To: Distribution

From: Peter Haber

Subject: Proposed new mail and send_message commands

Date: February 12, 1974

This document discusses the proposed implementation of the new mail and send_message commands. Please comment and return. A design review of these commands will take place on Monday, March 4 ~~February 24~~ at 2:30 p.m. at 545 Tech Square in the 5th floor conference room. Please remember in commenting that any number of options are possible for these commands. Consider the usefulness of an option before suggesting it.

I. New type of message segments.

This section discusses changes to the message segment facility needed for implementation of the new commands. The mail and send_message commands will both use a mailbox message segment in which to add and from which to read and delete messages. A mailbox message segment

a. resides in a user's home directory.

b. has the name Person.mbx, where person is the registered name of the owner.

c. has associated with it the following extended access:

1) (a,d,r,o,s) - as defined for queue message segments.

2) w - If a user has this access to a mailbox, he may send a wakeup to an interactive process which "owns" the mailbox.

3) u - If a user has this access he may send a process which owns the mailbox an "urgent" wakeup (the differences between the two types of wakeups are discussed in the writeup of send_message).

II. This section discusses the reasons for the proposed change to the usage of the send_message command.

The current send_message command allows a sender to say

send_message Person Project Message.....,

causing Message..... to be sent to Person.Project. The sender may also type

send_message Person Project

and the send_message facility will respond with

Input

whereupon the user types on one line the message he wishes to send.

It is proposed that the interface be modified to

send_message path Person1 Project1...PersonN ProjectN...option1...optionN

If path is "*" the command responds with

Input

and the user types his message terminating with the line ".HL". If path is not "*", it is assumed to be the relative pathname of a file which send_message sends as a message.

This new interface has one disadvantage...a sender must type an extra new line to send a one line message. For this disadvantage the following is gained:

a. The sender may send a message to more than one user with one invocation. The current implementation will not allow distinction between

send_message Person1 Project1 Message Word1 Message Word2

and

send_message Person1 Project1 Person1 Project2

b. Options may be more easily specified. With the new implementation an argument either has a leading hyphen, in which case it is an option, or does not have a leading hyphen, in which case it is a part of a person-project pair. With the current implementation, a nonhyphenated argument might be part of a person-project pair.

c. The user interface to the mail and send_message commands will be the same.

Attached are writeups which define the command interface to the mailbox message segment facility.

Name: mailbox_create, mbcr

This command is used to create a mailbox message segment.

Usage

```
mailbox_create -path
```

where

path is the relative pathname of the mailbox to be created. If not specified, the mailbox message segment "mailbox.mbx" is created in the caller's working directory.

Notes

The mailbox will be created with full extended access for the creator and null extended access for *.*.*. See mailbox_setacl notes for a description of available extended access.

If the suffix ".mbx" is not specified in path, it is assumed.

Command

Name: mailbox_delete, mbdl

This command is used to delete a mailbox message segment. In order to delete a mailbox, the caller must have "modify" access to the containing directory and "delete" extended access to the message segment.

Usage

```
mailbox_delete -path
```

where

path is an optional relative pathname of the mailbox to be deleted. If not specified, the segment "mailbox.ms" in the working directory will be assumed.

Notes

If -path does not end with the suffix ".mbx", this suffix will be assumed.

See mailbox_setacl for a description of the extended access to mailbox message segments.

Name: mailbox_setacl, mbsa

This command adds items to the Access Control List (ACL) of a mailbox message segment.

Usage

mailbox_setacl pathname acl acname1...acn acnamen

where

1) pathname — is the relative pathname of mailbox message segment whose ACL is to be updated. The star convention may be used.

2) aci — is the access of the access control name specified by the next argument with respect to pathname. It may consist of any or all of the letters "adroswu" or, to deny access to acnamei, it may be n or "".

Notes

The following table defines briefly the extended access attributes for mailbox message segments. If the user has access he may perform action.

access	action
a	add messages
d	delete messages
r	read messages
o	read and/or delete own messages
s	obtain status information
w	send "send_message" wakeup
u	send urgent "send_message" wakeup

If the suffix ".mbx" does not appear at the end of pathname, it is assumed.

Name: defer_message, dm

This command is used to control access to the facility which sends "send_message" wakeups to a user on behalf of another user (see send_message, allow_message). Its usage requires that the allow_message command has been invoked.

Usage

```
defer_message -option
dm -option
```

where

-option	may be:
-all, -a	if specified, defer all messages. The default is to defer only messages of normal priority and not to defer urgent messages.

Name: allow_message, am

This command is used to control access to the facility which sends "send_message" wakeups to a user on behalf of another user. Upon creation of his process a user must invoke this command before send_message facility will send him a wakeup (see send_message, defer_message).

Usage

allow_message, am

There are not arguments.

