netid is  
 : not available.
- : • If an existing LDAP entry is an Xalias entry and the value of the universityid or  
 : universityidref attribute is the user’s universityid, the requested netid is available.
- : • Otherwise, the requested netid is not available.

### Updating AD

- A USEREXISTS query is sent to winadmin to determine whether it knows about the old netid. If not, no further processing is required for AD.
- If AD does know about the old netid, a MAILBOXEXISTS query is sent to determine whether the old netid belongs to an Exchange user. If it does, the entire rename operation is disallowed, as renaming an Exchange user is not supported at this time.
- Otherwise, Whale signals winadmin to rename the netid by issuing this CGI GET:  

```
action=RENAME&netid=oldnetid&newnetid=newnetid
```

### Updating LDAP

- Any existing LDAP Xalias entry for the new netid is deleted.
- If the netid is contained in the email address in the RDN of the user's LDAP entry, a modrdn operation is performed to change it, and the paburi attribute and the user's PAB are replaced as specified in Section IX.
- The values of all other attributes in the user's LDAP entry that contain the old netid as a "word" are replaced with new values in which the new netid has been substituted for the old netid.
- A mail-forwarding Xalias entry is added for the old netid with a termination date set three months in the future for students or two years in the future for all others. A new dummy universityid is established for this LDAP entry. The Xalias entry, which has a distinguished name of the format "uid=**oldnetid**,ou=Xaliases,o=Princeton University,c=US", contains the following:  

```

campusid: oldnetid
cn: Xalias
emailbox: newnetid@Princeton.EDU
mail: oldnetid@Princeton.EDU
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: puPerson
objectclass: top
ou: Xaliases
pinternalinfo: mm/dd/yyyy delete Xalias entry
sn: Xalias
uid: oldnetid
universityid: universityid
universityidref: universityidref

```

: **Updating IMAP**

: IMAP is notified to rename the user's mailbox via a message sent to the IMAP server that has the  
 : mailbox (port 8321 for production or port 8331 for test on imap1 and port 8322 for production or  
 : port 8332 for test on imap2). The message consists of three linefeed-terminated lines, "olduid:  
 : **oldnetid**", "newuid: **newnetid**", "". The response begins with "success" or "failure".

: **Updating Kerberos**

: Messages are sent to kprime to delete the Kerberos principal for the old netid and to create the  
 : Kerberos principal for the new netid (as in Sections II. and XI. above).

: **Updating NIS**

| After the LDAP entry has been renamed, a signal is sent to port 2823 on  
 | nismaster.Princeton.EDU. The message is in the form "rename **oldnetid newnetid**" (followed by  
 | a linefeed, as always). The response (after the connection banner) is either "OK account **oldnetid**  
 | renamed to **newnetid**" or an error message beginning "ERR".

: **Updating Exchange**

: Not yet implemented.

### **XIII. RECEIVING CHANGED LOGIN SHELLS FROM NISMASTER**

When nismaster receives a request to change a user's default login shell, it sends mail to a daemon on PUCC (vmuoper) instructing it to retrieve the change. Vmuoper uses FTP to get /etc/whale/shlog, which it parses and converts to Whale ALTER SHELL commands. Issuing these commands causes the Whale meta-data and the LDAP loginshell attribute to be updated to match what nismaster has as the shell for the user.

### **XIV. POSTING UNIX LOGIN DATES AND SAMBA MOUNT DATES**

#### **Unix Login Dates**

: When the accounting logs are rolled on the public login systems each night, a program called  
 : /usr/cit/etc/report\_logins.sol2 (on the Arizonas) or /usr/cit/etc/report\_logins.linux (on the Hats) is  
 : run to extract the latest login date for each user in the wtmp(x) files. These eight files are emailed  
 : to a daemon on PUCC (vmuoper3) which merges them and invokes Whale to update the  
 : puunixlastlogin LDAP attribute for each user (if the date is later than the one already there).

: The files from tombstone and sixtyfour are also emailed to the perfddata id, which reads them into  
 : /u/perfddata/[hostname]/logins. A cron job on merrimac invokes /u/melinda/mungers/postlogins64  
 : to process these files and update the puunix64lastlogin LDAP attribute for each user (if the date is  
 : later than the one already there). The date is expressed in seconds since the unix epoch.

A program called `/var/log/syslog.reports/report_logins.auth` is run when the `auth.log` file is rolled on loghost each night to extract the latest login date for each user from the firewall and `sshd2` and `snksu` entries in the `auth.log`. This file is also emailed to `vmuoper3` on PUC and processed in the same way.

