NSMAINTAIN

NSMaintain allows you to view and modify objects in the Clearinghouse data base from inside Lisp. Similar operations are available when chatting to a Clearinghouse service.

Requirements

Xerox NS network environment with Clearinghouse server(s).

DES.LCOM.

Installation

Load NSMAINTAIN.LCOM from the library. This file automatically loads DES.LCOM. DES is currently only used by the Change Password command, so its loading can be omitted if you do not need that command.

Clearinghouse Concepts

The Clearinghouse maintains a distributed data base of objects, each of which has a set of properties. The objects are such things as users, groups, and network servers; the properties are such attributes as a server address or a user's mailbox location.

Clearinghouse objects are partitioned into a three-level hierarchy: each object is contained in a domain, which in turn is part of an organization. A fully qualified object name is a three-part name in the form object:domain:organization. Similarly, a domain name is a two-part name of the form domain:organization. Lisp maintains a notion of the default domain, which is typically the domain in which you and the servers in your immediate area are registered. When typing object names, you may omit the organization field or both domain and organization fields if they are the same as your default domain. Similarly, when typing a domain name, you may omit the organization field if it is the same as the default.

When printing the names of objects, the system usually elides the domain and/or organization, following the same rules. For example, for the object named "John Jones:Sales:ACME", the system would print "John Jones:" if the default domain were "Sales:ACME", or "John Jones:Sales" if the default domain were "Admin:ACME". NSMaintain, however, prints fully qualified names in certain places that do not need the compactness of the elided names, so as to reduce potential confusion. For the same reason, whenever NSMaintain prompts for an object name and you omit one or two of the fields, NSMaintain automatically echoes the defaults for you. You can change the defaults with the "Change Default Domain" command.

Any object in the Clearinghouse can have one or more aliases, which are Clearinghouse names that point directly to the object. An alias can be thought of as a "nickname", and can be used interchangeably with the "real name" for virtually all operations. For example, it is common practice to register users with their full names and provide at least one alias consisting of their last names.

Some objects in the Clearinghouse are groups, rather than individuals. Groups are described further in the section on group commands.

For more information on the Clearinghouse service, consult the *Interlisp-D Reference Manual* or the Clearinghouse documentation, which is part of the Network Systems documentation kit.

User Interface

NSMaintain runs in an Exec window. To start it up, evaluate:

`(NSMAINTAIN)` [Function]

Starts an NSMaintain session. It prompts with "CH:" in the current window and awaits commands from you. Command names complete automatically following one- or two-letter inputs. Type Q, for the Quit command, when you wish to finish.

Most of the commands take as input from you one or more Clearinghouse object names. For many commands, NSMaintain offers you the same name as you last used in a similar context. For example, if you use the Describe command to learn about an object that is a group and then use the List Members command, NSMaintain offers you the group name just described. To accept the name, just press the carriage return; otherwise, start typing the desired name; your type-in replaces the offered name. The alphabetic case of names is not significant; you may type in either upper or lowercase. However, the commands that create objects preserve the exact case of the name as you first type it.

Typing a null name to most commands aborts the command. If a sample name is offered, you have to backspace over it, or use Control-Q to erase the whole name. You can also usually use Control-E to abort a command.

The description of the commands below is partitioned into two parts: general user commands and administrator commands. The user commands can be used by anyone, and mostly are concerned with viewing the data base. The administrator commands allow system administrators to modify the data base; these commands cannot be used by ordinary users. However, there are two administrator commands, Add Self and Remove Self, that can be used by anybody to join or leave groups with open access.

User Commands

Anyone can use these commands to obtain information, change passwords, and change NSMaintain's defaults.

Obtaining Information

These commands let you examine the database.

Most of the List commands enumerate items in the database matching a particular pattern. A pattern is a Clearinghouse name optionally containing asterisks as wild cards, which match zero or more characters. Wild cards are permitted only in the first component of the name; the remaining parts must be a valid domain and organization. In a two-part name, wild cards are permitted in the domain name but not the organization. Thus, for example, "*John*:Sales:ACME" is a valid pattern matching objects whose name component contains the substring "John"; "Joe:*:ACME" is not a valid pattern.

Following a List command (except List Domains), you can use the Show Details command to get more information about any of the names listed.