Command

Name: mailbox_listacl, mbla

This command lists some or all of the items on the Access Control List (ACL) of a mailbox message segment.

Usage

mailbox_listacl pathname acname1...acnamen -option

where

1) pathname — is the relative pathname of the mailbox message segment whose ACL is to be listed. The star convention may be used.

2) acnamei — is an access control name. If no acname is specified or if it is -all (-a), the entire ACL of the mailbox message segment is listed. Otherwise, acname must be of the form namei.projecti.tagi. If all three components are present, the ACL is searched for the identical name. If one or more of the components is missing, the ACL is searched for all names with the given components. Note: any components missing on the left must be delimited by dots; however, the dots may be omitted on the right.

Notes

If the suffix ".mbx" does not appear at the end of pathname, it is assumed.

See mailbox_setacl for a definition of extended access with respect to mailbox message segments.

Name: mailbox_deleteacl, mbda

This command deletes some or all of the deleteable items form the Access Control List (ACL) of a mailbox message segment.

Usage

mailbox_deleteacl pathname acname1...acnamen

where

1) pathname — is the relative pathanme of the mailbox message segment whose ACL is to be changed.

2) acnamei — is an access control name. If there is no acname specified, the user's process group ID is deleted. If it is -all (-a) the entire ACL will be replaced with an ACL specifying null extended access form the caller and *.SysDaemon.*. Otherwise, acnamei must be of the form namei.projecti.tagi. If all three components are present, that identical name is deleted from the ACL if it is found there. If one or more of the components is missing, all names on the ACL with the given components are deleted. Note: any components missing on the left must still be delimited by dots. However, the dots may be omitted on the right.

Notes

If the suffix ".mbx" does not appear at the end of pathname, it is assumed.

See mailbox_setacl for a definition of extended access with respect to mailbox message segments.

Name: send_message, sm

This command is used for the purpose of sending console messages. Its usage requires that the receiver of a message have created a mailbox message segment "mailbox.ms" in his home directory.

Usage

send_message file Pers1 Proj1...PersN ProjN -opt1...-optN

where

file	specifies the nature of the input to the send_message command. If equal to "*", this command will type "Input:NL". The command will then read input from the console, terminating the read when a line ".NL" is read, and use the preceding input as a message. If not equal to "*", the argument is assumed to be the relative pathanme of an ASCII file to be used as a message.
Pers(i) Proj(i)	specifies a person-project to whom the message is to be sent. The star convention will not be recognized.
opt(i)	may be chosen from the following:
1) -nosave, -ns	do not save console input if the message is unsuccessfully sent. Default is to save console input in the segment "unsent_message" in the sender's working directory.
2) -reply, -rp	have the receivers process send an acknowledgement message when it reads the message. The default is not to request a reply.
3) -nohold, -nh	do not send the message if a wakeup cannot be sent to the user. The default is to leave the message in the receiver's mailbox.

Notes

If the planned receiver of a message has deferred messages and has not deferred urgent messages, and if the sender has access to send urgent messages, the sender will be asked if the message is urgent. If he answers "yes", a wakeup will be sent to the receiver and the message will be sent. If he answers "no"

the wakeup will not be sent. The message will be sent if the "-nohold" option was not specified.

Name: mail, ml

This command is used to send and read ASCII messages. Its usage requires that the receiver of a message have created a mailbox message segment "mailbox.ms" in his home directory.

Usage

To read mail:

mail -path -option(1)...-option(N)

where

1) path — is the relative pathname of the mailbox to be read. If not specified, the caller's mailbox will be read.

2) option(i) — may be chosen from the following:

-person Person Project -pers Person Project	specifies that all messages from Person.Project are to be read. Project may be "*". The default is to read all new messages. (See notes.)
-all, -a	specifies that all messages are to be read. The default is to read all new messages. (See notes.)

To send mail:

mail file Person(1) Project(1)...Person(N) Project(N) option(1)...option(N)

where

1) file — specifies the nature of input to the mail command. If equal to "*", the command types "InputNL" and accepts the message from the console, terminating the input when a line ".NL" is read. If not equal to "*", the argument is assumed to be the relative pathname of an ASCII file to be used as a message.

2) Person(i) Project(i) — specifies a person-project to whom the mail is to be sent. The star convention is not recognized.

3) -option(i) may be chosen from the following:

-nosave, -ns	do not save console input if the message is unsuccessfully sent. Default is to save console input in the segment "unsent_message" in the sender's working directory.
-reply, -rp	have the receiver's process send an acknowledgement message when it reads the mail. The default is not to request a reply.

Notes

Once a message has been read by the owner of the mailbox in which it resides and not deleted, it will not be printed again by this command unless the command is invoked with the "-all" option. If the command is invoked without the "-all" option and previously read messages exist in the mailbox, a message to that effect will be printed.