A program called `/logs/syslog.reports/report_logins.mac` is run when the `user.log` file is rolled on loghost each night to extract records sent by a logger command in our MacOSX login script. This file is also mailed to `vmuoper3` to use to update the last unix login date.

When Whale updates the `puunixlastlogin` attribute, it writes a message to its log displaying the previous value of that attribute (for use in analyzing account breakins).

### Samba Mount Dates

A program called `/u/melinda/mungers/mungesmbd.pl` is run every four hours from a crontab entry on tango to scan `/var/local/samba/var/log.smbd.old` to extract the most recent Samba mount date for each user. Its output is emailed to a daemon on PUC (`vmuoper4`) which merges the files each night and invokes Whale to update the `pusambalastmount` LDAP attribute for each user (if the date is later than the one already there).

When Whale updates the `pusambalastmount` attribute, it writes a message to its log displaying the previous value of that attribute (for use in analyzing account breakins).

## XV. BILLING FOR DISK QUOTAS

On the first of each month, the `CHGMINI` program runs against the Whale meta-data to generate charges for Exchange, IMAP, and unix disk quotas for the non-default accounts. The charges are posted to Whale's meta-data for both the user account and the corresponding University account, as well as to journal vouchers for the Treasurer's Office.

The meta-data for each non-default account contains the amount of quota, the account to be billed, and the date of the last charging run. With this information, a new charge is calculated and the meta-data records are updated to record the new charge date and to add the amount of the charge to the "amount used" for the account. The rates are as follows:

- Exchange: 25 cents per day for the base charge (55 megabytes) and 10 mils per megabyte per month for additional quota.
- IMAP: 10 mils per megabyte per month.
- Unix: 3 mils per megabyte per month.

Once the charges have been calculated, a file is written containing records with these tab-separated fields:

1. Record type: "D"
2. Request ID: null
3. Charge description:
  - "Unix quota"
  - "Imap quota"
  - "Exchange base"

- “Exchange quota”
4. Requestor: **displayname** (netID **uid**)
  5. Request Date: mmddyyyy
  6. Customer ID: null or 7-digit customer id for outside customers
  7. Project Grant: billed account
  8. PUOptionalID1: null
  9. PUOptionalID2: null
  10. Sub\_class: null
  11. Service Type ID: “FEE”
  12. Service ID: null
  13. Quantity: quota in megabytes
  14. Rate: null
  15. Amount: charge in dollars and cents
  16. Service Description:
    - “Unix added quota for month ending mm/dd/yy”
    - “Imap added quota for month ending mm/dd/yy”
    - “Exchange base service for month ending mm/dd/yy”
    - “Exchange added quota for month ending mm/dd/yy”
  17. Department Card ID: null
  18. Unit of Measure: “Megabytes”
  19. Comment: null

To post these data:

1. Invoke this Web page: <https://issserver11.princeton.edu:7900/>
2. Click on “Departmental Charges” and then sign in with “EMAIL AND DISK STORAGE SERVICES” selected.
3. From left-hand menu, select “Upload Data File” and fill in the path to the file and click on “submit”.
4. Click on “Edit Report” to get an overview of what would be posted.
5. Click on “Detail Data” to get a look at what would be posted.
6. Click on “Release JV” to send the charges to Treasurer.

## Appendix A

### RULES FOR DATA MAINTAINED BY WHALE

#### Variable LDAP Attributes

: Whale maintains the LDAP attributes listed below and populates them according to the rules  
 : given below. Some of these attributes may have other values filled in by other systems; for  
 : example, the attributes with passwords values can contain SHA/SHAA encryptions, but Whale  
 : gives them only unix crypts.

: **campusid:** User's login name for some campus system (not necessarily in the OIT namespace).  
 : There is always one campusid attribute with the value of the OIT-assigned uid attribute, but there  
 : may be others. By testing a proposed new OIT uid against all campusid attributes, Whale  
 : prevents the confusing case of two people having the same login name on different systems on  
 : campus.

**cn:** Name of person or entity in the form "first middle last" without titles or suffixes and with no  
 periods following initials. Other cn attributes (up to two more per entry) include the value of the  
 distinguished name and that value up to its first comma. Those cn entries are created  
 automatically by LDAP.

**displayname:** Name of person or entity in the form "first middle last" without titles or suffixes.  
 Periods are used after initials. User's names are prohibited from containing parentheses or  
 at-signs.

**mailbox:** If the user is at PPL (as indicated by "C-Site" or "PPL" in the **street**), the mailbox is  
 in the form "**uid**@pppl.gov"; otherwise, it is in the form "**uid**@mail.Princeton.EDU", unless the  
 user is subscribed to the Exchange service, in which case it is in the form  
 "**uid**@exchange.Princeton.EDU".