`Describe`	Gives a description of any object registered in the Clearinghouse, what its registered name is (in case you typed an alias), and all interesting properties of the object. If the object is a group, its Owner and Friends are also listed; to see its members, use the List Members command.
`List Domains`	Lists all domains matching a specified domain pattern; for example, "*:Xerox" to list all domains in the Xerox organization.
`List Clearinghouses`	Lists all Clearinghouse servers that serve a specified domain.
`List Administrators`	Lists the administrators for a domain.
`List Aliases`	Lists all aliases matching a given pattern. Note that none of the other List commands (except for List Objects with property any) match your pattern against an alias, so you may want to use the List Aliases command if you do not find the object you were looking for otherwise.
`List Groups`	Lists all groups matching a given pattern.
`List True Groups`	Same as List Groups, but filters out all names that also have a "user" property. These "groups" are typically used for mail forwarding. This command requires considerably more computation than List Groups.
`List Servers`	Lists objects matching a given pattern and registered as a server. You are prompted for the type of server; for example, Mail, File, or Print. Type ? to see the choices.
`List Users`	Lists the names of all users matching a given pattern.
`List Objects`	Lists all registered objects of of an arbitrary Clearinghouse type that match a given pattern. You are prompted for the type(a Clearinghouse property name) and pattern. To list all objects, that is, those with any property, press the carriage return to the property prompt, or supply the property "*".
`List Members`	Lists the members of a specified group.
`Show Details`	Prompts you for a name, performing automatic spelling completion from the names printed in the most recent List command such as List Users, and then performs the Describe command on the name. Press the carriage return in response to the name prompt to return to the main CH: prompt. If there was only one name in the list, this command does a Describe on it without further prompting.
`Type Entry`	Synonym for Describe.
`Type Members`	Synonym for List Members.

Miscellaneous

`Change Default Domain` Changes the defaults used on type-in inside NSMaintain for domain and organization. NSMaintain asks if you also want to change the defaults globally; if you say yes, the variables `CH.DEFAULT.DOMAIN` and `CH.DEFAULT.ORGANIZATION` are changed, so that the new defaults have effect outside of NSMaintain as well.

The defaults are never used when typing aliases in the Add Alias and Add User commands; these commands default the domain to be the same as the domain of the main object.

Note: Unless you change the defaults globally, this command does not affect type-out and the Change Login command, which are still performed with respect to the global defaults. This may change in a future implementation.

`Change Login` Prompts you for a new name and password, which becomes the default NS login on your machine and for NSMaintain. You can also use this to fix your password if you were incorrectly logged in before you started NSMaintain.

`Change Password` Allows you to change your password. Prompts for a user name, offering your logged-in name as default. A domain administrator can also change other users' passwords. After you type the new password, you are asked to retype the password, to ensure that you typed what you thought you had. Neither password is echoed.

`Quit` Exit NSMaintain.

Administrator Commands

These commands modify the Clearinghouse database. In general, they require that you have the appropriate administrator access.

Creating and Deleting Objects

To create or delete objects in a domain, you must be an administrator for the domain.

`Add Alias` Assigns an alias for a specific object registered in the Clearinghouse database.

`Add User` Creates a new user. You are prompted for the user's name (preferably a full name), a brief description (for example, the user's affiliation, office number, etc.), an initial password, and one or more aliases.

`Change Remark` Allows you to change the remark; that is, text description, of any object in the database. NSMaintain prompts you for the object name, then for a new remark, offering the old remark as default.

`Remove Alias` Removes an Alias from the database. You are prompted for the alias. This has no affect on the primary object for which the removed name is an alias.

`Remove User` Undoes the effect of Add User; that is, removes a user name and all its properties from the database.

`Remove Registered Object` Removes a specified object and all its properties from the database. The primary name and description of the object are printed first and you are asked to confirm the deletion. You can use this to remove groups and other kinds of objects.

Manipulating Groups

A group is a Clearinghouse object with members. Groups are most commonly used for mailing lists and access control. The members can be either individuals or other groups. In the case of groups used for access control, a member can also be a pattern in which "*" usually replaces one or more entire fields of a three-part name.

A group has associated with it two access control lists: owners and friends. Owners can make any change to a group; they are like domain administrators for the narrow scope of the group itself. Friends are allowed to add or remove themselves from the group. For example, common interest groups typically have "open" membership, consisting of a friends list of "*:domain", or even "*:*:*" for a completely open group.