**emailrewrite:** Email address in the form "**uid**@Princeton.EDU".

**gecos:** A three-part string consisting of the **displayname**, the **street** (which may be null and  
 which must have any commas removed), and the **telephonenumber**, each followed by a comma.

**gidnumber:** The default unix gid number is assigned according to the value of **pustatus**:

- 29, stf
- 33, fac
- 34, gNN
- 35, uNN
- 36, adm
- 37, all others

: When a netid is being re-added, the gidnumber is retrieved from the meta-data.

**givenname:** There is no givenname attribute for non-human entities. For humans, the cn is examined for the first word (other than the surname) that is not given as an initial. The first initial is used if only initials are known.

**homedirectory:** The unix home directory in the form “/u/**uid**”.

**loginshell:** The unix shell is originally set to /bin/csh for undergraduates and to /bin/nologin for all others. Legal values are:

- /bin/ash
- /bin/bash
- /bin/bash2
- /bin/bsh
- /bin/csh
- /bin/ksh
- /bin/sh
- /bin/tcsh
- /bin/zsh
- /bin/nologin

When a netid is being re-added, the loginshell is retrieved from the meta-data.

**mail:** The generic Princeton email address for this user. If the user is at PPL (as indicated by “C-Site” or “PPL” in the **street**), the mail attribute is in the form “**uid**@pppl.gov”; otherwise, it is in the form “**uid**@Princeton.EDU”.

**mailalternateaddress:** A multi-value attribute. The default values are “**uid**@Princeton.EDU” and “**uid**@phoenix.Princeton.EDU” and “**uid**@imap1.Princeton.EDU” (undergraduates only) or “**uid**@imap2.Princeton.EDU” (all others).

**mailmessagestore:** The name of the mailmessagestore for the user, either “primary” or “storeNN”.

**mailquota:** IMAP mail quota. For the default account, this is 20 MB for students of any flavor and 55 MB for all other users, except that it is -1 for non-undergraduates associated with departments that pay for private IMAP stores. The value must be a number less than 4294966272. For quotas larger than that, the value should be expressed in megabytes, gigabytes, or terrabytes, using the appropriate suffix (M, G, or T). In practice, new and updated values are always stored in megabytes, but existing values may be in bytes.

**memberofmanagedgroup:** Distinguished name of a quotagroup LDAP entry, *e.g.*, for Electrical Engineering, “cn=eequotagroup,ou=Groups,o=Princeton University,c=US”.

**ou:** University department name.

**paburi:** Whale does not populate this attribute originally (it is built by Webmail), but Whale does modify it appropriately whenever it modifies the relative distinguished name for a user’s LDAP entry. When the paburi attribute is modified, all of the LDAP entries for the user’s Personal Address Book must also be replaced.

**puidisplayname:** Name of person or entity. Up to 32 characters. For humans, the name should be in the form “last, first middle” with no titles or suffixes. Periods are used following initials.

**puhomedepartmentnumber:** University department number (3 digits). Only Whale super-users are allowed to set department numbers of 000 and 999.

**puimapfilesystemdept:** University department number (3 digits) indicating that the user’s IMAP files are kept in a private filesystem paid for by this department.

**pumacosxhomedirectory:** Directory path in the form “/u/**uid**/MACOSXFILES”.

**pumacosxcifsmount:** An XML tag of the form:

```
<home_Dir><url>smb://smbserve/uid</url><path>MACOSXFILES</path></home_Dir>
```

**pupsynch:** Multi-valued attribute listing the systems for which P-synch should allow this user to set passwords. Legal values are:

- AD
- KRB5
- LDAP
- PSYNCH

**pupsynchpw:** P-synch crypted password, in the form “{CRYPT}*cryptstring*”.

**pupwadmodifytimestamp:** *zulu timestamp* when a real AD password was set.

**pupwldapmodifytimestamp:** *zulu timestamp* when a real LDAP (IMAP) password was set.

**pupwpsynchmodifytimestamp:** *zulu timestamp* when a P-synch password was set.

**pupwstring:** Four characters (either the last four digits of the SSN or the birthday in MMDD format) used in setting a starter password. Changing pupwstring (or pupwstringtype) does not alter the current starter password; these attributes are maintained so that future password reset operations will produce a starter password that is meaningful to the user.

**pupwstringtype:** “S” or “B” specifying whether the value of the pupwstring attribute is derived from the SSN or the birthday.