If the Owners or Friends list is empty, it defaults to the administrators of the domain. However, if the Owners list is non-empty, it overrides the administrators list. For example, if you remove yourself from the owners of a group, you can no longer modify the list, even though you are a domain administrator. The defaulting can lead to confusion, especially since the Describe command does not (and unfortunately cannot) indicate whether the owners and friends it displays are explicit or defaulted. For example, if a group previously had no explicit owners, then the Remove Owner command cannot be used, and any use of the Add Owner command implicitly removes all the domain administrators.

`Add Group` Creates a new user group. You are prompted for the group's name, a short description of the group, its initial members one at a time, its owners and its friends. NSMaintain checks all of the names except those that are patterns to ensure that you gave valid Clearinghouse names, and to resolve aliases. The Clearinghouse does not actually require that members of a group be registered Clearinghouse names, as it does not attach explicit meaning to the contents of a group until told to do so. Thus, if you type an invalid name, NSMaintain asks whether you really meant it, and keeps the name if you answer yes. Note, however, that any group used for access control must contain only registered Clearinghouse names or patterns.

If you specify any owners, you are always made an owner yourself as well, whether you explicitly said so or not, so as to avoid the anomaly of your not being able to further modify the group. You can, of course, remove yourself afterward if you really meant to.

`Add Member`
`Add Friend`
`Add Owner` Adds a member, friend or owner to a group.

`Add Self` Adds you, the currently logged in user, to a group. You must be a friend or owner of the group.

`Remove Member`

Remove Friend Remove Owner	Removes a specified member, friend or owner from a group.
Remove Self	Removes you, the currently logged in user, from a group. You must be a friend or owner of the group.

Manipulating Domains

These commands change the list of administrators of a domain. You must be an administrator of the domain or the parent organization to do this.

Add Domain Administrator	Adds a user to the set of administrators for a domain.
Remove Domain Administrator	Removes a specified user from the administrators for a domain.

Errors

NSMaintain always ends each command with some sort of feedback about the completion of the operation. In information commands, the feedback is, of course, the requested information. In commands that change the database, NSMaintain usually prints "done". If a command fails, NSMaintain prints a terse error message. Listed here are some of the more common ones:

NoSuchObject	You asked about a name that does not exist in the Clearinghouse database. Check that the spelling and the domain are correct.
IllegalOrganization IllegalDomain IllegalObject	The name you gave is not legal as a Clearinghouse name. Since NSMaintain already checks for incorrect use of asterisks, this usually means the name is too long. (The name component must be no more than 40 characters long; domains and organizations are limited to 20 characters each.)
Missing	The name you specified for a group, such as in the AddMember command, is not a group.
CredentialsInvalid VeriferInvalid	You are logged in incorrectly; that is, either your name or your password is incorrect■ You can use the Change Login command to log in correctly.
AccessRightsInsufficient	You do not have the authority to make the change you requested. You can find out who does have the authority by using the command Describe for changing a group, or List Domain Administrators for all other changes.
NoChange	The change you requested would have no effect; for example, you added to a group a name that was already a member, or requested to remove a name that was not there.
TooBusy	The Clearinghouse contacted by NSMaintain was too busy to field the request. Lisp's present Clearinghouse implementation, unfortunately, does not handle this error, so passes it along to you. If you repeat the operation it may succeed. If this error persists for a long time, you may

want to evaluate `(START.CLEARINGHOUSE T)` to completely clear the Clearinghouse cache; the system may then succeed in locating a more responsive server.