**pusambalastmount:** Date in the form “YYYYMMDD” of most recent Samba mount of the user’s home filesystem.

**pustatus:** University status of person or entity. Legal values are:

- adm - Administrative id
- aff - Affiliate (*e.g.*, staff of McCarter Theatre)
- cap - Community auditor program
- cas - Casual hourly employee
- cwk - Coursework account
- dcu - Departmental computer user (id on departmental system only)
- eme - Emeritus faculty member
- fac - Faculty member
- gNN - NNth year graduate student (NN not greater than 10)

- gxx - Generic graduate student (FERPA requirement)
- lsv - LISTSERV mailing list
- org - Organization
- oth - Other non-human id
- out - Outside (commercial) user
- pup - Princeton University Press user
- ret - Retiree
- sps - Special student
- stf - Staff
- svm - VM service virtual machine
- svr - Unix server
- trs - Trustee
- uNN - Undergraduate member of class of 'NN
- uxx - Generic undergraduate (FERPA requirement)
- vis - Visiting student
- xal - Xalias placeholder
- 'NN - Undergraduate alumnus/alumna, class of 'NN
- \*NN - Graduate alumnus/alumna, degree in 'NN

**puunixfilesystemdept:** University department number (3 digits) indicating that the user's unix files are kept in a private filesystem paid for by this department.

**puunixlastlogin:** Date in the form "YYYYMMDD" of most recent login on a public unix system.

**puunix64lastlogin:** Date in the form of seconds since the unix epoch of most recent login on a public 64-bit unix system.

**puunixquota:** The public unix quota (in kilobytes). The current value for default ("STUDY") accounts is 250 MB. Charged accounts can have any amount of quota, but increases of more 10 GB are allowed only after consultation with the Unix Group to ascertain that space is available. If the user's department has a private unix filesystem, the value of this attribute should be 0.

**seealso:** Distinguished name of a related LDAP entry. Whale does not populate this attribute, but it does remove it when changing the value of the universityidref attribute.

**sn:** User's surname (family name). For non-human entities, this is the same as the cn.

**ssn:** Social Security number (9 digits). Dummy Social Security numbers starting "999" are assigned by Human Resources or the Registrar when the SSN is unknown. **Note:** Whale does not display the Social Security number when it receives the command to display a netid record, unless the Whale operator has Whale super-user privileges.

**street:** The local street address, up to 35 characters.

**telephonenumber:** The telephone number, 12 characters in the form "999-999-9999" or "NONE".

**uid:** The "netid", that is, the login name for OIT systems. Two to eight characters composed of lower-case alphabets, hyphens (discouraged), and numerals. Netids for students should not contain numerics, because of restrictions in Dormnet.

**uidnumber:** Whale assigns unix uidnumbers sequentially as it adds new unix users. The next available uidnumber is stored in the meta-data. Numbers in the range 60000-60009 are to be skipped over. When a netid is being re-added, the uidnumber is retrieved from the meta-data.

**universityembref:** Real universityid for the mailbox to which mail should go (used in “alter-ego” LDAP entries). There should always be a universityidref attribute with the same value, but the reverse is not true.

**universityid:** 9 digits. Dummy universityids assigned by Whale are in the following ranges:

- 099103000-099189999: People
- 098200400-098209999: Lists
- 098105000-098199999: Other non-human entities.

**universityidref:** Real universityid for the person responsible for this netid. The presence of a universityidref attribute indicates that there is another LDAP entry with the specified universityid and that it is the main LDAP entry for this person.

**userpassword:** LDAP (IMAP) crypted password, in the form “{CRYPT}init” (starter password) or “{CRYPT}cryptstring” (real password).

#### Meta-data for Netids

- Netid expiration date
- Flag: Inactive netid
- Flag: Obsolete netid
- Flag: Hidden netid (LDAP entry should not be displayed)
- ADstatus: “student” for undergraduates and special students; “graduate” for graduate students; and “facstaff” for all others.
- List of all netids ever assigned
- List of netids assigned in the alumni system
- Next unassigned unix uid number

#### Meta-data for Accounts

- Account deletion date
- Account registration date
- Dollar amount budgeted
- Dollar amount used
- Billed University account
- Disk quotas (unix, IMAP, and Exchange)
- Disk quota charge dates (unix, IMAP, and Exchange)

**Meta-data for University Accounts**

- Netids of “authorized signatures”
- Dollar amount budgeted
- Dollar amount used

**Meta-data for Departments**

- Department name
- Department number
- Flag: Has Private IMAP Quota
- Flag: Has Private Unix Quota
- Current private IMAP managed quotagroup name
- Current private IMAP mailstore number
- . • Current Exchange server for department
- . • Current Exchange storage group number for department
- . • Current Exchange mailstore for department