Examples

In the example session that follows, all user input is in boldface; everything else is typed by the system. To avoid clutter, carriage returns typed by the user are not shown. In many cases, a simple carriage return accepts the default input typed by the system, or completes a partially typed name. For clarity, most of the user input is in uppercase, although lowercase is equally acceptable. Commentary is in italics.

```
64> (NSMAINTAIN)
[Default login:  Arthur Dent:Research:ACME;
 Default domain: Research:ACME]
                                        NSMaintain shows me the defaults.
                                        (My password has not, however, been verified.)

CH: Describe name: EDISON:Research:ACME ...

Thomas A. Edison:Research:ACME is a User (Electronics Div., Rm 2732)
Aliases: Edison:, Wizard:               Domain and organization are elided here
Mailboxes: [Time:  8-Aug-86 17:30:54; Mail.Service: (Snail:)]
Userdata: [Last.Name.Index: 10; File.Service: Phylum:]
                                        The Userdata property is used by Viewpoint

CH: List Groups by pattern: *:Research:ACME ... AllResearch, Consultants,
ED, LispImplementors, LispInterest, NetAdministration, Skiiers, Staff,
WireBusters

CH: Show Details of previously listed names

  name: Consultants                     "C<cr>" is all that I typed

Consultants:Research:ACME is a User Group (Part-time personnel)
Owners: Staff:
Friends: NetAdministration:

  name: SKiiers

Skiiers:Research:ACME is a User Group (Snow sport enthusiasts)
Owners: UserAdministration:All Areas, Perry White:
Friends: *:*                            Anyone in organization ACME can join

  name:

CH: List Members of group: Skiiers:Research:ACME ...
Alexander G. Bell:Telcom, Christopher Craft:, Staff:

CH: Add Self to group: Skiiers:Research:ACME ...  done
                                        Skiiers was offered as default, being the last
                                        group I mentioned—I had only to type a cr.
```

```
CH: Add Self to group: Skiiers:Research:ACME ...  failed: NoChange
                                        I.e., I'm already a member

CH: List Servers of type FIle
 by pattern: *:Development:ACME ... Arrow, Quiver

CH: List Users by pattern: Ed*:Research:ACME ... (none)
                                        There are no users whose full name starts "Ed"

CH: List ALiases by pattern: Ed*:Research:ACME ... Edison, Educators
                                        But there are some aliases (not necessarily all users)

CH: List objects having property wORKSTATION
 by pattern: *M*:Research:ACME ... Archimedes, Camero, Cardamom, Homestead,
MayDay, Mendel, Ramanujan, SatanicMechanic, TheTajMahal

CH: Show Details of previously listed names
  name: Archimedes

Archimedes:Research:ACME is a Workstation (1186 in Rm. 2732)
Address.List: (6285#0.125101.20200#0)
Authentication.Level: [Simple: true; Strong: false]

CH: Change Default Domain (for name entry) to be: Development:ACME
Set this default globally as well (i.e. for use outside Maintain)? N

CH: Add Alias for object: Newton:Development:ACME
 Alias: Isaac:Development:ACME ...  done

CH: Describe name: Newton:Development:ACME ...

S. Isaac Newton:Development:ACME is a User (Apple Tester, Rm. 34)
Aliases: Isaac:Development, SIN:Development
Userdata: [Last.Name.Index: 0; File.Service: Arrow:Development]

CH: Remove Alias alias: SIN:Development:ACME ... done, alias was removed
from S. Isaac Newton:Development

CH: Add User
New user's name: Charles S. Brown:Development:ACME ...
Remark (terminate with CR): Test Team captain
Alias: Chuck:Development:ACME
Alias: Brown:Development:ACME
Alias: CSB:Development:ACME
Alias:  xxx                             Bare <cr> was typed to end the list
Initial password: ******* (retype password) *******...  done
                                        Chuck can later use Change Password to
                                        set a password of his own choosing.

CH: Add Group
New group name: Entomologists:Development:ACME ...
Remark (terminate with CR): Seekers of bugs

Enter names of members, owners and friends, one per line, terminated with a
blank line.
```

```
Member: brown:Development:ACME = Charles S. Brown:Development:ACME
                                        NSMaintain resolves alias, so that member
                                        names are in canonical form
Member: F. Kafka:Development:ACME
Member:  xxx

(If you enter no owners, the group will be owned by the administrators of
Development:ACME.)

Owner: brown:Development:ACME = Charles S. Brown:Development:ACME
Owner:  xxx

Friend: *:Development:ACME
Friend: *:Research:ACME
Friend:  xxx

Adding members... done
Adding owners... (including Arthur Dent:Research:ACME) done
                                        I'm an owner, too (else I couldn't modify the group)
Adding friends... done

CH: Remove User: BILBO:Development:ACME ...
Bilbo Baggins:Development:ACME (Furry ring finder)
 Confirm deletion (y or n): Y
 done

CH: Quit [confirm]                      Back to the exec now.
```

[This page intentionally left blank]